



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Solução Orientada a Agentes Inteligentes para Controle Glicêmico de Pacientes Internados em UTI

Tiago Henrique Faccio Segato

Dissertação apresentada como requisito parcial
para conclusão do Mestrado em Informática

Orientadora
Prof.^a Dr.^a Célia Ghedini Ralha

Brasília
2021

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Mestrado em Informática

Coordenadora: Prof.^a Dr.^a Genáina Nunes Rodrigues

Banca examinadora composta por:

Prof.^a Dr.^a Célia Ghedini Ralha (Orientadora) — CIC/UnB
Prof.^a Dr.^a Maria Emília Machado Telles Walter — CIC/UnB
Prof. Dr. Hervaldo Sampaio Carvalho — FM/UnB

CIP — Catalogação Internacional na Publicação

Segato, Tiago Henrique Faccio.

Solução Orientada a Agentes Inteligentes para Controle Glicêmico de Pacientes Internados em UTI / Tiago Henrique Faccio Segato. Brasília : UnB, 2021.

89 p. : il. ; 29,5 cm.

Dissertação (Mestrado) — Universidade de Brasília, Brasília, 2021.

1. sistemas multiagentes, 2. controle glicêmico, 3. unidade de terapia intensiva, 4. raciocínio baseado em regras, 5. modelos de predição

CDU 004

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília-DF — Brasil

Dedicatória

Dedico esta dissertação a minha esposa Alessandra e ao meus pais, Sidnei e Maurise por todo o amor e incentivo.

Dedico também a minha orientadora Prof.^a Dr.^a Célia Ghedini Ralha, pelos ensinamentos, orientação, paciência, confiança e amizade.

Agradecimentos

Agradeço

A Deus, pela vida, inteligência e oportunidade do estudo.

Aos meus pais, Sidnei e Maurise por sempre me apoiar e incentivar a levar os estudos com muito afinco, além de serem uma fonte de amor e carinho inesgotável.

A minha esposa Alessandra que acompanhou todo o processo desse trabalho de perto compartilhando todos os momentos difíceis, me confortando e me encorajando a seguir em frente.

A Prof.^a Dr.^a Célia Ghedini Ralha pela orientação e direcionamento ao longo desse processo, além dos conselhos, ensinamentos e experiências. É imensurável a quantidade de conhecimento adquirido por mim nesses mais de 60 encontros.

Aos meus amigos pelos momentos de alegria, por todo apoio e pela compreensão da minha ausência nos últimos tempos.

Aos meus colegas Diego Cesar Florencio de Queiroz e Rafael Moura da Silva que contribuíram com ensinamentos relevantes para o desenvolvimento desse trabalho.

A meu terapeuta Marcelo Marinho Vieira Brandão que em um dos momentos mais difíceis trouxe em suas palavras a calma que necessitava para seguir em frente nesse processo.

A todos os profissionais da saúde que contribuíram com conhecimentos e opiniões acerca da aplicação prática desse trabalho.

Agradeço especialmente o Dr. Sérgio Eduardo Soares Fernandes pela colaboração durante toda a pesquisa, fornecendo embasamento teórico e conhecimento técnico na área médica.

Resumo

Este trabalho apresenta uma solução baseada em agentes inteligentes para controle glicêmico (MAS4GC - *Multi-Agent System for Glycemic Control*) de pacientes críticos internados em UTI. O controle glicêmico está incluído no mnemônico FAST HUG, este é utilizado para lembrar os profissionais da saúde dos itens essenciais no tratamento e suporte aos pacientes criticamente enfermos. Mais especificamente, o MAS4GC gerencia a glicemia dos pacientes por meio de previsões glicêmicas, além de fazer recomendações de monitoramento e tratamento aos profissionais de saúde. Os modelos de previsão são aplicados para monitorar os níveis de glicose no sangue dos pacientes, permitindo que os profissionais de saúde realizem acompanhamento preventivo. As recomendações se baseiam nos mecanismos de inferência dos agentes que atuam com regras de produção definidas segundo conhecimento de um médico especialista em terapia intensiva. O processo de desenvolvimento metodológico do MAS4GC é apresentado por meio da modelagem Tropos, projeto arquitetônico e implementação com o *framework PADE*. Dois experimentos foram conduzidos para avaliar os resultados do MAS4GC utilizando dados sintéticos de pacientes. Os resultados considerando os pacientes críticos incluem: (1) o modelo de previsão alcançou 90% de precisão de glicose no sangue para as próximas quatro horas e (2) 88% de similaridade das recomendações de tratamento em comparação com um especialista humano, além de 96% nas recomendações para monitorar a glicemia de pacientes críticos.

Palavras-chave: sistemas multiagentes, controle glicêmico, unidade de terapia intensiva, raciocínio baseado em regras, modelos de previsão

Abstract

This work presents a solution based on intelligent agents for glycemic control (MAS4GC - Multi-Agent System for Glycemic Control) of critically ill patients admitted to the ICU. Glycemic control is included in the FAST HUG mnemonic, which is used to remind health professionals of the essential items in the treatment and support of critically ill patients. More specifically, the MAS4GC manages patients' blood glucose through glycemic predictions, in addition to making monitoring and treatment recommendations to healthcare professionals. Prediction models are applied to monitor patients' blood glucose levels, allowing healthcare professionals to perform preventive monitoring. The recommendations are based on the inference mechanisms of agents who work with defined production rules according to the knowledge of a physician specialized in intensive care. The MAS4GC methodological development process is presented through Tropos modeling, architectural design, and implementation with the PADE framework. Two experiments were conducted to evaluate the results of the MAS4GC using synthetic patient data. Results considering critical patients include: (1) the prediction model achieved 90% blood glucose accuracy for the next four hours and (2) 88% similarity of treatment recommendations compared to a human specialist, in addition to 96% in the recommendations for monitoring blood glucose in critically ill patients.

Keywords: multi-agent systems, glycemic control, intensive care unit, rule-based reasoning, prediction models

Sumário

1	Introdução	1
1.1	Problema e motivação	2
1.2	Objetivos	3
1.3	Hipótese e questões de pesquisa	3
1.4	Organização do manuscrito	3
2	Fundamentação Teórica	4
2.1	Gerenciamento de UTI	4
2.1.1	FAST HUG	5
2.1.2	Controle glicêmico	7
2.2	Sistema multiagente	12
2.2.1	Modelagem de agentes	12
2.2.2	Raciocínio de agentes	16
2.2.3	Tecnologias de SMA	19
2.3	Avaliação de resultado	22
3	Revisão da Literatura	25
4	Apresentação da Proposta	34
4.1	Pré-projeto de agentes	35
4.2	Modelagem Tropos	36
4.2.1	Requisitos iniciais	36
4.2.2	Requisitos finais	37
4.2.3	Projeto arquitetural	37
4.2.4	Projeto detalhado	40
4.2.5	Diagramas de implementação	41
4.3	Arquitetura	44
4.4	Implementação	45
4.4.1	Interface Web Glycon	45
4.4.2	<i>Framework</i> PADE	48
4.4.3	Biblioteca Experta	50
5	Experimentos	54
5.1	Definição do <i>dataset</i>	54
5.2	Experimento 1	56
5.2.1	Experimento 1: análise dos resultados	58
5.3	Experimento 2	59

5.3.1	Experimento 2: análise dos resultados	61
5.3.2	Validação de teste diagnóstico	62
6	Conclusões	65
6.1	Publicações	65
6.2	Trabalhos futuros	66
	Referências	67
A	Tabela de atributos das classes do MAS4GC	72
B	Lista de Fórmulas usadas no Simulador	75

Lista de Figuras

2.1	Escala de valores glicêmicos.	11
2.2	Componentes visuais do <i>framework I*</i>	13
2.3	Elementos de conexão do <i>framework I*</i>	14
2.4	Diagrama ilustrativo de componentes e elementos do <i>framework i*</i>	14
2.5	Validação de um teste diagnóstico (adaptado de Knottnerus and Buntinx (2011))	24
3.1	Taxonomia de métodos de IA (Contreras and Vehi, 2018).	26
3.2	Trabalhos correlatos conforme a taxonomia de Contreras and Vehi (2018)..	31
4.1	Metodologia de desenvolvimento do trabalho.	34
4.2	Diagrama Tropos referente a fase de requisitos iniciais.	37
4.3	Diagrama Tropos referente a fase de requisitos finais.	38
4.4	Diagrama Tropos referente a fase de projeto arquitetural.	39
4.5	Diagrama de sequência referente a fase de projeto detalhado.	40
4.6	Diagrama de atividades referente a fase de projeto detalhado.	41
4.7	Diagrama de classes do sistema MAS4GC.	42
4.8	Estrutura da <i>collection</i> paciente.	43
4.9	Arquitetura do MAS4GC.	44
4.10	<i>Dashboard</i> de gerenciamento dos pacientes - Glycon.	46
4.11	Tela inicial com a lista de pacientes e respectivas recomendações - Glycon.	46
4.12	Dados de paciente gerado pelo simulador.	48
4.13	Execução da troca de mensagem ente os agentes.	49
4.14	Diagrama de representação do conhecimento do agente PAA.	51
4.15	Diagrama de representação do conhecimento do agente PTA.	52
4.16	Diagrama de representação do conhecimento do agente AMA.	52
5.1	Dados de um paciente gerados na planilha eletrônica.	55
5.2	Ilustração de dados de um paciente simulado.	56
5.3	<i>Workflow</i> metodológico do experimento.	57
5.4	Tabela comparativa entre os modelos de regressão utilizados.	60
5.5	Resultados da comparação dos modelos de RLS1 e RLM3 utilizando o coeficiente R^2 e medidas de avaliação de erro.	60
5.6	Resultado da validação de teste diagnóstico para tratamento.	63
5.7	Resultados do validação de teste diagnóstico para Monitoramentos.	64
B.1	Detalhamento das fórmulas usadas no simulador.	75

Lista de Tabelas

2.1	Escala de valores glicêmicos.	10
2.2	Indicações de monitoramento e tratamento.	11
3.1	Comparação dos trabalhos correlatos.	33
4.1	PEAS do PAA.	35
4.2	PEAS do PTA.	36
4.3	PEAS do AMA.	36
4.4	Dados do grupo usados pelo simulador.	47
5.1	Comparação de recomendações de tratamento e monitoramento.	62
A.1	Atributos da classe <i>Paciente</i>	72
A.2	Atributos da classe <i>Glicemia</i>	73
A.3	Atributos da classe <i>Aplicacao</i>	74
A.4	Atributos da classe <i>Agentes</i>	74

Lista de Abreviaturas e Siglas

- ACL** *Agent Communication Language*. 20
- AMA** *Adjust Monitoring Agent*. 35, 40, 44, 50, 51, 53
- AMGC** AutoMonitoramento da Glicemia Capilar. 9
- AMS** *Agent Management System*. 19
- ANN** Artificial Neural Networks. 29
- Anvisa** Agência Nacional de Vigilância Sanitária. 4
- BD** Banco de Dados. 42, 44
- BRMS** *Business Rule Management System*. 20
- CAPES** Coordenação de Aperfeiçoamento de Pessoal de Nível Superior. 26, 27
- CEG** *Clarke Error Grid*. 29, 33
- CGM** Continuous Glucose Monitoring. 30
- CNN** *Convolutional Neural Network*. 26
- CPA** *Closest Point of Approach*. 27, 33
- CTI** Centro de Terapia Intensiva. 4
- CV** Coeficiente de Variação. 28, 33
- DF** *Directory Facilitator*. 19
- DL** *Deep Learning*. 30, 32, 33, 66
- DM** *Data Mining*. 29, 56
- DP** desvio padrão. 56
- DT1** Diabetes Tipo 1. 29, 30
- DT2** Diabetes Tipo 2. 30, 32, 33
- EA** *Ensemble Algorithms*. 29, 33

FIPA *Foundation for Intelligent Physical Agents*. 19, 20, 44, 48

FL *Fuzzy Logic*. 26, 31, 33, 66

GE *Grammatical Evolution*. 29

GREI *Grupo de Redes Eléctricas Inteligentes*. 20

GRU *Gated Recurrent Unit*. 30, 33

GSM *Glicose Scoring Metric*. 28, 33

IA *Inteligência Artificial*. x, 1, 2, 16, 22, 25–33, 66

IdC *Internet das Coisas*. 45, 66

IFL *Intuicionist Fuzzy Logic*). 29, 33

IFS *IntuicionistFuzzy System*. 28, 33

IMC *Índice de Massa corporal*. 47, 55, 57, 59

IMT *Ideal Medical Technologies*. 28, 33

iStar *Intentional Strategic Actor Relationships modelling*. 13, 16

IV *intravenosa*. 9–11

JADE *Java Agent Development framework*. 19, 22

JRE5 *Java Run Time 5*. 19

JSON *JavaScript Object Notation*. 22, 41–43

JVM *Java Virtual Machine*. 20

KB *Knowledge Based*. 17

KBS *Knowledge Based Systems*. 17, 20

KNN *K-Nearest Neighbors*. 26

LSTM *Long-Short Term Memory*. 30, 33

MAE *Mean Absolute Error*. 22, 23, 58

MAPE *Mean Absolute Percent Error*. 22, 23, 30, 33, 58

MAS4GC *Multiagent System for Glicemic Control*. 32–35, 37, 42–45, 54, 59, 61–66

MCC *Matthews Coeficient Correlation*. 29, 33

Med *média*. 56

MEWS *Modified Early Warning Score*. 28, 33

MIMIC *Medical Information Mart for Intensive Care*. 54

ML *Machine Learning*. 16, 29, 30, 33

MLP *Multi-layer Perceptron classifier*. 30

MQO *Mínimos Quadrados Ordinários*. 17

MTS *Message Transport Service*. 19

NLP *Natural Language Processing*. 26

NN *Neural Networks*. 16, 66

NoSQL *Not Only SQL*. 42

NPH *Neutral Protamine Hagedorn*. 11

OLS *Ordinary Least Squares*. 17

PAA *Patient Analyzer Agent*. 35, 36, 40, 44, 48–51

PADE *Python Agent DEvelopment framework*. 19, 20, 22, 44, 45, 48, 49

PAGE *Perceptions, Actions, Goals, Environment*. 12

PEAS *Performance, Environment, Actuators, Sensors*. 12, 34, 35

POA *Programas Orientados a Agentes*. 12, 13

PTA *Propose Treatment Agent*. 35, 40, 44, 45, 48–50, 53

QP *Questões de Pesquisa*. 3

RBR *Rule-Based Reasoning*. 27, 31–33, 54

RF *Random Forest*. 26

RLM *Regressão Linear Múltipla*. 17, 50, 54, 57–59

RLS *Regressão Linear Simples*. 17, 54, 57–59

RMSE *Root Mean Square Error*. 22, 23, 30, 33, 58

RNN *Recurrent Neural Network*. 30, 32, 33

RT-CGM *Real Time-Continuous Glucose Monitoring*. 9

SC *subcutânea*. 10, 11, 63

SGBD *Sistema Gerenciador de Banco de Dados*. 42, 45

SMA Sistema Multiagente. 1–4, 12, 16, 19, 25–34, 36, 44, 45, 48, 64, 65

SMCG Sistema de Monitoramento Contínuo da Glicose. 9

SVM *Support Vector Machine*. 16, 26, 29, 30, 66

UML *Unified Modeling Language*. 12, 13, 16, 34, 36, 40, 41

UTI Unidade de Terapia Intensiva. 1–10, 25–34, 36, 37, 44, 45, 47, 54–56, 60–62, 64, 65

UTI-A Unidade de Terapia Intensiva - Adulto. 4

UTI-N Unidade de Terapia Intensiva - Neonatal. 4, 25, 26, 28, 29, 33

UTI-P Unidade de Terapia Intensiva - Pediátrica. 4

UTI-Pm Unidade de Terapia Intensiva - Mista. 4

UTIs Unidades de Terapia Intensiva. 4

Capítulo 1

Introdução

Aplicações de Inteligência Artificial (IA), combinadas com os recursos tecnológicos atuais, têm grande potencial de permitir a criação e a prestação de serviços de gerenciamento em diversas áreas. Inclusive em áreas críticas como a da saúde, a qual lida com doenças crônicas e pacientes em estados graves. Especialmente em Unidade de Terapia Intensiva (UTI), onde esses casos ocorrem com maior frequência (Contreras and Vehi, 2018).

A UTI é considerada um cenário assistencial de alto risco, onde descuidos ou erros médicos podem causar mortes ou complicações a saúde dos pacientes. O acompanhamento de pacientes em UTI pode ser feito com auxílio de protocolos, *checklists* e até mesmo com a aplicação do FAST HUG. Esse último é um mnemônico simples e significativo que destaca sete das principais áreas que devem ser acompanhadas pela equipe de saúde para cada paciente internado na UTI. Segundo Vincent (2005), o FAST HUG tem a finalidade de minimizar os possíveis problemas no acompanhamento de pacientes em UTI incluindo sete áreas: *Feeding* (alimentação), *Analgesia* (analgesia), *Sedation* (sedação), *Thromboembolic prevention* (prevenção tromboembólica), *Head of the bed elevated* (cabeceira elevada), *stress Ulcer prophylaxis* (prevenção de úlcera por estresse) e *Glucose control* (controle glicêmico).

Relacionado às áreas do FAST HUG, pode-se destacar o controle glicêmico (*glucose control*), o qual tem como principal função verificar o nível de glicose no sangue do paciente. Apesar de Cooksley et al. (2018) recomendar a glicemia ideal (faixa alvo) entre 140 a 180 mg/dL, alguns autores, inclusive o médico intensivista envolvido nesse trabalho, dizem que esse valor pode variar dependendo do hospital no qual é aplicado. Independente de qual é o valor ideal, o mais importante é manter o acompanhamento do nível de glicose no sangue do paciente. Segundo DeJournett and DeJournett (2017), o controle eficaz da glicemia no ambiente da UTI tem o potencial de diminuir as taxas de morbimortalidade e o tempo de permanência do paciente nas unidades intensivas, conseqüentemente otimizando os recursos empregados na área de saúde.

Para potencializar a eficiência do FAST HUG, a automatização dos processos aliados ao uso de recursos de IA tem se mostrado muito atraente (Dankwa-Mullan et al., 2019). Considerando que um Sistema Multiagente (SMA) é uma abordagem inteligente distribuída, adequada para aplicações modulares, mutáveis e complexas, onde os agentes possuem características como autonomia deliberativa, integração, reatividade e flexibilidade, torna-se uma solução interessante para sistemas de saúde de grande escala. Um SMA é formado por um conjunto de entidades ou agentes autônomos que interagem para

alcançar objetivos individuais ou coletivos (Darabi et al., 2015).

Durante a revisão da literatura foi possível encontrar trabalhos que utilizam sistemas com técnicas de IA para o controle glicêmico em UTI (DeJournett and DeJournett (2016); DeJournett and DeJournett (2017); DeJournett et al. (2020)). Outros trabalhos aplicam SMA para o controle glicêmico (Darabi et al. (2015)) e também no contexto de UTI (Jemal et al. (2019); Malak et al. (2019)). Encontrou-se ainda trabalhos que aplicam modelos de predição como solução para o controle glicêmico (Vehí et al. (2020); Bertachi et al. (2020); Kim et al. (2020)). Porém nenhum dos trabalhos pesquisados contemplam a utilização de SMA que combine modelos de predição com técnicas de IA para o controle glicêmico de pacientes internados em UTI.

1.1 Problema e motivação

Segundo Chase et al. (2019), melhorar a produtividade, o custo, o atendimento, e o tratamento de pacientes internados em UTI são os principais desafios atuais no gerenciamento hospitalar, onde a personalização e automação do atendimento oferecem oportunidades de causar impactos significativos.

Relacionado aos pacientes de UTI, o controle glicêmico, envolvendo alta ou baixa quantidade de glicose no sangue, hiperglicemia e hipoglicemia, respectivamente, é um problema frequente. Desta forma, esses pacientes estão sujeitos a sofrer danos à saúde pelo descontrole da glicemia. Além disso, é importante ressaltar que alguns fatores contribuem para a elevação da glicose no sangue em situações de doenças críticas, como a descompensação do diabetes mellitus pré-existente e dextrose na nutrição parenteral ou em fluidos antibióticos (Cooksley et al., 2018).

Estudos demonstram que o controle glicêmico intensivo reduz a falência múltipla de órgãos, infecções sistêmicas, o tempo de permanência do paciente em hospitais ou UTI, e até mesmo casos de mortalidade em pacientes críticos, ressaltando a necessidade de um controle rigoroso (Braga et al., 2015).

Outros estudos, como o elaborado por Liu et al. (2020), ressaltam a importância do controle glicêmico de pacientes acometidos pela COVID-19, principalmente os que necessitam ser internados em UTI. Onde, durante a internação, 89,7% dos pacientes necessitaram receber tratamento glicêmico, incluindo insulina ou hipoglicemiantes orais. Os autores destacam que o uso de doses adequadas, monitoramento rigoroso da glicose e normalização oportuna dos níveis de glicose no sangue são essenciais durante o tratamento contra a COVID-19.

Em tempos de pandemia, como o da COVID-19, ficou ainda mais evidente a importância da UTI no tratamento de pacientes críticos. Pois estes necessitam de cuidados especiais, que incluem além de alguns requisitos básicos, a necessidade de equipamentos para monitorização do paciente e suporte de vida, como por exemplo, os respiradores e ventiladores mecânicos (Noronha et al., 2020). A sobrecarga dos profissionais de saúde nesse período também esteve em evidência. Fatores como os apontados ressaltam a importância de sistemas informatizados que automatizem o monitoramento e auxiliem no tratamento de pacientes, trazendo benefícios especialmente em situações pandêmicas como a atual.

1.2 Objetivos

Este trabalho tem como objetivo principal o desenvolvimento de uma solução baseada em agentes inteligentes capaz de fazer o monitoramento e tratamento do controle glicêmico de pacientes internados em UTI. Para alcançar o objetivo descrito elenca-se os seguintes objetivos específicos:

- propor um modelo de SMA utilizando a modelagem Tropos, bem como definir o projeto arquitetural do sistema.
- implementar o SMA projetado utilizando modelos de predição e mecanismo de inferência para representar o conhecimento de um especialista em cuidados intensivos.
- determinar a acurácia da solução desenvolvida através de experimentos utilizando dados sintéticos de pacientes comparados com a atuação humana e avaliados utilizando métricas estatísticas.

1.3 Hipótese e questões de pesquisa

Durante a realização deste trabalho assume-se como hipótese que uma solução baseada em agentes inteligentes pode fazer recomendações de monitoramento e tratamento de pacientes internados em UTI utilizando modelos de predição e uma base de conhecimento com regras de inferência baseadas no conhecimento de médicos intensivistas. No entanto, consideramos uma restrição deste trabalho o fato de ter sido utilizado o conhecimento especializado de apenas um médico intensivista especialista por falta de disponibilidade médica em época de pandemia.

Como Questões de Pesquisa (QP) investigativa foi definido:

- QP1 - um sistema baseado em agentes é capaz de fazer predições de glicemia utilizando modelos preditivos para auxiliar os profissionais de saúde no acompanhamento de pacientes internados em UTI?
- QP2 - um sistema baseado em agentes pode fazer recomendação de monitoramento e tratamento glicêmico a pacientes de UTI utilizando bases de conhecimento com regras de inferência, definidas com auxílio de um médico intensivista especialista?

1.4 Organização do manuscrito

O restante do documento está organizado da seguinte forma. No Capítulo 2 é apresentada a fundamentação teórica incluindo conceitos relacionados ao gerenciamento de UTI, controle glicêmico, técnicas de desenvolvimento de SMA, e técnicas de avaliação de modelos estatísticos. No Capítulo 3 é apresentada a revisão da literatura incluindo uma taxonomia e uma tabela comparativa dos principais trabalhos pesquisados. No Capítulo 4 é apresentada a proposta de solução com pré-projeto, modelagem, arquitetura e implementação do sistema para controle glicêmico de pacientes internados em UTI. No Capítulo 5 são descritos os experimentos realizados com uma discussão de resultados. As conclusões, publicações e trabalhos futuros são apresentados no Capítulo 6.

Capítulo 2

Fundamentação Teórica

Neste capítulo são apresentados os conceitos relacionados a proposta de solução. Considerando a área médica são abordados os conceitos de gerenciamento de UTI, incluindo o mnemônico FAST HUG e mais especificamente o controle glicêmico. Também são apresentados conceitos referentes a SMA com foco na modelagem e raciocínio dos agentes, bem como as tecnologias utilizadas para implementação da solução. Finalmente, serão apresentadas técnicas utilizadas na avaliação dos resultados alcançados pela solução desenvolvida.

2.1 Gerenciamento de UTI

Segundo a Agência Nacional de Vigilância Sanitária (Anvisa), a UTI é uma área crítica destinada à internação de pacientes graves, que requeiram atenção profissional especializada de forma contínua, materiais específicos e tecnologias necessárias ao diagnóstico, monitoração e terapia (Anvisa, 2010). A UTI deve estar localizada em um hospital, sendo que o agrupamento de UTI em uma mesma área física é chamado de Centro de Terapia Intensiva (CTI). As UTIs podem ser divididas por áreas de acordo com a faixa etária do paciente: Adulto (UTI-A), Pediátrica (UTI-P), Neonatal (UTI-N) e Mista (UTI-Pm). A UTI-Pm é destinada à assistência de pacientes recém-nascidos e pediátricos na mesma sala. As UTI podem ser divididas também de acordo com a especialidade, que é o caso das unidades destinadas à assistência a pacientes selecionados por tipo de doença ou intervenção, como cardiopatas, neurológicos, cirúrgicos, entre outras.

Toda UTI é composta por diversos profissionais formando uma equipe multidisciplinar que normalmente contém médico diarista/rotineiro (intensivista), médico plantonista, enfermeiro assistencial, fisioterapeuta, técnicos de enfermagem, além de auxiliares administrativos e funcionários exclusivos para serviços de limpeza da unidade. Vale ressaltar que os médicos plantonistas, enfermeiros assistenciais, fisioterapeutas e técnicos de enfermagem devem estar disponíveis em tempo integral para assistência aos pacientes internados de acordo com suas respectivas escalas de trabalho (Anvisa, 2010).

A assistência prestada, evolução do estado clínico, intercorrências, os cuidados prestados e critérios para admissão e alta de pacientes na UTI devem ser registrados pelas equipes médicas, de enfermagem e de fisioterapia no prontuário do paciente. Esses registros devem ser feitos em cada turno, de forma legível, contendo o número de registro no respectivo conselho de classe profissional e atendendo as regulamentações dos conselhos

de classe, além das normas institucionais. O prontuário do paciente deve ser datado e conter os registros assinados pelo responsável técnico e divulgados para toda a instituição (Anvisa, 2010).

Pode-se dizer que em um ambiente hospitalar, a UTI é uma das unidades mais complexas de ser gerenciada, pois é o local onde se concentram os pacientes com estado crítico e que merecem maiores cuidados. Tais pacientes são difíceis de gerenciar com segurança e eficácia mediante a sua fisiologia complexa, não-linear, altamente variável e com resposta não-determinística e heterogênea à terapia. Fato esse que acarreta diretamente no aumento dos custos aos cuidados intensivos, reduzindo a capacidade dos sistemas de saúde de manter a equidade no acesso aos cuidados desses pacientes. Sendo assim, melhorar a produtividade e o custo, conseqüentemente o atendimento e tratamento dos pacientes são os principais desafios atuais para o atendimento na UTI, onde a personalização e automação do atendimento oferecem oportunidades de causar impactos significativos (Chase et al., 2019).

Com o intuito de melhorar a qualidade do atendimento a pacientes e até mesmo minimizar a ocorrência de erros médicos em UTI, mecanismos como protocolos, listas de verificação e rodadas médicas a beira leito são indicadas no acompanhamento dos pacientes e devem ser feitos tanto pelo médico intensivista quanto por qualquer outro profissional de saúde atuante no ambiente, como enfermeiros e fisioterapeutas (Vincent, 2005).

2.1.1 FAST HUG

Os mnemônicos são comumente usados em procedimentos médicos como auxiliares cognitivos para orientar os profissionais de saúde em todo o mundo. Vincent (2005) sugeriu o mnemônico FAST HUG, que apesar de ser simples e curto contém sete aspectos essenciais que devem ser verificados durante o atendimento de pacientes críticos nas UTI, mais especificamente durante as rodadas médicas (visitas diárias que os profissionais fazem aos pacientes internados em seus respectivos leitos). O fato de ser um mnemônico ajuda o profissional a se lembrar dos itens essenciais que devem ser verificados dispensando uma lista de checagem. Tais verificações devem acontecer pelo menos uma vez ao dia, e idealmente sempre que algum profissional atender o paciente.

Vincent (2005) defende cada um dos sete itens que compõe o mnemônico FAST HUG:

- por alimentação (*feeding*) deseja-se evitar a desnutrição do paciente durante sua internação. Sabendo que alguns pacientes já chegam desnutridos na UTI, não podem se alimentar sozinhos, ou passam por alguma dificuldade para se alimentar, em decorrência da sua enfermidade necessitando de alimentação não oral. Sendo assim a revisão diária da sua alimentação possibilita o suporte nutricional adequado.
- a analgesia (*analgesia*) é importante para aliviar a dor dos pacientes, pois estes estão sujeitos a diversos procedimentos de rotina que podem sofrer durante a sua internação como curativos, manipulações, entre outros. Vale ressaltar que a dor pode afetar a recuperação fisiológica e psicológica do paciente e seu alívio é indispensável. Toda administração de medicação pode causar efeitos colaterais, portanto deve-se tomar cuidado para garantir que a analgesia seja adequada e não excessiva.

- relacionado à medicação, a sedação (*sedation*) é fundamental em pacientes de UTI para reduzir o estresse, mantê-los calmos e silenciosos, colaborando com os procedimentos médicos e consequentemente auxiliando a recuperação. É importante salientar que a administração do sedativo varia para cada paciente e que em casos de coma a sedação pode ser contínua ou seguir algumas escalas. Pode passar por algumas interrupções transitórias diárias, diminuindo a frequência e dosagem gradativamente até que não seja mais necessária, consequentemente desocupando o leito de UTI e reduzindo custos.
- a tromboembolia (*thromboembolic*) é mais frequente em pacientes com traumas e a sua prevenção pode evitar a trombose venosa profunda. Este é um dos fatores que pode acarretar mortalidade de pacientes, principalmente por ser um dos aspectos que são frequentemente esquecidos e consequentemente não realizada pelos profissionais.
- cabeceira elevada (*head of the bed elevated*), como o próprio nome sugere, consiste em manter a cabeceira da cama e principalmente o tórax do paciente elevados a 45 graus. Pode ser que alguns pacientes, principalmente os que estejam sedados deslizem pela cama, mas a elevação evita que a alimentação recebida seja “devolvida”. Esse procedimento simples diminui a possível incidência de refluxo gastroesofágico e pneumonia nosocomial, entre outros.
- a profilaxia de úlceras por estresse (*stress ulcer prophylaxis*) é importante, principalmente em pacientes com insuficiência respiratória ou alguma anormalidade de coagulação, submetidos à terapia aerodinâmica ou com histórico de úlcera gastroduodenal, com risco aumentado de desenvolver hemorragia gastrointestinal relacionada ao estresse.
- o controle glicêmico (*glucose control*) deve garantir que o nível de glicose do paciente seja rigorosamente controlado, pois se atingir picos (tanto alto quanto baixo) pode trazer graves prejuízos ao paciente e inclusive acarretar a morte.

FAST HUG é o mnemônico mais comumente usado para auxiliar os profissionais de saúde em recordar os procedimentos a serem verificados em pacientes de UTI. Porém houveram vários acréscimos e modificações a este, para se adequar a diferentes subgrupos de pacientes. Alguns deles são FAST HUGS BID (que acresce: *Spontaneous breathing trial, Bowel regimen, Indwelling catheters and lines e Deescalate antibiotics*) e FAST HUG MAIDENS (*Medication reconciliation, Antibiotics or anti-infectives, Indications for medications, Drug dosing, Electrolytes, hematology, and other laboratory results, No drug interactions, allergies, duplications, side effects e Stop dates*), por exemplo, com propósitos específicos. Inclusive um novo mnemônico FAST HUGS BID está sendo proposto como ONE FAST HUGS BID COVID. Sua finalidade é facilitar a avaliação diária, bem como as decisões de tratamento para pacientes criticamente enfermos com COVID-19. Outros fatores do cuidado crítico são essenciais para os pacientes com COVID-19 envolvendo questões de respiração assistida, suporte de oxigênio, ventilação mecânica, práticas de controle de infecção, entre outros (Kumar et al., 2020).

Quanto à aplicação do FAST HUG pode-se destacar três características importantes:

1. não é restrito a nenhum grupo específico de pacientes, podendo ser aplicada a todos internados na UTI;

2. o mnemônico é de grande abrangência e inclui aspectos fundamentais de cuidado que envolvem todos os membros da equipe, sendo curto o suficiente para ser facilmente lembrado; e
3. tem um toque pessoal, pois todos gostam de um abraço e os pacientes não são exceção.

Enfim, o FAST HUG pode auxiliar na qualidade do atendimento recebido pelos pacientes nas UTI (Vincent, 2005).

2.1.2 Controle glicêmico

Basicamente o controle glicêmico diz respeito a quantidade de glicose (açúcar) que o paciente possui no sangue. A glicose é um carboidrato de cadeia mais simples (monosacarídeos) e também a principal fonte de energia do organismo, sendo essencial para o bom funcionamento do corpo humano. Quando a ingestão de glicose no organismo utiliza o necessário para suas atividades e armazena o excedente, acaba ocasionando o aumento da concentração de glicose no sangue. Ou seja, o organismo armazena o excedente na forma de glicogênio que fica no fígado e nos músculos estriados esqueléticos. Em geral, se o paciente não for diabético, o corpo nunca deixa a concentração de glicose passar de 180mg/dL (Moraes, 2020).

O responsável pelo controle de glicose no organismo é o pâncreas. Com a alta concentração de glicose, o pâncreas começa a produzir um hormônio responsável por encaminhar as moléculas de glicose do sangue para dentro das células musculares, reduzindo a concentração de glicose no sangue, tal hormônio é chamado de insulina. Caso a concentração de glicose no sangue esteja baixa, o pâncreas produz outro hormônio responsável por pegar a glicose estocada e encaminhá-la para o sangue, aumentando a concentração de glicose, esse é chamado de glucagon (Moraes, 2020).

A glicemia alta (hiperglicemia) é predominante nos cuidados intensivos e é um bom exemplo de que o controle rigoroso de glicemia desses pacientes podem ter grande impacto. Tal episódio acontece pela produção não suprimida de glicose pelo organismo, medicamentos, supressão da secreção de insulina do corpo, e perda de sensibilidade a insulina. Todos esses fatores danificam efetivamente os mecanismos normais de controle de *feedback* do corpo, resultando em menor captação de glicose mediada por insulina, causando a hiperglicemia. Portanto, é necessário algum tipo de controle glicêmico suplementar, se possível em circuito fechado e altamente personalizado em cuidados intensivos (Chase et al., 2019).

O trabalho de Vincent (2005) aborda principalmente dois artigos que exaltam a importância do controle glicêmico em UTI. O primeiro é um estudo realizado por Krinsley (2004) que demonstra que manter os níveis de glicose no sangue em aproximadamente 140 mg/dL resulta em uma diminuição de 29,3% nas taxas de mortalidade hospitalar e uma redução de 10,8% no tempo de permanência na UTI. O segundo estudo realizado por Berghe et al. (2001) inclui principalmente pacientes cirúrgicos com altos índices de gravidade, porém com taxa de mortalidade relativamente baixa. Devido aos resultados satisfatórios tal estudo fez com que os médicos dessem mais atenção ao controle glicêmico fazendo até mesmo que alterassem as práticas médicas aderindo ao controle rigoroso da glicose de seus pacientes.

Tendo em vista que o controle glicêmico é um aspecto importante para a saúde e a vida do paciente crítico, principalmente em relação às taxas de morbimortalidade dos pacientes internados em UTI, esse item foi escolhido para ser estudado nesta pesquisa. No entanto, no futuro, a solução desenvolvida neste trabalho pode ser ampliada para incluir os demais itens do FAST HUG. Em relação ao controle glicêmico, tanto o monitoramento ou a frequência de coletas de sangue para aferição da glicemia, quanto o nível glicêmico ideal ou o intervalo de valores mínimo e máximo que devem ser mantida a glicemia (faixa alvo ou normoglicemia) variam na literatura de acordo com a região e até mesmo entre instituições hospitalares.

Faixa glicêmica alvo

Este trabalho teve a colaboração do médico especialista em medicina intensiva Dr. Sérgio Eduardo Soares Fernandes, que além de acompanhar a pesquisa, forneceu embasamento teórico e conhecimento técnico na área médica. Dr. Sérgio é graduado em medicina pela Universidade Federal do Piauí (1999), pós-graduado em clínica médica, terapia intensiva, geriatria e gestão (2002-2013), pós-graduado em epidemiologia para gestão pela *Johns Hopkins Bloomberg School of Public Health* (2013), mestre em Saúde Coletiva/Epidemiologia pela Universidade de Brasília (2015). Atualmente é docente na Escola Superior de Ciências da Saúde no Distrito Federal e consultor independente de qualidade em serviços de saúde¹.

De acordo com o especialista a glicemia pode ser mensurada por duas unidades de medida: mg/dL (concentração mássica) ou mmol/l (concentração molar), que significam respectivamente miligramas por decilitro e milimoles por litro. A conversão entre as unidades pode ser facilmente calculada pelos seguintes fatores de conversão: 1 mg/dL = 0.0555 mmol/L ou 1 mmol/L = 18,018 mg/dL. Nesse trabalho serão considerados os valores em mg/dL.

Segundo Cooksley et al. (2018), a terapia com insulina é recomendada para hiperglicemia persistente, onde os valores são maiores que 180 mg/dL. E depois que a aplicação de insulina é iniciada, recomenda-se um intervalo alvo de glicose de 140 a 180 mg/dL para a maioria dos pacientes gravemente enfermos. Casos de hipoglicemia são considerados quando a glicemia está abaixo de 70 mg/dL. No passado, várias diretrizes recomendavam a faixa alvo em 80-110 mg/dL, mas em casos de insulinoterapia intensiva (em UTI), pelo tratamento contínuo com insulina, vários estudos demonstraram o aumento do risco de hipoglicemia em comparação com a terapia convencional.

De acordo com Braga et al. (2015), atualmente é proposto que em pacientes internados em UTI os índices aceitáveis de glicemia são entre 140 a 180 mg/dL. O início da aplicação de insulina intravenosa inicie assim que os valores ultrapassem os 180 mg/dL. Além disso, devem ser evitadas as glicemias abaixo de 100 mg/dL, definindo a hipoglicemia em valores menores que 70 mg/dL e hipoglicemia grave para valores menores que 40 mg/dL.

Apesar dos trabalhos de Cooksley et al. (2018) e Braga et al. (2015) apresentarem valores semelhantes, outros trabalhos apontam faixas diferentes. Em DeJournett and DeJournett (2016), por exemplo, os índices aceitáveis de glicemia são entre 70 e 140 mg/dL, variando de acordo com as refeições, onde considera-se 70-130 mg/dL antes da consumação de alimentos e 90-180 mg/dL depois. Darabi et al. (2015) ressalta que outros

¹<http://lattes.cnpq.br/9797758799188189>

fatores podem influenciar os índices aceitáveis de glicemia tais como a alimentação e o diabetes, entre outros.

Monitoramento e coleta

Sabe-se que a glicemia pode ser medida de acordo com a concentração de açúcar presente no sangue, logo a coleta de amostras de sangue do paciente se faz necessária. Em UTI o monitoramento comumente é feito por meio de coletas de glicemia capilar, bioquímica ou até mesmo automatizada (SDB, 2015):

- o sangue capilar é obtido através de punção transcutânea, com maior proporção de sangue arterial do que sangue venoso. Normalmente essa punção é feita com um pequeno furo na ponta do dedo, também conhecido como AutoMonitoramento da Glicemia Capilar (AMGC).
- a bioquímica é feita no laboratório através de reações químicas automatizadas. O princípio da reação é a oxidação da glicose da amostra sob a ação catalisadora da enzima glicose-oxidase presente no reagente. O produto final é uma substância vermelha cuja intensidade da cor é proporcional à concentração de glicose na amostra. Esse método requer muito volume de sangue para análise, além de tempo e custo para a obtenção dos resultados.
- um Sistema de Monitoramento Contínuo da Glicose (SMCG) consiste em um sensor implantado no tecido subcutâneo, que transmite informações a um aparelho monitor. Este método permite medir continuamente a glicose no líquido intersticial, o que pode identificar tendências do perfil glicêmico que não tenham sido identificadas pela coleta AMGC.

Para uma melhor efetividade no monitoramento da glicemia é importante que as coletas sejam feitas em um intervalo adequado. Cooksley et al. (2018) recomenda o monitoramento da glicemia beira leito em UTI de acordo com a situação do paciente:

- se o paciente estiver fazendo refeições, faça o monitoramento da glicose antes das refeições para combinar com a ingestão de alimentos;
- se o paciente não estiver fazendo refeições, faça o monitoramento da glicose a cada 4-6 horas;
- se o paciente estiver recebendo insulina intravenosa (IV), realize o monitoramento da glicose com mais frequência, a cada 30 minutos-2 horas.

Cooksley et al. (2018) complementa que o ajuste das doses de insulina devem ser feitas conforme necessário para evitar hipoglicemia. Porém o monitoramento contínuo da glicose em tempo real (*Real Time-Continuous Glucose Monitoring* (RT-CGM)) e autônomo, em UTI ou na sala de cirurgia não é recomendado (recomendação forte, evidência de qualidade muito baixa). Outra recomendação é a cautela ao interpretar resultados de medidores de ponto de atendimento, pois as medidas de glicose no plasma e no sangue total diferem em relação aos medidores de glicemia. Além disso, níveis anormais de hemoglobina, hipoperfusão, bilirrubina elevada ou ácido úrico e alguns medicamentos podem resultar em discrepâncias significativas entre as amostras de plasma capilar, venoso e arterial.

Valores utilizados no trabalho

Considerando a falta de padrão entre os valores que regulam a glicemia de pacientes em UTI, este trabalho combina valores encontrados na literatura (DeJournett and DeJournett (2017) e Cooksley et al. (2018)). Com base nisso, a princípio foi elaborada uma tabela contendo a escala glicêmica, em seguida os valores dessa tabela foram comparados com as respostas (indicações de monitoramento e tratamento) de cinco profissionais da área médica durante a realização do Experimento 2 (Seção 5.3). Com auxílio do especialista alguns valores foram ajustados até se obter os valores disponíveis na Tabela 2.1. Vale ressaltar que tais valores podem variar de acordo com instituições hospitalares.

Conforme apresentado na Tabela 2.1, cada faixa de valor glicêmico do paciente é representado por um código relacionado aos episódios. Episódio significa em que situação o paciente se encontra em relação a sua glicemia, referindo-se a uma faixa de valor que representa a quantidade de açúcar que o paciente possui no seu sistema sanguíneo, utilizando a unidade mg/dL. Esses códigos serão utilizados na representação do conhecimento dos agentes e nas variáveis de implementação do sistema.

Tabela 2.1: Escala de valores glicêmicos.

Código	Episódio	Valores (mg/dL)
hipoG	Hipoglicemia Grave	0 até 40
hipoL	Hipoglicemia Leve	> 40 até 70
gAlvo	Normoglicemia (Faixa Alvo)	> 70 até 140
hiperL	Hiperglicemia Leve	> 140 até 180
hiperG	Hiperglicemia Grave	> 180 até 250
hiperGG	Hiperglicemia Gravíssima	acima de 250

A Figura 2.1 ilustra a escala de valores de cada um dos episódios glicêmicos em relação aos valores das glicemias em mg/dL conforme apresentado na Tabela 2.1.

Basicamente para o controle glicêmico existem dois procedimentos a serem realizados: monitoramento e tratamento. Monitoramento refere-se a periodicidade que o sangue do paciente deve ser coletado no dia, fato esse que viabiliza o acompanhamento da glicemia do paciente. Para o tratamento, é necessário saber qual droga (insulina ou glicose) deve ser aplicada, por qual via, intravenosa (IV) ou subcutânea (SC), bem como a quantidade/dose. Na Tabela 2.2 além do código indicando os episódios glicêmicos é apresentado para cada episódio qual a periodicidade das coletas no dia e a indicação de tratamento recomendado.

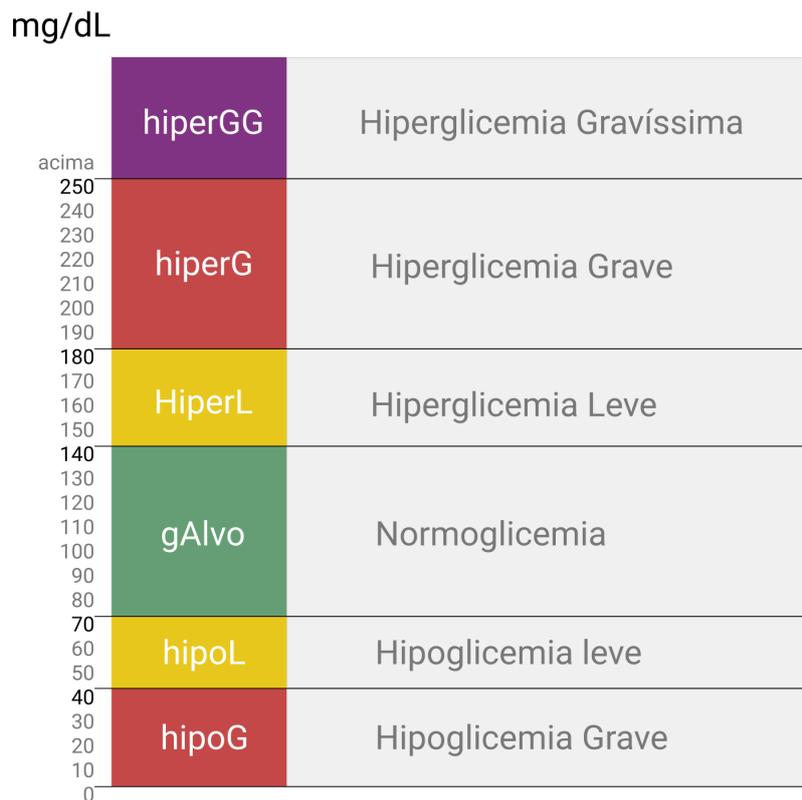


Figura 2.1: Escala de valores glicêmicos.

Tabela 2.2: Indicações de monitoramento e tratamento.

Código	Monitoramento (intervalo de coleta)	Tratamento (recomendação de aplicação)
hipoG	de 1h a 2h	4 ampolas de glicose a 50% IV
hipoL	de 2h a 4h	2 ampolas de glicose a 50% IV
gAlvo	de 6h a 8h	manter observação
hiperL	de 4h a 6h	2 unidade de insulina regular SC
hiperG	de 2h a 4h	4 unidade de insulina regular SC
hiperGG	em 1h	6 unidade de insulina regular SC

De acordo com o especialista, para a coleta além de data e hora que a coleta da glicemia foi realizada é necessário saber o valor da glicemia, o tipo da coleta (bioquímica ou capilar), além de dados complementares (e.g., tipo de alimentação) e a quantas horas o paciente recebeu sua última alimentação. Quanto às aplicações é necessário registrar: data e hora, tipo (de resgate ou de horário), via de administração (IV, SC ou oral), a droga (insulina - NPH (*Neutral Protamine Hagedorn*), insulina lenta - Detemir, insulina ultra-lenta - Glargina, insulina ultra-rápida, insulina regular, glicose a 50% e glicose a 25%) e a posologia. Em relação às doses, cada unidade de insulina diminui 30 mg/dL o valor glicêmico do paciente, e cada ampola de glicose a 25% aumenta em 50 mg/dL a glicemia.

2.2 Sistema multiagente

Um SMA é composto por dois ou mais agentes capazes de perceber eventos no ambiente utilizando sensores, raciocinar e agir no ambiente por intermédio de atuadores. Os agentes de um SMA podem desempenhar diversos papéis para atingir objetivos através da execução autônoma de ações, se adaptando a mudanças do ambiente e interagindo com outros agentes. As ações serão escolhidas de acordo com as condições percebidas no ambiente onde o agente está inserido. A inteligência do agente se dá pela escolha de ações mais adequadas para alcançar os objetivos individuais ou coletivos (Russell and Norvig, 2010).

Segundo Wooldridge (2009), para que um agente seja considerado inteligente ele deve ser autônomo, reativo, pró-ativo e possuir habilidades sociais. Ou seja, um agente é uma entidade computacional que está inserido em um ambiente, sendo capaz de agir de forma flexível e autônoma a fim de alcançar os objetivos definidos no projeto.

Para Russell and Norvig (2010), os agentes podem ser classificados de acordo com sua atuação no ambiente através de seu comportamento: reativo simples, reativo baseado em modelo, baseado em objetivo, baseado em utilidade e com aprendizado. O ambiente no qual o agente está inserido pode ser classificado de acordo com características específicas do mesmo, bem como o acesso aos recursos disponíveis. A complexidade do projeto de um SMA está diretamente ligada às propriedades do ambiente, as quais podem ser: observável vs. parcialmente observável, estático vs. dinâmico, determinístico vs. não-determinístico, discreto vs. contínuo, e episódico vs. sequencial.

Um projeto de SMA inclui a identificação de quais são as percepções, ações, objetivos, performance e o ambiente de cada agente do sistema. O pré-projeto inclui essas definições através do acrônimo PEAS (Performance, Environment, Actuators, Sensors) ou PAGE (Perceptions, Actions, Goals, Environment). Como o próprio nome diz, o pré-projeto de SMA serve basicamente para identificar em qual ambiente o agente atuará e as respectivas características. Além disso, baseado nos objetivos de cada agente é possível descrever também quais são os mecanismos pelos quais os agentes perceberão as informações e como agirão em resposta a tais estímulos. Nesse sentido, um projeto de SMA necessita da modelagem do sistema para definir qual o comportamento e forma de raciocínio dos agentes, bem como quais ferramentas serão utilizadas na implementação (Russell and Norvig, 2010).

2.2.1 Modelagem de agentes

Existem diversas abordagens de modelagem de projetos orientado a agentes. A modelagem de Programas Orientados a Agentes (POA) necessita de ferramentas e técnicas de modelagem específicas, diferente da *Unified Modeling Language* (UML) por exemplo. Na modelagem POA podem ser analisados os estados mentais e as crenças dos agentes, em vez de estados de máquina; planos e ações, em vez de procedimentos; métodos de comunicação, negociação, interação e capacidade social, em vez de funcionalidades de entrada e saída; objetivos, desejos e assim por diante. Apesar disso, a UML pode contribuir na diagramação de fases relacionadas a implementação de um SMA por meio dos diagramas de classes, sequência, entre outros diagramas (Bresciani and Perini, 2004).

Bresciani and Perini (2004) citam que a metodologia de desenvolvimento de software Tropos, refere-se a modelagem orientada a objetivos (*goal modeling*), tendo sido desenvol-

vida especificamente para explorar a flexibilidade requerida pela modelagem de POA, que diferentemente da UML, concebe modelos no nível do conhecimento ao modelar programas orientados a objetivos e não somente em níveis de programação.

Modelagem Tropos

Tropos define uma metodologia de desenvolvimento de software para sistemas de software orientados a agentes baseada no framework i^* (*Intentional Strategic Actor Relationships modelling* (iStar)). Tropos permite modelar as funcionalidades de uma aplicação baseada em objetivos por meio de diagramas. Cada um dos diagramas, conforme ilustrado na Figura 2.2, possuem componentes visuais utilizados para representar graficamente atores, agentes e papéis (*Actor*, *Agent*, *Role*), objetivos (*Goal*, *Soft-Goal*), tarefas e planos (*Task/Plan*) e recursos (*Resource*) (Pimentel and Castro, 2018).

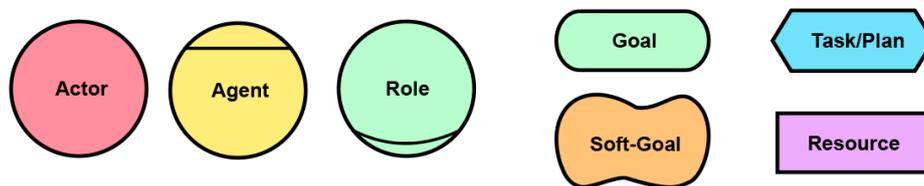


Figura 2.2: Componentes visuais do *framework I**.

Segundo Dalpiaz et al. (2016), os atores são ilustrados por círculos, que por sua vez representam outros sistemas ou pessoas envolvidas no ambiente a ser modelado e pode ter três significados diferentes. O círculo comum é usado para atores (*Actor*) que não tem uma representação específica, atores genéricos e sem especialização. Os agentes (*Agent*) também são representados por círculos, porém tem um detalhe em formato de linha reta na sua base superior indicando por exemplo um agente inteligente. Já os papéis (*Role*) além do círculo possui uma linha curva na parte inferior, representando um ator que possui algum tipo de especialização no ambiente em que está inserido.

O elemento *Goal* representa um objetivo ou subobjetivo que deverá ser alcançado pelo ator ou agente do sistema. Um estado de coisas que o ator deseja alcançar e que possui critérios claros de conquista (i.e., *hard goal*). Já o *Soft-Goal* representa uma capacidade ou habilidade que um ator ou agente possui em definir, escolher e executar um plano para o cumprimento de um objetivo. Pode representar também questões relacionadas a qualidade e desempenho.

O elemento *Task/Plan* representa as ações (tarefas ou planos) a serem executadas pelo ator ou agente para atingir seus objetivos. O recurso (*Resource*) representa uma entidade física ou informacional necessária ao ator ou agente para executar uma tarefa. Ou seja, o recurso que pode ser gerado ou utilizado pelos elementos envolvidos para viabilizar o alcance dos objetivos.

Além dos círculos que representam os atores, agentes e papéis, os ovais ilustram graficamente os objetivos, enquanto os *soft-goals* são representadas como formas mais curvas semelhantes a uma nuvem. As tarefas são representadas como hexágonos e os recursos são representados como retângulos. Geralmente esses elementos se relacionam entre si

por meio de inter-conexões. Cada conexão representa o nível de relacionamento existente entre os elementos conforme apresentado na Figura 2.3.



Figura 2.3: Elementos de conexão do *framework I**

Segundo Dalpiaz et al. (2016), quando dois elementos (e.g., *task* ou *goal*) estão ligadas a um *goal* com a associação do tipo *And*, indica que ambas devem ser executadas para que o objetivo seja atingido. Diferentemente de quando se utiliza a associação *Or*, onde basta que um dos elementos seja executado ou satisfeito. A *Qualification* serve para conectar os *Soft-Goals* aos elementos aos quais ela se refere. Graficamente uma ponta de seta em forma de T indica o refinamento *AND*, uma seta sólida direcionada ao elemento pai para representar o refinamento *OR* e uma linha tracejada representa uma *Qualification*.

A associação *is a* serve para representar que um ator é uma generalização/especialização de outros atores ou papéis generalizados deste. Já a *participates-in* representa qualquer tipo de associação entre dois atores, exceto generalização/especialização. Alguns exemplos de elementos e relacionamentos utilizando as associações apresentadas podem ser vistas na Figura 2.4. Essa figura apresenta um diagrama do modelo Tropos utilizando os elementos do *framework i**.

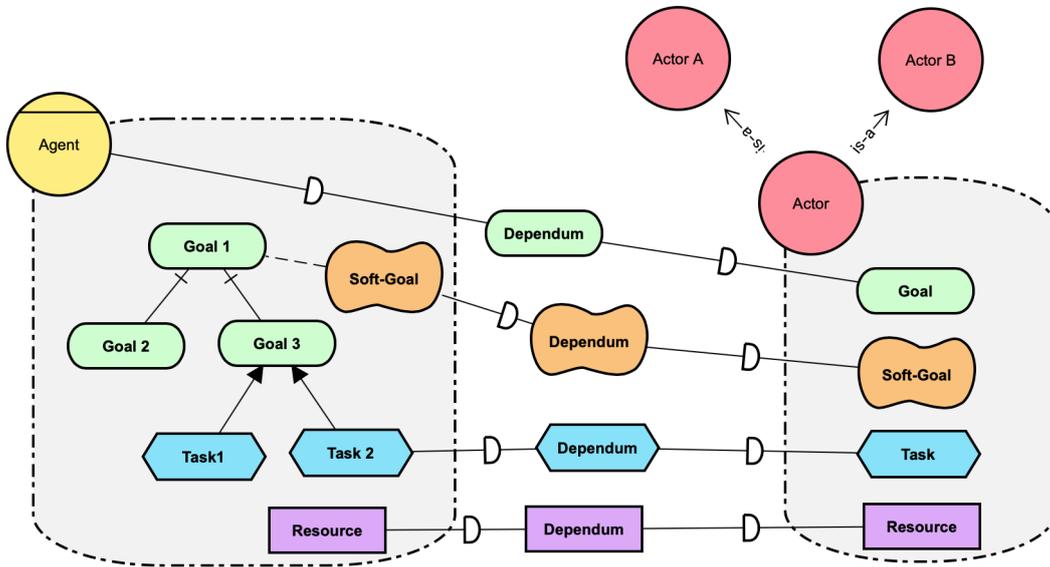


Figura 2.4: Diagrama ilustrativo de componentes e elementos do *framework i**

Além das associações, os atores, agentes e elementos podem ser interligados por meio das dependências, essas por sua vez podem ser *Goal Dependency*, *Task Dependency*, *Resource Dependency* e *Soft-Goal Dependency*, dependendo de quais informações eles trocam entre si. Cada *Agent* ou *Actor* possui uma área tracejada que serve para agrupar seus

respectivos elementos. Nessa área são encontrados os objetivos a serem alcançados, consequentemente as tarefas a serem executadas, os recursos que serão gerados ou utilizados e os *Soft-Goals* caso hajam (Dalpiaz et al., 2016). No exemplo da Figura 2.4 existe um *Goal Dependency* entre o *Agent* e um *Goal* de *Actor* representando que existe uma dependência de objetivo entre eles. Os demais tipos de dependência também podem ser observados neste exemplo. Além do respectivo elemento tal associação é representada por uma linha e uma letra “D” que significa dependência e indica a direção do elemento para onde a informação flui.

Um objetivo pode ter subobjetivos, como é o caso do *Goal* 1 que possui os subobjetivos *Goal* 2 e *Goal* 3. Note que os subobjetivos estão associados ao *Goal* 1 por meio de associações do tipo *And*, ou seja, ambas serão executadas. Os objetivos podem ser alcançados por meio de uma ou mais tarefas, como é o caso do *Goal* 3 que possui conectadas a ele as tarefas *Task* 1 e *Task* 2, por meio de associações do tipo *Or*, sendo então somente uma obrigatória. O objetivo *Goal* 1 possui um *Soft-Goal*, ou seja, necessita de uma capacidade extra para ser atingido. É possível também ver o exemplo de generalização/especialização utilizando as associações *is-a* entre o *Actor* e os atores *Actor* A e *Actor* B.

Fases da modelagem Tropos

Segundo Bresciani and Perini (2004), a metodologia Tropos engloba todas as cinco fases do desenvolvimento de software, visando dar apoio desde a análise de requisitos iniciais até a implementação. As fases incluem os requisitos iniciais, os requisitos finais, o projeto arquitetural, o projeto detalhado e a implementação:

- no diagrama de requisitos iniciais as partes interessadas do domínio são identificadas e modeladas como atores, estes por sua vez possuem dependência para atingir seus objetivos, podendo ou não compartilhar recursos. Nesta fase é possível declarar o porquê, o quê e como das funcionalidades do sistema.
- no diagrama de requisitos finais, o modelo anterior é estendido, incluindo novos atores, que representam o sistema a ser desenvolvido, e suas prováveis dependências com os demais atores já identificados no ambiente. Tais dependências permitirão definir os requisitos funcionais e não funcionais do futuro sistema.
- o diagrama do projeto arquitetural representa uma visão global do sistema expandindo o diagrama de requisitos finais e exibindo mais detalhes do sistema. Ou seja, define a arquitetura global do sistema em termos de subsistemas, interconectados por meio de fluxos de dados e controle. Além disso, também é possível observar um mapeamento dos atores do sistema para um conjunto de agentes de *software*, cada um caracterizado por recursos específicos.
- no diagrama de projeto detalhado, a ideia é especificar os recursos e interações dos agentes. Para um melhor nível de detalhamento, o ideal é que a plataforma de implementação já tenha sido escolhida. Isso deve ser levado em consideração para executar um *design* detalhado que será mapeado diretamente para o código.
- o diagrama de implementação segue passo a passo a especificação detalhada do projeto com base no mapeamento estabelecido entre as construções da plataforma de implementação e as noções detalhadas do projeto.

Cada uma das fases da metodologia Tropos fornece um diagrama específico para representar graficamente a etapa. Os requisitos iniciais e finais utilizam diagramas que representam a análise de requisitos do sistema, enquanto os diagramas de projeto arquitetural e detalhado representam as etapas de especificação do sistema.

Para esses três primeiros diagramas o *framework i** atende perfeitamente, pois trata da modelagem de requisitos e objetivos da aplicação. Já para o projeto detalhado e implementação os diagramas UML podem ser aplicados, pois estão mais relacionados a especificidades do desenvolvimento da aplicação.

Para a modelagem do SMA desenvolvido nesse trabalho foi utilizado a ferramenta online *piStar* (*online iStar modelling*²). O *piStar* é uma ferramenta de código aberto executada no navegador, dispensando instalação no *desktop*, com boa fidelidade visual, permitindo que os diagramas sejam bem apresentados mesmo quando impressos e suporta o padrão iStar 2.0 (Pimentel and Castro, 2018). Os diagramas que representam a solução proposta nesse trabalho serão apresentados na proposta (Capítulo 4).

2.2.2 Raciocínio de agentes

Uma vez elaborada a modelagem dos agentes é importante definir também como será o comportamento autônomo de cada um dos agentes com base nos objetivos que os mesmos devem alcançar.

Partindo do pressuposto que os agentes deverão recomendar tratamento aos pacientes, é interessante que além de identificar o estado de saúde atual do paciente, seja feito também uma previsão de como o paciente estará em um determinado momento no futuro. Com a predição será possível que os profissionais de saúde acompanhem o paciente mais frequentemente. Para desempenhar seus papéis, fazendo recomendações de tratamento de forma autônoma, os agentes podem raciocinar conforme um conhecimento especializado individual a semelhança de um especialista humano. Desta forma, identificou-se a necessidade de utilizar técnicas de predição e regras de produção para representação de conhecimento médico.

Modelos de regressão e predição

O uso da predição pode auxiliar a tomada de decisão dos agentes. No entanto, a escolha da técnica de predição deve ser feita de forma sistemática, uma vez que uma determinada técnica pode atender melhor as necessidades específicas que outras. Um modelo de regressão viabiliza prever informações ou dados futuros a partir de dados históricos.

Segundo Mendes (2017), técnicas de IA tais como máquina de vetor de suporte (*Support Vector Machine* (SVM)) e redes neurais (*Neural Networks* (NN)), entre outras relacionadas a aprendizado de máquina (*Machine Learning* (ML)), também podem ser utilizadas para predição. Porém, modelos de regressão linear, diferentemente das técnicas de predição de IA, geralmente conseguem resultados satisfatórios quando treinados com um conjunto de dados menor. As SVM e as NN necessitam de um conjunto de dados maiores para alcançar bons resultados de treinamento. Além disso, o uso de modelos de regressão já apresentaram resultados satisfatórios em trabalhos na área da saúde, mais especifica-

²PiStar está disponível em <https://www.cin.ufpe.br/~jhcp/pistar/tool/>.

mente para a predição de níveis de glicose em pacientes criticamente enfermos (Zhang, 2015), e mais recentemente de novos casos de COVID-19 (Rath et al., 2020).

Um modelo de regressão se baseia na correlação entre duas ou mais variáveis, onde existe dependência entre elas. Quando a relação é de um para um tem-se uma Regressão Linear Simples (RLS). Mas quando a relação é de um para muitos tem-se uma Regressão Linear Múltipla (RLM). Ou seja, a RLS é basicamente uma função de primeiro grau, cujo objetivo dado um conjunto de dados é entender o padrão entre eles que possa ser descrito por uma função de primeiro grau com uma única variável. Quando se trata de mais de uma variável caracteriza-se uma RLM (Barbettam et al., 2010).

Regressão Linear Simples A RLS busca apresentar a relação entre duas variáveis, como pode ser visto na Equação 2.1, onde o y é a variável dependente, ou seja, o valor que se deseja prever. O β_0 é a constante que representa a interceptação (intercepto) da reta no eixo y e β_1 é a constante que representa o coeficiente angular da reta. O x é a variável independente, ou variável preditora que tem o poder de influenciar na variável que se deseja encontrar, e ε a variável que representa os fatores de resíduos dos erros de medição (Rath et al., 2020).

$$y = \beta_0 + \beta_1 x + \varepsilon \quad (2.1)$$

Regressão Linear Múltipla A RLM envolve três ou mais variáveis, ou seja, uma variável dependente (y) e duas ou mais variáveis explanatórias ($\beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k$). A descrição das variáveis segue a mesma da RLS. Nesse caso, o objetivo também é o de estabelecer uma equação que possa prever valores de y para valores dados das k variáveis independentes. O modelo da RLM está expresso na Equação 2.2 (Rath et al., 2020).

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k + \varepsilon \quad (2.2)$$

Geralmente os valores de x e y são conhecidos, pois podem estar presentes em uma base de dados. Já os coeficientes precisam ser calculados e para isso se utiliza do Mínimos Quadrados Ordinários (MQO) ou (*Ordinary Least Squares* (OLS)). Tal método busca o melhor valor que os coeficientes possam atingir, de maneira que a diferença entre o valor predito pela função e o real sejam os menores (Barbettam et al., 2010). Munido dos valores necessários a regressão pode ser calculada, e seus resultados aferidos por meio da validação por meio de diversos fatores conforme apresentados na Seção 2.3.

Agente baseado em conhecimento

Segundo Russell and Norvig (2010), agentes baseados em conhecimento (*Knowledge Based* (KB) ou *Knowledge Based Systems* (KBS)) são compostos por uma base de conhecimento e um mecanismo de inferência lógica. Os agentes utilizam inferência lógica para derivar novas informações sobre o mundo a partir do conhecimento presente já existente na base de conhecimento. O agente incorpora suas percepções na base de conhecimento e consulta a base para escolher a melhor ação, registrando também a ação executada na base. Ou seja, inferir é o processo de produzir novas sentenças lógicas a partir de sentenças anteriores e utilizá-las para executar as ações dos agentes.

Segundo Contreras and Vehi (2018), sistemas baseados em conhecimento além da base de conhecimento e os mecanismos de inferência devem conter também um sistema de aquisição de conhecimento. O sistema de aquisição é usado para coletar informações os quais manipulados podem ser úteis para incremento do conhecimento inicial da base. Aspectos importantes aqui incluem relações, condições, recomendações, diretrizes e estratégias, sendo que o mecanismo de inferência vincula a base de conhecimento às informações coletadas.

Sistemas baseados em conhecimento viabilizam o raciocínio envolvendo o uso de técnicas lógicas como dedução e indução para gerar conclusões a partir do conhecimento especializado disponível na base de regras. O objetivo principal dos sistemas que implementam mecanismos de raciocínio é executar tarefas em nível de especialista humano de maneira restrita e especializada no domínio de interesse. Esses sistemas geralmente aplicam heurísticas para orientar o raciocínio e reduzir o espaço de pesquisa de possíveis soluções (Contreras and Vehi, 2018).

Segundo Russell and Norvig (2010), a inferência lógica pode ser alcançada por meio de dois tipos de algoritmos:

- no encadeamento para frente (*forward-chaining*) a inferência é iniciada a partir de fatos já conhecidos pela base de conhecimento, os quais são premissas conhecidas de uma implicação lógica, onde sua conclusão será acrescentada ao conjunto de fatos conhecidos. Esse é um exemplo de raciocínio orientado a dados, ou seja, o raciocínio cujo foco da atenção começa com os dados conhecidos e pode ser usado em um agente para derivar conclusões a partir de percepções de entrada; e
- o encadeamento reverso (*backward-chaining*), o qual tem a inferência iniciada a partir do objetivo, encadeando regras até encontrar fatos conhecidos que apoiem a prova. Essa é uma forma de raciocínio orientado por metas sendo o tipo de inferência utilizado como base para a programação lógica.

DeJournett and DeJournett (2016) associa os sistemas baseados em conhecimento a controladores de IA que procuram capturar o processo de pensamento humano, criando regras que imitam a razão usada pelos seres humanos. Onde um sistema típico baseado em conhecimento é criado quando um especialista em domínio se senta com um engenheiro de conhecimento e explica suas linhas de raciocínio para desempenhar certas funções. O engenheiro por sua vez transforma as linhas de raciocínio em uma série de regras *if-else* que imitam o pensamento dos especialistas no domínio. Esse tipo de controle é considerado encadeamento direto, pois os dados disponíveis, como por exemplo a faixa desejada de controle da glicose do paciente são fatos conhecidos ou premissas de uma implicação lógica.

Em suma, um sistema baseado em conhecimento humano pode simular a capacidade de tomada de decisão de um especialista. Um exemplo de regra de inferência lógica utilizando a escala de valores glicêmicos apresentados nas Tabelas 2.1 e 2.2 é apresentada na sequência. A regra estabelece para um paciente na UTI que apresenta hiperglicemia grave (hiperG = 180 a 250 mg/dL) a aplicação de insulina:

IF glicemia > 180 AND glicemia < 250 THEN “Insulina Regular: 4 UN - SC”

Este exemplo simples de regra de inferência ilustra a possibilidade da base de conhecimento do sistema abranger vários cenários com tratamento relacionados ao controle glicêmico de pacientes na UTI.

2.2.3 Tecnologias de SMA

A implementação de um SMA com agentes baseado em conhecimento se dá por meio de dois tipos de ferramentas distintas para:

- implementar o funcionamento, comunicação e interação dos agentes; e
- implementar o raciocínio dos agentes utilizando o formalismo de regras de inferência, permitindo uma abordagem declarativa para processar a decisão inteligente do agente.

Neste trabalho foram estudadas ferramentas para implementar um SMA, tais como o JADE (Bellifemine et al., 2007) e o PADE (Melo et al., 2019). Para implementar agentes baseado em conhecimento foi estudado o Drools (The Drools Team, 2020) e o Experta (Pérez, 2019). Essas ferramentas serão apresentadas na sequência. Serão apresentadas também algumas bibliotecas da linguagem Python que auxiliam nos cálculos estatísticos propostos.

JADE

O *Java Agent DEvelopment framework* (JADE) é um *middleware* de desenvolvimento de SMA escrito na linguagem de programação Java (Bellifemine et al., 2007). O JADE foi definido em conformidade com as especificações da *Foundation for Intelligent Physical Agents* (FIPA) e por meio de um conjunto de ferramentas que suportam a fase de depuração e implantação. O JADE pode ser distribuído entre máquinas de múltiplas plataformas e sua configuração pode ser alterada em tempo de execução, permitindo criar novos agentes, inclusive agentes móveis de uma máquina para outra, desde que essas máquinas tenham instalado o *Java Run Time 5* (JRE5).

Uma das grandes vantagens do JADE é a conformidade com a especificação FIPA, facilitando a integração com outros serviços de agentes. São serviços nativos do JADE: o gerenciamento do ciclo de vida, o transporte de mensagens e o serviço de páginas amarelas e brancas. Alguns serviços opcionais incluem a integração agente-software, o serviço de ontologia e a interação agente-homem. O serviço de transporte de mensagens *Message Transport Service* (MTS) é o responsável por entregar as mensagens trocadas entre os agentes de uma mesma plataforma ou entre plataformas. Todos os agentes FIPA têm acesso a no mínimo um MTS e somente mensagens endereçadas aos agentes podem ser enviadas devido a mecanismos de segurança.

Conforme especificação FIPA, o *framework* JADE inclui dois recursos especiais o *Agent Management System* (AMS) e o *Directory Facilitator* (DF), os quais são automaticamente ativados quando o JADE é inicializado. O AMS é um sistema gerenciador de agentes que exerce o controle de acesso e uso da plataforma, tratando da criação, finalização e demais etapas do ciclo de vida dos agentes. O DF oferece o serviço de páginas amarelas para os agentes, fazendo o papel de um banco de dados centralizado cujas entradas associam um agente aos seus serviços.

PADE

O *Python Agent DEvelopment framework* (PADE) é um *framework* para desenvolvimento, execução e gerenciamento de sistemas multiagentes em ambientes de computação distribuída escrito em linguagem Python (GREI-UFC, 2019). PADE é software livre, licenciado sob os termos da licença MIT, desenvolvido pelo *Grupo de Redes Elétricas Inteligentes* (GREI) do Departamento de Engenharia Elétrica da Universidade Federal do Ceará e utiliza as bibliotecas do projeto Twisted para implementar a comunicação entre os nós da rede.

O PADE foi desenvolvido considerando os requisitos para sistema de automação e oferece os seguintes recursos: abstração para implementação dos agentes utilizando Orientação a Objetos; ambiente de execução de agentes inteiramente em código Python; módulo para a implementação dos protocolos definidos pela FIPA; módulo para construção e tratamento de mensagens no padrão FIPA-ACL (*Agent Communication Language* (ACL)), além de permitir o envio de objetos serializados como conteúdo das mensagens ACL. PADE provê implementação de comportamentos cíclicos e temporais, a interação com banco de dados, entre outras funcionalidades.

Os criadores do PADE afirmam que o *framework* é de fácil instalação e configuração, multiplataforma, podendo ser instalado e utilizado em hardwares embarcados que executam sistema operacional Linux, como Raspberry Pi e BeagleBone Black, bem como sistema operacional Windows. Encontra-se disponível tanto o código, quanto exemplos de utilização do *framework* no Github³ que podem ser baixados, instalados e executados por meio de comandos presentes na documentação⁴.

Drools

Drools (*Business Rule Management System* (BRMS)) é um software de código aberto, lançado sob a licença Apache 2.0., escrito 100% em Java e execução em qualquer *Java Virtual Machine* (JVM) (The Drools Team, 2020). Drools disponibiliza um sistema de raciocínio que utiliza mecanismos de inferência com encadeamento *forward-chaining* e *backward-chaining*, permitindo a avaliação de regras de negócios e processamento de eventos complexos. As regras de produção são estruturadas em condições e ações (*IF-THEN*). O mecanismo Drools armazena, processa e avalia dados para executar as regras de negócio ou modelos de decisão definidas pelo usuário. A função básica do mecanismo Drools é corresponder dados ou fatos recebidos as condições das regras e determinar como executar as regras.

Experta

Experta é uma biblioteca Python que pode ser usada no desenvolvimento de sistemas especialistas. A biblioteca foi construída com objetivo de implementar uma alternativa Python ao CLIPS (ferramenta conhecida para implementação de KBS). Apesar de alcançarem objetivos de forma semelhante, Experta é uma biblioteca Python enquanto CLIPS é uma linguagem de programação escrita em linguagem C, fato esse que impõe algumas limitações nas construções. Como principais características Experta é compatível

³<https://github.com/grei-ufc/pade>

⁴<https://pade.readthedocs.io/en/latest/>

com Python 3, implementação pura em Python e correspondente baseado no algoritmo RETE (Pérez, 2019).

De acordo com Pérez (2019), um sistema desenvolvido com Experta é capaz de emparelhar um conjunto de fatos com um conjunto de regras para esses fatos e executar ações com base nas regras de correspondência. Os fatos são a unidade básica de informação sendo usados pelo sistema para raciocinar sobre o problema. As regras são uma chamada exigível (*callable*), decorado com a notação *@Rule*. As regras têm dois componentes, LHS (lado esquerdo) e RHS (lado direito). O LHS descreve (usando padrões) as condições nas quais a regra deve ser executada (ou disparada). O RHS é o conjunto de ações a serem executadas quando a regra é acionada.

Além dos fatos e regras Experta trabalha com o *KnowledgeEngine*, onde todo o conhecimento é gerado. A primeira etapa é fazer uma subclasse dele e usar a regra para armazenar seus métodos, ou seja, as ações que devem ser executadas quando a regra é acionada. Depois disso, instanciá-lo, preenchê-lo com fatos e executá-lo, permitindo que o sistema alcance o conhecimento acerca do problema. É possível encontrar exemplos no GitHub⁵ (Pérez, 2019).

Bibliotecas Python

Conforme apresentado na Seção 2.2.2, a utilização dos modelos de regressão e predição requer diversos cálculos matemáticos. Para customizar esse processo foram utilizadas bibliotecas (*libs*) na linguagem Python que disponibilizam além de cálculos matemáticos, estatísticos, transformação e manipulação de dados, e uso de técnicas de aprendizado de máquina. Além disso possui vasta comunidade e documentação. Pode-se citar por exemplo as seguintes bibliotecas:

- SciPy - é uma ferramenta de código aberto para trabalhar com computação científica na linguagem Python, que acopla outras bibliotecas formando um ecossistema baseado em Python, criada com o intuito de facilitar o desenvolvimento de cálculos matemáticos e estatísticos. Entre as bibliotecas contidas no SciPy estão NumPy e Matplotlib. (<https://www.scipy.org>)
- NumPy - usada principalmente para realizar cálculos em *arrays* multidimensionais. Fornece um grande conjunto de funções e operações que ajudam os programadores a executar cálculos numéricos das mais diversas natureza. Ainda de acordo com (<https://numpy.org/>) essa lib é interoperável, performática, fácil de usar e de código aberto.
- Pandas - possui código aberto, voltada a análise e manipulação de dados que permite trabalhar de forma rápida e eficiente com arquivos do tipo csv (Comma Separated Values), sendo esse formato presente na maioria dos datasets disponíveis para experimentos (<https://pandas.pydata.org/>).
- Scikit-learn - biblioteca de aprendizado de máquina desenvolvida em linguagem Python, de código aberto. Ela inclui vários algoritmos de classificação, regressão, agrupamento, entre outros e foi projetada para interagir com bibliotecas Python numéricas e científicas como por exemplo a NumPy (<https://scikit-learn.org/stable/>).

⁵<https://github.com/nilp0inter/experta/tree/develop/docs>

A combinação da utilização dessas ferramentas permitem trabalhar facilmente e de maneira bem eficiente com os mais diversos cálculos e modelos estatísticos.

Embora as linguagens Java e Python apresentem recursos necessários para a implementação de agentes, inclusive fornecendo *frameworks* tais como os apresentados, JADE e PADE, o uso da linguagem Python com o *framework* PADE e a biblioteca Experta, foram os escolhidos para esse trabalho. Tal escolha se deu devido ao maior domínio / afinidade por parte do desenvolvedor com a linguagem Python por conta da sintaxe intuitiva e facilidade em programar. Além de apresentar inúmeros recursos e bibliotecas para cálculos estatísticos, manipulação de dados e objetos *JavaScript Object Notation* (JSON), ferramentas de IA entre outras funcionalidades (Buitinck et al., 2013).

2.3 Avaliação de resultado

Conforme Barbettam et al. (2010), existem fatores que precisam ser considerados na análise de modelos para estabelecimento de performance de resultados. Acerca dos modelos de regressão e predição tais fatores podem ser avaliados, como:

- teste F de significância global - verifica se ao menos uma variável do modelo está relacionada com a variável alvo. Como resultado de significância do modelo, o valor-p (probabilidade de se obter uma estatística de teste igual ou maior que a observada em uma amostra) precisa ser menor que 0.05.
- teste de significância individuais ou valor-p dos coeficientes - verifica quanto das variáveis preditoras explicam a variável alvo. A métrica padrão é o valor-p ser menor que 0.05.
- coeficiente R^2 - verifica quanto o modelo explica seus resultados. É um valor entre 0 e 1, sendo que quanto mais próximo de 1 melhor o modelo ajusta os dados.

O coeficiente R^2 indica a porcentagem de variação da variável de resposta explicada pela relação com uma ou mais variáveis preditoras. Esse coeficiente é amplamente utilizado para validar uma regressão, porém tal modelo de validação usado de forma isolada pode se tornar tendencioso, sendo necessário utilizar outros métodos estatísticos para avaliar o modelo. Neste sentido, outras medidas de avaliação podem ser utilizados para validar a acurácia dos modelos, como a aferição dos erros contidos em uma predição, tais como: a média do erro absoluto *Mean Absolute Error* (MAE), a raiz quadrada do erro médio *Root Mean Square Error* (RMSE) e o percentual de erro médio absoluto *Mean Absolute Percent Error* (MAPE) (Mendes, 2017).

Segundo Mendes (2017), é importante analisar os erros ou resíduos de uma regressão, estes por sua vez representam os valores de variabilidade em y não explicados por x . O menor valor dos resíduos significa uma melhor modelagem entre as variáveis dependentes e independentes. A avaliação dos resíduos tem como dependência a normalidade e a distribuídos independentemente dos erros, com média zero e desvio-padrão constante. Para validar o modelo de regressão é necessário realizar a análise dos resíduos e verificar se atendem os aspectos citados.

Validação cruzada *k-fold*

Quando se tem poucos dados ou um único dataset, por exemplo, uma maneira de aumentar a carga de testes nesse seria utilizando um método de *cross-validation*.

Um dos métodos da validação cruzada é o *k-fold*, este consiste em particionar um conjunto de dados, como um *dataset* por exemplo, em k partes e usar $k-1$ partes para treino e o restante para teste, executando k testes. Para cada um dos k testes o modelo em questão é testado utilizando uma métrica escolhida, podendo esta ser por exemplo os erros (MAE, RMSE ou MAPE) apresentados anteriormente. Por fim, de posse das k medidas pode-se chegar ao resultado final efetuando a média ou desvio-padrão dos valores. Tanto o *k-fold* quanto outros métodos de validação cruzada podem ser facilmente implementados com o auxílio da biblioteca scikit-learn (Buitinck et al., 2013).

Validação de teste diagnóstico

Para avaliar se o sistema baseado em conhecimento raciocinou e teve respostas similares aos profissionais de saúde uma comparação entre as respostas acompanhada pela contagem das combinações pode ser elaborada, extraíndo-se a porcentagem de casos coincidentes.

Tal análise pode ser complementada por meio da avaliação de teste diagnóstico, que consiste basicamente em comparar respostas ou diagnósticos com um padrão ouro (*golden standard*). O padrão ouro pode ser determinado como o verdadeiro estado do paciente, se a informação estiver disponível, um conjunto de exames julgados mais adequados, ou uma outra forma de diagnóstico que sirva de referência. Esse tipo de validação está muito presente na investigação clínica e epidemiologia. Para determinar a validade, compara-se os resultados do teste com o padrão ouro. O teste diagnóstico ideal deveria fornecer sempre a resposta correta, ou seja, um resultado positivo nos indivíduos com a doença e um resultado negativo nos indivíduos sem a doença, por exemplo. Além do que, trata-se de um teste rápido de ser executado, seguro, simples, inócuo, confiável e de baixo custo (Knottnerus and Buntinx, 2011).

Segundo Knottnerus and Buntinx (2011), dois fatores podem ser utilizados para mensurar a qualidade do teste: a sensibilidade e a especificidade. Para defini-los são utilizados como exemplos, testes com resultados dicotômicos, ou seja, resultados expressos em duas categorias: positivos ou negativos. A Figura 2.5 apresenta as relações entre os resultados de um teste e o diagnóstico verdadeiro, onde o teste é considerado positivo (anormal) ou negativo (normal), e a doença presente ou ausente.

Sendo assim, na avaliação de um teste diagnóstico existem quatro interpretações possíveis para o resultado do teste: duas em que o teste está correto e duas em que está incorreto. O teste está correto quando ele é positivo na presença da doença (resultados verdadeiros positivos - a), ou negativo na ausência da doença (resultados verdadeiros negativos - d). Entretanto, o teste está incorreto quando ele é positivo na ausência da doença (falso positivo - b), ou negativo quando a doença está presente (falso negativo - c). Os melhores testes diagnósticos são aqueles com poucos resultados falso-positivos e falso-negativos (Knottnerus and Buntinx, 2011).

Com base nos valores da amostra, sensibilidade e especificidade é possível se extrair a acurácia do teste, onde:

- sensibilidade é a capacidade que o teste diagnóstico apresenta de detectar os indivíduos verdadeiramente positivos, ou seja, de diagnosticar corretamente os doentes

podendo ser definida pela Equação 2.3.

$$sensibilidade = \frac{a}{a + c} \quad (2.3)$$

- especificidade é a capacidade que o teste diagnóstico tem de detectar os verdadeiros negativos, isto é, de diagnosticar corretamente os indivíduos sadios definida pela Equação 2.4.

$$especificidade = \frac{d}{b + d} \quad (2.4)$$

- acurácia é a classificação correta do teste como um todo, considerando a sensibilidade e especificidade, sendo definida pela Equação 2.5.

$$acuracia = \frac{n}{a + d} \quad (2.5)$$

Além da acurácia, o valor de *F1-score* pode ser obtido por meio da média harmônica entre a sensibilidade e especificidade. De modo a trazer um número único que indique a qualidade geral do modelo, onde quanto mais próximo de 1, melhor. Esse valor pode ser obtido por meio da Equação 2.6.

$$F1 - score = \frac{2 * especificidade * sensibilidade}{especificidade + sensibilidade} \quad (2.6)$$

		Doença (padrão-ouro)		
		Presente	Ausente	
Resultado	Positivo	a) verdadeiro positivo	b) falso-positivo	a + b
	Negativo	c) falso-negativo	d) verdadeiro negativo	c + d
		a + b	b + d	n(a+b+c+d)

Figura 2.5: Validação de um teste diagnóstico (adaptado de Knottnerus and Buntinx (2011))

Em suma, a sensibilidade está relacionada a probabilidade de um desfecho esperado dar positivo, enquanto a especificidade é a probabilidade de que um paciente sadio tenha como desfecho negativo.

Capítulo 3

Revisão da Literatura

A revisão da literatura foi iniciada em junho de 2020 através da pesquisa de artigos que aplicassem técnicas de IA nas áreas de controle glicêmico e UTI. A princípio foram detectados dois artigos de revisão no tema: Contreras and Vehi (2018) e Dankwa-Mullan et al. (2019). Esses trabalhos incluem a utilização de técnicas de IA no diagnóstico, monitoramento e tratamento de controle glicêmico, incluindo ferramentas de software desenvolvidas, mais especificamente para auxílio a pessoas com Diabetes. Tal patologia tem forte relação com controle glicêmico já que o diabetes é mensurado por meio da quantidade de glicose no sangue. Ambos os trabalhos ressaltam a importância da utilização de técnicas de IA nesse contexto utilizando o grande volume de informação disponível, o que dificulta a manipulação por superar os limites de capacidade humana. Outra revisão feita por Malak et al. (2019) foi encontrada para a utilização de técnicas de IA em UTI-N (UTI Neonatal). Além de abordar diversas técnicas de IA os autores destacam positivamente a utilização de SMA em ambientes de UTI devido a complexidade desses ambientes.

A revisão de Contreras and Vehi (2018) estuda os esforços recentes para usar técnicas de IA para auxiliar no gerenciamento e suporte a decisão no tratamento do diabetes. Em seu estudo foram pesquisados trabalhos no repositório digital da PubMed, na língua inglesa, incluindo o período de 2010 a 2018, totalizando 1849 artigos, dos quais foram selecionados 141 para uma revisão detalhada. Baseado no resultado do estudo dos artigos selecionados foi definida uma taxonomia funcional para o gerenciamento do diabetes com aplicação de técnicas de IA conforme apresentado na Figura 3.1. A taxonomia foi categorizada considerando o objetivo da técnica de IA aplicada, a saber: explorar e descobrir informações (*Discovering*), aprender a usar as informações (*Learning*) e extrair conclusões das informações (*Reasoning*). Os autores citam que há uma aceleração da atividade de pesquisa e desenvolvimento de aplicações que utilizam técnicas de IA para predição e prevenção de complicações associadas ao diabetes, as quais estão sendo usadas na prática clínica diária e no auto gerenciamento do diabetes pelos próprios pacientes.

O trabalho de Dankwa-Mullan et al. (2019) tem como objetivo analisar artigos relacionados ao diagnóstico e tratamento do diabetes por meio de ferramentas que utilizam técnicas de IA. O estudo foi realizado no repositório digital da PubMed entre março e maio de 2018 e abrange artigos de 2009 a 2018. Foram encontrados 763 artigos dos quais 450 foram selecionados para uma revisão detalhada. A análise dos artigos, identificou aplicação das técnicas de IA em quatro áreas principais: triagem automatizada da retina, suporte à decisão clínica, estratificação preditiva de risco populacional e ferramentas de

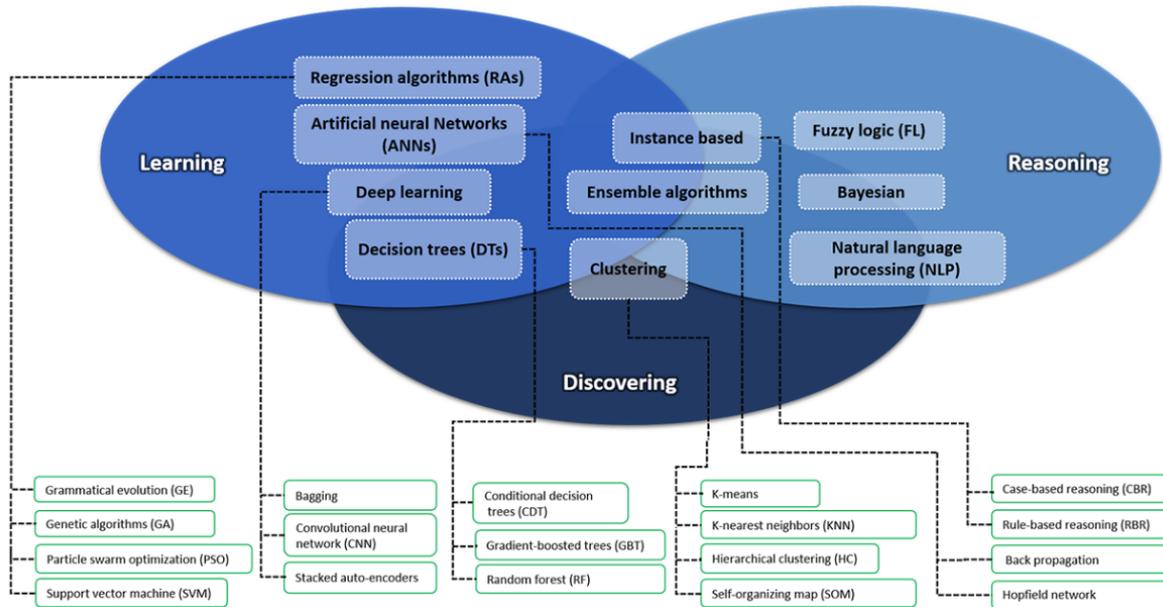


Figura 3.1: Taxonomia de métodos de IA (Contreras and Vehi, 2018).

autogestão do paciente. A partir dessas aplicações foi possível notar quais os métodos de IA mais utilizados: *Multilayer Perceptron*, *Convolutional Neural Network* (CNN), *Random Forest* (RF), *Fuzzy Logic* (FL), *Support Vector Machine* (SVM), Logistic Regression, *Natural Language Processing* (NLP) e *K-Nearest Neighbors* (KNN). Por fim um resumo contendo 24 artigos dos principais estudos selecionados de IA em diabetes é apresentado por meio de uma tabela comparativa. Considerando as ferramentas estudadas, os autores afirmam que o uso de técnicas de IA apresentam potencial de melhorar o tratamento do diabetes. Nesse potencial os autores citam: o melhor controle da glicemia e a redução de complicações na saúde dos pacientes, através de uma maior precisão, eficiência e facilidade de uso de ferramentas com técnicas de IA.

Na revisão de Malak et al. (2019) os autores estudaram as diferentes tecnologias de IA nos sistemas de suporte à decisão neonatal. Os SMA se destacaram em relação ao seu papel no diagnóstico, prognóstico, monitoramento e gestão da assistência médica. As pesquisas abrangem além da PubMed, as bases Science Direct e IEEE e incluíram artigos publicados no período de 1990 a 2018. Foram encontrados 90 artigos e após a análise 67 se mostraram relevantes. Os autores relatam a evidência de que devido ao ambiente dinâmico e heterogêneo da UTI-N torna-se inviável que uma única ferramenta ou tecnologia possa cobrir todas as necessidades de UTI. Desta forma, faz-se necessário combinar diversas técnicas de IA, sensores, trabalhar com vários agentes para fornecer soluções integradas e em tempo real.

Foi possível notar que as revisões de Contreras and Vehi (2018) e Dankwa-Mullan et al. (2019) destacam que técnicas de IA são amplamente utilizadas para diagnóstico e acompanhamento de controle glicêmico, enquanto a revisão de Malak et al. (2019) cita que SMA são os mais indicados para resolver problemas relacionados a UTI.

Após o estudo das revisões citadas foi realizada uma revisão da literatura utilizando as seguintes palavras de busca: *multiagent systems*, *glucose control*, *intensive care unit*. A pesquisa de artigos na língua inglesa foi realizada no portal de periódicos da Coorde-

nação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) ¹ incluindo publicações de 2015 a 2019 (últimos 5 anos), onde 26 trabalhos foram encontrados. Após a análise dos títulos e resumos quatro artigos foram selecionados e após a leitura completa somente dois foram considerados relevantes para este trabalho. Foram excluídos artigos com cunho teórico de controle glicêmico e que não citaram uso de tecnologias. A partir desses artigos buscou-se artigos dos mesmos autores e mais três foram adicionados, além de outros dois artigos que apareceram entre as sugestões feitas pelo gerenciador de referências do Mendeley, totalizando sete trabalhos correlatos, os quais serão detalhados na discussão de trabalhos correlatos.

Revisão 2021

Em um segundo momento (fevereiro de 2021), a busca por artigos foi refeita com o intuito de atualizar as revisões anteriores. Desta vez, a busca limitou-se a artigos publicados entre 2020 e 2021 na língua inglesa. As mesmas palavras de busca foram utilizadas - *multiagent systems, glucose control, intensive care unit* - no portal de periódicos CAPES. Tal busca resultou em 19 artigos, onde quatro se mostraram pertinentes (diferentes dos encontrados anteriormente), porém tratavam apenas de glicemia e UTI, não traziam nada em relação a SMA ou sobre o uso de técnicas de IA.

Desta forma, decidiu-se refazer a busca alterando a palavra de busca *multiagent systems* por *artificial intelligence* para ficar mais genérico. Dessa vez 117 artigos foram retornados. Foi feita uma primeira seleção com base nos títulos e resumos, sendo que muitos dos resultados eram de técnicas de IA, mas não tinham relação com controle glicêmico ou UTI. Um total de oito artigos foram selecionados e após a leitura integral dos mesmos três foram selecionados e adicionados a essa revisão. Ao final de ambas as revisões foram selecionados dez trabalhos (sete da primeira revisão e três dessa revisão), ambos serão detalhados na sequência.

Discussão de trabalhos correlatos

O trabalho de Darabi et al. (2015) apresenta o protótipo de um SMA para auto-gerenciamento de pacientes diabéticos Tipo 2. Em termos gerais a aplicação recebe dados do usuário e de sua alimentação e fornece dados referentes a sua glicemia, além de oferecer dicas de dieta adequada para que o próprio usuário possa tomar as devidas providências. É uma aplicação projetada para uso pessoal e não para aplicação no contexto de UTI. Os dados relacionados a situação glicêmica do paciente são calculados por meio do algoritmo *Closest Point of Approach* (CPA) e utilizam os modelos de *Ackerman* e *random walk* para controle e monitoramento das glicemias. Porém, tanto o algoritmo quanto os modelos não foram detalhados no artigo. Os autores afirmam que o sistema tem a capacidade de desenvolver uma solução que pode ser facilmente usado por pacientes com diabetes, mas não foi apresentado uma validação de resultados. Os autores complementam que avaliar a funcionalidade da solução para o uso de pacientes diabéticos pode ser o próximo passo.

DeJournett and DeJournett (2016) realizaram uma análise *in silico* de um controlador de glicose baseado em IA projetado para uso no ambiente da UTI. Tal controlador utiliza raciocínio baseado em regras (*Rule-Based Reasoning* (RBR)) com definição das regras feita

¹<https://www-periodicos-capes-gov-br.ez1.periodicos.capes.gov.br/index.php?>

por um dos autores que é especialista na área de controle glicêmico. Foram simulados 80 pacientes e realizadas 126.000 simulações únicas de cinco dias. Devido o curto tempo de simulações dispensou possibilidade de se usar sistemas de aprendizado. Os autores afirmam que apesar do viés do simulador e ter sido levado em conta apenas a glicemia dos pacientes os resultados se mostraram satisfatórios. Com testes validados estatisticamente mostrou-se que os níveis glicêmicos permaneceram na grande maioria do tempo em faixas aceitáveis (97.8% na faixa alvo, 2.1% em hiperglicemia e 0,09% em hipoglicemia). Os autores concluem dizendo que o estudo viabiliza a criação de sistemas de pâncreas artificial para uso na configuração da UTI.

No trabalho de DeJournett and DeJournett (2017) foi proposto uma métrica composta que combina cinco métricas comumente usadas (hipoglicemia grave, hipoglicemia leve, glicemia normal - faixa alvo, hiperglicemia e Coeficiente de Variação (CV)) para avaliar o que constitui um controle eficaz da glicose e as compara com seis controladores de glicemia (*Ideal Medical Technologies* (IMT), *Protocolo Yale*, *Glucommander*, *Controlador PID de Wintergerst e cols*, *GRIP* e *NICE-SUGAR*). O chamado *Glicose Scoring Metric* (GSM) é uma tentativa de desenvolver um escore global de glicose e se propõe a fornecer um número indicativo de desempenho geral para comparação dos controladores de glicose. Cada controlador avaliou 80 pacientes simulados em cinco dias gerando 2400 simulações. Como validação os seis controladores foram comparados entre si em relação as cinco métricas além do GSM. Como resultado, o IMT (único controlador baseado em IA) se mostrou melhor do que todas as cinco métricas comparadas sendo 76% melhor do que o próximo melhor controlador. A validação do GSM foi feita na consistência de seus valores em relação aos outros índices de avaliação. Os autores definem tal proposta como sendo promissora para avaliar novos e existentes controladores de glicose baseados em UTI devido a sua modularização permitindo a inserção de novas métricas.

Em Jemal et al. (2017) foi proposto a modelagem, implementação e validação de um sistema de apoio à decisão especializado para a detecção do grau de risco de pacientes em UTI utilizando SMA combinado com uma base de conhecimento e *IntuicionistFuzzy System* (IFS). A validação de eficácia do protótipo se fez por meio da comparação entre o risco calculado pelo sistema proposto, intitulado de Sistema Médico Intuicionista de Apoio à Decisão Fuzzy (MIFEDSS) e o risco apresentado pela pontuação do *Modified Early Warning Score* (MEWS) em um estudo de caso real em 16 pacientes. O MEWS é um escore fisiológico para estimativa de pacientes, ou seja, é um guia para os profissionais de unidade de emergência para encontrar o nível de doença de um paciente com base em seus dados fisiológicos. O MIFEDSS foi implementado e testado dentro de um contexto da vida real da ESSALEMA Policlínica, Sfax, Tunísia e tanto os especialistas que colaboraram e utilizaram o sistema quanto um questionário proposto aos participantes o julgou como bom. Tal protótipo se mostrou mais eficiente do que o MEWS, com graus mais altos de sensibilidade e especificidade sendo registrados.

Em Malak et al. (2018) é apresentada uma arquitetura baseada em agentes com suporte à decisão e em tempo real para gerenciamento de recém-nascidos de alto risco internados em UTI-N. Foi realizada uma breve revisão de artigos sobre os sistemas de suporte à decisão de UTI-N e optou-se por utilizar SMA principalmente pelo motivo destes atuarem de maneira autônoma, proativa e colaborativa, além de suas capacidades de negociação, caracterizando-os como ideais para gerenciar ambientes em tempo real, complexos e críticos, como UTI. Para validação dos resultados foi feito um estudo comparativo com outros

sistemas existentes de suporte à decisão da UTI-N com base em suas funcionalidades. Os autores concluem dizendo que a arquitetura proposta ainda está sendo trabalhada na previsão dos resultados acerca da utilização de IA para previsão de risco de mortalidade neonatal e sua implementação completa é prevista para próximos trabalhos.

O trabalho de Jemal et al. (2019) investiga e discute sobre o escopo de eficiência da *Intuitionist Fuzzy Logic* (IFL) integrado à arquitetura SMA, com o objetivo de detectar o estado de saúde do paciente em UTI. Basicamente aborda o mesmo contexto e validação apresentadas no trabalho anterior Jemal et al. (2017), porém neste o sistema é chamado de *Intuitionistic Fuzzy Logic Decision Support System* (IFLDSS), trazendo mais detalhes em relação ao funcionamento do SMA. As validações trazem uma breve revisão da literatura, além de ter sido ampliado e estar funcionando em várias instituições médicas do país.

DeJournett et al. (2020) utiliza um sistema autônomo de controle de glicose (pâncreas artificial) com o intuito de diminuir problemas decorrentes da glicemia de pacientes em UTI. Tal sistema é composto de três partes: Monitor de Glicose (EIRUS); Software para controle da glicemia baseado em IA (DeJournett and DeJournett, 2016); e Bombas de Infusão automáticas. Tem como objetivo avaliar a segurança e desempenho deste sistema aplicando-o testes simulados em suínos em ambiente clínico. A amostra foi composta de 14 suínos que foram submetidos a desafios hipo e hiperglicêmicos por cinco horas. O sensor capturava os dados de glicose a cada 30s. Os animais foram separados em dois grupos (controle e tratamento), onde, no grupo controle, o médico acompanhou e tratou seis animais e o sistema autônomo (grupo tratamento) outros oito animais. Após análise estatística dos dados ficou comprovado que o grupo tratamento manteve os níveis de glicemia por mais tempo na faixa alvo (55,4% contra 32,8% do grupo de controle), além de ter abolido todos os eventos hipoglicêmicos graves superando até mesmo o desempenho do médico experiente na UTI.

No trabalho de Vehí et al. (2020) foi proposto a aplicação de quatro técnicas de aprendizado de máquina para resolver o problema de segurança no gerenciamento do Diabetes Tipo 1 (DT1), mais especificamente na previsão e prevenção de eventos hipoglicêmicos. Os métodos usados foram *Grammatical Evolution* (GE) para a previsão contínua de médio prazo dos níveis de glicose no sangue, SVM para prever eventos hipoglicêmicos durante os períodos pós-prandiais, Artificial Neural Networks (ANN) para prever episódios hipoglicêmicos durante a noite e *Data Mining* (DM) para criar perfis de cenários de gerenciamento de diabetes. De acordo com a taxonomia usada neste trabalho a técnica de IA foi classificada como *Ensemble Algorithms* (EA), pois abordou diversas técnicas diferentes. Embora a proposta consiste na combinação das capacidades de predição e classificação das abordagens implementadas, cada método foi analisado separadamente, pois cada sistema preditivo tem um desempenho melhor em cenários específicos. Foram feitos testes simulados com base em três *datasets* diferentes e a sensibilidade, especificidade e precisão de cada método foi avaliada por meio do *Matthews Coefficient Correlation* (MCC) e da *Clarke Error Grid* (CEG) e se mostraram satisfatórios em todos os casos. Os autores concluem dizendo que o uso simultâneo desses diferentes modelos de predição personalizados causará a redução significativamente no número de episódios de hipoglicemia, melhorando a segurança e proporcionando aos pacientes diabéticos maior confiança na tomada de decisões.

No artigo de Bertachi et al. (2020), os autores investigam a viabilidade de um modelo de predição baseado em ML para antecipar episódios de hipoglicemia noturna em pacien-

tes DT1. O experimento contou com dez adultos com DT1 que foram estudados durante 12 semanas. As informações sobre a glicemia dos pacientes foram capturadas por meio de entrada de dados do próprio paciente, monitoramento contínuo de glicose e um rastreador de atividade física (para identificar os momentos de sono). As predições foram geradas individualmente por meio dos algoritmos de aprendizado supervisionado *Multi-layer Perceptron classifier* (MLP) e SVM. Os resultados foram avaliados aplicando o Coeficiente de Variação e Gmean. Após a avaliação dos resultados constatou-se que mais de 70% das hipoglicemias noturnas podem ser evitados com a metodologia proposta. Os autores concluem afirmando a viabilidade do uso de técnicas de ML para abordar a previsão de hipoglicemia noturna.

O trabalho de Kim et al. (2020) propõe um modelo de predição de glicose para pacientes hospitalizados com Diabetes Tipo 2 (DT2) utilizando *Deep Learning* (DL), mais especificamente foi utilizado algoritmo de *Recurrent Neural Network* (RNN). Para o experimento foram coletados dados de 20 pacientes por uma semana em aparelho de Continuous Glucose Monitoring (CGM). O modelo explora os últimos 35 minutos e prevê a glicemia para os próximos 30 minutos. Apenas alguns desses pacientes se encontram internados na UTI. Foram comparados dois modelos de predição, o *Long-Short Term Memory* (LSTM) e o *Gated Recurrent Unit* (GRU), onde o segundo apresentou o melhor desempenho. Como resultado foi obtido uma média do erro quadrático médio (RMSE) de 21,5 e um erro percentual absoluto médio (MAPE) de 11,1%. Os autores concluem que devido o baixo erro de predições da proposta é possível prever o nível de glicose desses pacientes e complementam dizendo que melhorias podem ser feitas no modelo mensurando outros dados do paciente além da glicemia, ampliando também a base de dados para testes.

A escolha dos métodos de IA utilizados nesse trabalho teve grande influência dos trabalhos estudados durante a revisão. Buscou-se aproveitar as características de cada técnica de IA apresentada de acordo com a necessidade da solução em questão com o intuito de combinar os benefícios de cada tecnologia. Na sequência será apresentado um breve comparativo entre a solução apresentada nesse trabalho e as demais soluções apresentadas.

Trabalhos correlatos: Comparação

A Figura 3.2 apresenta um mapeamento dos trabalhos correlatos utilizados nesse trabalho associados a categorização taxonômica de Contreras and Vehi (2018). Cada técnica utilizada foi identificada utilizando uma coloração específica conforme as técnicas de IA utilizadas. O mapeamento considerou o objetivo da técnica de IA aplicada incluindo descoberta de informações (*Discovering*), aprendizado automático (*Learning*) e raciocínio (*Reasoning*).

No trabalho de Darabi et al. (2015) identificamos que SMA é uma solução atraente para trabalhar com soluções na área médica, inclusive no controle e monitoramento de glicemia, porém esse trabalho não é aplicado no contexto de UTI.

Já nos trabalhos de DeJournett (DeJournett and DeJournett (2016), DeJournett and DeJournett (2017) e DeJournett et al. (2020)) foi apresentado um controlador de glicose totalmente autônomo utilizado no contexto de IA, porém o monitor utilizado para a coleta da glicemia, leva apenas esta informação em consideração, enquanto o nosso trabalho utiliza varias variáveis que incluem, por exemplo, dados relacionados a saúde e alimentação

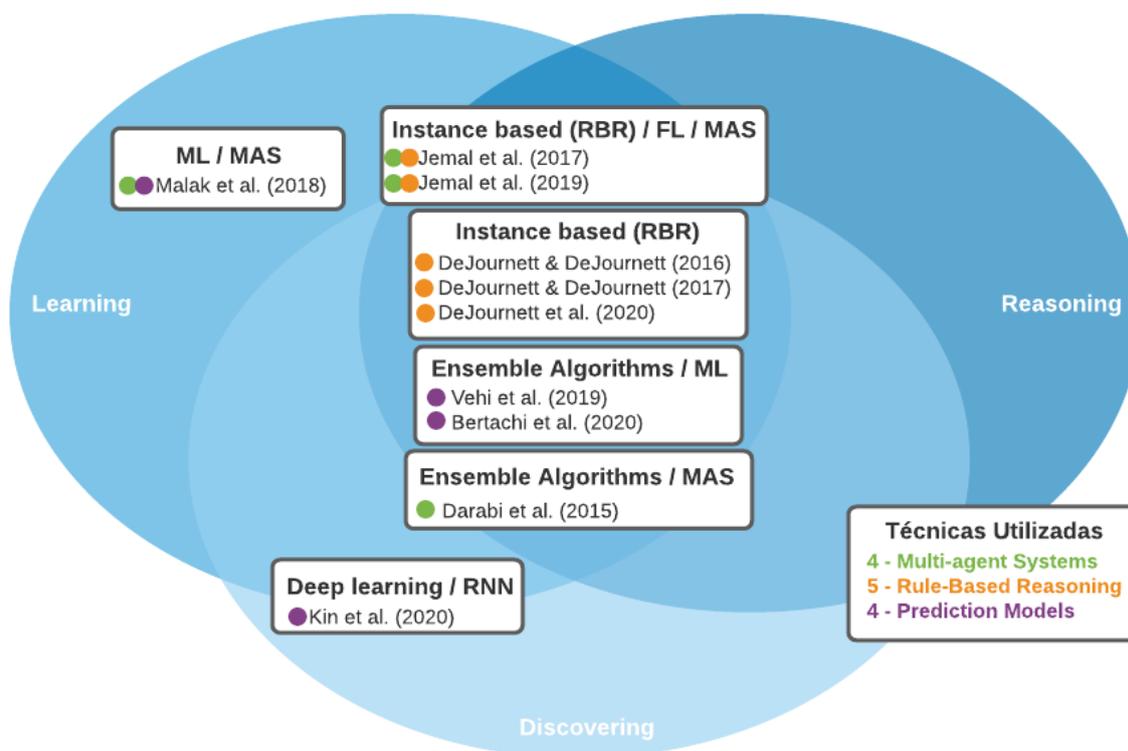


Figura 3.2: Trabalhos correlatos conforme a taxonomia de Contreras and Vehi (2018).

no paciente. Como técnica de IA, foi usado RBR que se mostrou uma ótima escolha, já que tal técnica utiliza como base o conhecimento de um médico especialista, o mais indicado para tratar desses pacientes.

Nos trabalhos de Jemal et al. (2017) e Jemal et al. (2019)) foi utilizado SMA e RBR, comprovando que combinar tais tecnologias é viável no contexto de UTI, pois a solução apresentada nesses trabalhos é um sistema de detecção de grau de risco de pacientes de UTI. O que difere do nosso trabalho é que os autores não focam apenas em eventos glicêmicos do paciente, mas faz uma análise geral do paciente utilizando FL. Em Malak et al. (2018) também foi utilizado um SMA no contexto de UTI, porém nada específico no controle glicêmico. Assim como Jemal et al. (2019), os autores fazem o gerenciamento de pacientes de UTI com foco em recém nascidos.

Os trabalhos de Vehí et al. (2020), Bertachi et al. (2020), Kim et al. (2020) apontam modelos de predição como uma solução interessante, especificamente no que diz respeito ao controle glicêmico. Onde a prevenção de eventos hipo ou hiperglicêmicos tende a ser mais eficiente do que a correção desses episódios.

- Vehí et al. (2020) apresentam uma solução para previsão e prevenção de eventos hipoglicêmicos em diabéticos e usa um combinado de técnicas/algoritmos de aprendizado de máquina.
- No trabalho de Bertachi et al. (2020) um modelo de predição para episódios de hipoglicemia noturna em diabéticos foi apresentado.

- Já Kim et al. (2020) utilizam DL, mais especificamente um algoritmo de RNN em um modelo de predição de glicose para pacientes DT2 hospitalizados.

Embora todos esses trabalhos estejam aplicados no controle glicêmico, nenhum deles tem foco em pacientes de UTI. Em Vehí et al. (2020) e Bertachi et al. (2020) o propósito de uso tem foco em pacientes diabéticos que podem nem estar internados, enquanto Kim et al. (2020) trata pacientes hospitalizados, porém não necessariamente em UTI.

O *Multiagent System for Glicemic Control* (MAS4GC) desenvolvido nesse trabalho apresenta uma solução baseada em SMA que utiliza modelos de predição e raciocínio baseado em RBR para fazer o controle glicêmico de pacientes internados em UTI. O conhecimento especializado fornecido por um médico intensivista foi formalizado através de regras de inferência. Esse conhecimento é usado pelos agentes para fazer predições de glicemia e recomendações de tratamento com o objetivo de manter a glicemia na faixa alvo por meio de recomendações de aplicações de glicose ou insulina além de sugerir o monitoramento de coletas. Vale ressaltar que as predições utilizadas nesse trabalho levam vários fatores em consideração, como dados alimentares e de saúde do paciente e não apenas sua glicemia.

A Tabela 3.1 apresenta as informações dos trabalhos correlatos incluindo o MAS4GC. Os aspectos analisados levam em consideração os métodos de IA apresentados na taxonomia de Contreras and Vehi (2018), o domínio de aplicação da solução, o método de validação, vantagens e desvantagens da proposta.

No Capítulo 4 será apresentado o modelo de solução desenvolvido incluindo o MAS4GC, com detalhamento da modelagem de agentes utilizando a metodologia Tropos (2.2.1), a arquitetura do sistema e o processo de raciocínio dos agentes com base no conhecimento médico do especialista em UTI. O MAS4GC foi validado através de dois experimentos que serão apresentados no Capítulo 5.

Tabela 3.1: Comparação dos trabalhos correlatos.

Referência	Método de IA	Domínio de Aplicação	Validação	Vantagens	Desvantagens
Darabi et al. (2015)	EA e SMA	sistema de controle e monitoramento de glicemia e dieta para pacientes diabéticos Tipo 2	não houve	utiliza SMA; detalha as funcionalidades dos agentes e cálculos utilizados em relação dieta/glicemia	não validou o sistema; não gerou dados para validação dos resultados; não apresenta o algoritmo CPA nem os modelos de Ackerman e Random Walk; não é aplicado no contexto de UTI
DeJournett and DeJournett (2016)	RBR	controlador de glicose (pâncreas artificial) para pacientes de UTI	pacientes e testes simulados, análise estatística	126.000 testes realizados em 80 pacientes simulados; conhecimento médico sobre glicemia foi modelado e implementado pelo especialista; controlador manteve a glicemia em 97.8% na faixa alvo	pacientes e testes simulados em apenas 5 dias; não possui aprendizado devido ao tempo limitado; viés nas simulações; leva somente a glicemia em consideração
DeJournett and DeJournett (2017)	RBR	cria métrica global para controle glicêmico em UTI	paciente e testes simulados, comparação dos controladores de glicemia	2400 simulações realizadas em 80 pacientes simulados; sistema modular que permite adição de novas métricas; controlador baseado em IA (<i>Ideal Medical Technologies</i>) 76% melhor que o próximo melhor; GSM se demonstrou consistente com as demais métricas	pacientes e testes simulados em apenas 5 dias; não fez a prova do GSM somente do controlador proposto
Jemal et al. (2017)	RBR, FL e SMA	detecção do grau de risco do pacientes em UTI	estudo de caso real, comparação com MEWS	utiliza SMA com base de conhecimento combinado com IFS; teste em ambiente real; graus mais altos de sensibilidade e especificidade que o MEWS	utilizou apenas 16 pacientes nos testes; depende da interação da equipe médica para entrada de dados e para acatar a decisão do sistema; não detalhou a arquitetura do SMA
Malak et al. (2018)	ML e SMA	gerenciamento de recém-nascidos em UTI-N	compara os recursos da sua proposta com os de outros sistemas disponíveis na literatura	utiliza SMA; apresenta uma arquitetura detalhada com ambiente colaborativo dos agentes que atuam em tempo real e proposta extensível	análise dos requisitos foi feita através da revisão de artigos; a implementação completa da arquitetura proposta e sua avaliação não foram concluídas
Jemal et al. (2019)	RBR, FL e SMA	diagnóstico do estado do paciente de UTI	estudo de caso real, comparação com MEWS	utiliza SMA com base de conhecimento combinado com IFL; teste em ambiente real; graus mais altos de sensibilidade e especificidade que o MEWS; implantado e utilizado em várias instituições médicas da Tunísia	utilizou apenas 16 pacientes nos testes; depende da interação da equipe médica para entrada de dados e para acatar a decisão do sistema; não detalhou a arquitetura do SMA
DeJournett et al. (2020)	RRBR	controlador de glicose (pâncreas artificial) para pacientes de UTI	testes clínicos simulados em suínos, análise estatística	teste em ambiente clínico; sistema totalmente autônomo; grupo tratamento (controlador IA) manteve os níveis de glicemia 55,4% do tempo na faixa alvo contra 32,8% do grupo de controle (médico); aboliu todos os eventos hipoglicêmicos graves; acesso a dados frequentes de glicose (ex. a cada 30s)	testes simulados por apenas 5 horas; sobredosagens de insulina e glicose sem correlação clínica; esta sujeito a falha mecânica tanto do sensor quando bombas de infusão
Vehí et al. (2020)	EA e ML	predição e prevenção de eventos hipoglicêmicos em diabéticos	testes simulados em 3 datasets e avaliação com MCC e CEG	testes com três <i>datasets</i> com dados específicos; resultados satisfatórios (como sensibilidade de 99% em um dos casos), combina diferentes técnicas; permite o monitoramento contínuo, pós-prandial e noturno	não unificou a implementação dos métodos; avaliação de cada método separadamente; avaliados em conjuntos de dados diferentes e em condições diferentes; específico para pacientes diabéticos; não se aplica ao cenário de UTI
Bertachi et al. (2020)	EA e ML	modelo de predição para episódios de hipoglicemia noturna em diabéticos	teste em pacientes reais (CV e Gmean)	testado em pacientes reais em ambiente de vida real; controle contínuo por meio de sensores; 70% de eficiência (prevendo eventos hipoglicêmicos)	testado em apenas 10 pacientes; dependia em partes, da entrada de dados e uso do dispositivo pelos pacientes; específico para pacientes diabéticos; não se aplica ao cenário de UTI
Kim et al. (2020)	DL e RNN	modelo de predição de glicose para pacientes diabéticos (DT2) hospitalizados	teste em pacientes reais; RMSE e MAPE; comparação entre os algoritmos LSTM e GRU	testado em pacientes reais e hospitalizados (alguns em UTI), monitoramento contínuo; baixo percentual de erro (11,1%)	Poucos pacientes (20); tempo curto de testes (uma semana); previsão em tempo curto (baseado nos 35 min últimos, prevê os próximos 30 min); analisa apenas glicemia;
MAS4GC	SMA e RBR	controle glicêmico de pacientes em UTI	experimentos com dados sintéticos	utiliza SMA combinado com RBR e modelos de predição; base de conhecimento fornecida pelo especialista; faz previsão das próximas glicemias levando em consideração dados de saúde e alimentação (não apenas a glicemia)	frequência nas coletas; depende do profissional da área médica para coletas de glicemia e aplicações (insulina/glicose); baseado em dados sintéticos

Capítulo 4

Apresentação da Proposta

Este trabalho apresenta uma solução baseada em agentes inteligentes para controle glicêmico de pacientes internados em UTI. A solução foi implementada através de um SMA intitulado *Multiagent System for Glicemic Control* (MAS4GC). O SMA é composto por três agentes que interagem entre si e também com um sistema Web chamado Glycon, utilizado como uma interface para coleta das glicemias e dados dos pacientes. O objetivo dos agentes é coletar, analisar os dados da base de dados do Glycon e fornecer informações importantes aos profissionais de saúde que atuam na UTI referente as glicemias dos pacientes. O desenvolvimento do MAS4GC foi acompanhado pelo médico especialista Dr. Sérgio Fernandes, o qual conduziu e validou por meio de seu conhecimento e experiência a fidelidade da solução.

A Figura 4.1 representa a metodologia utilizada na realização do trabalho. A primeira etapa da metodologia inclui o entendimento do problema, conforme os fundamentos teóricos apresentados no Capítulo 2. Na segunda etapa foi elaborado o pré-projeto de agentes utilizando o modelo PEAS. Na terceira etapa foi feita a modelagem dos agentes utilizando a metodologia Tropos, sendo composta pelos diagramas de requisitos iniciais, finais, projeto arquitetural, projeto detalhado e diagramas de implementação (UML). A quarta etapa refere-se a implementação do SMA. A seguir cada uma das etapas é detalhada.

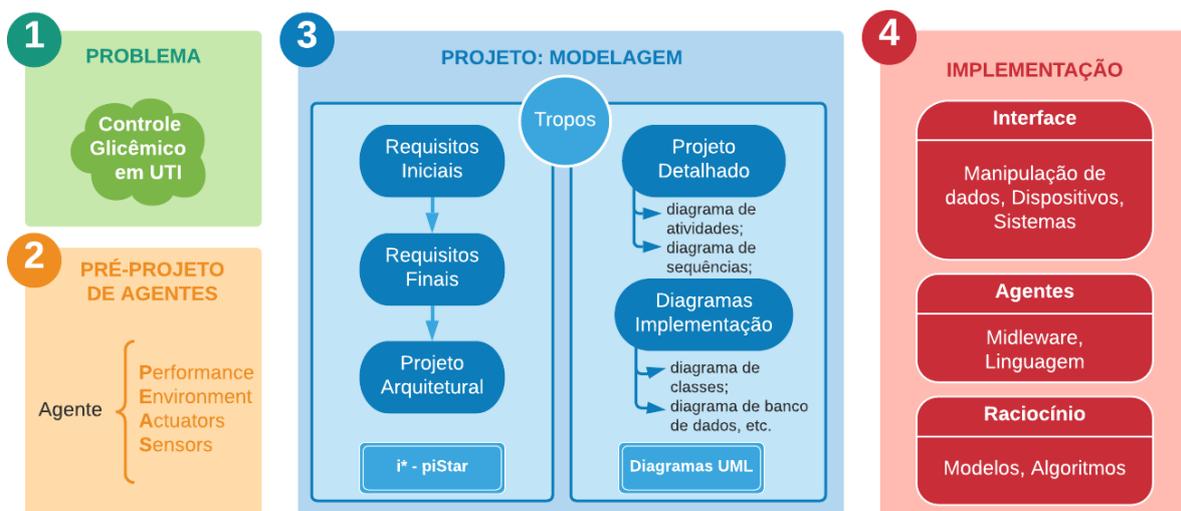


Figura 4.1: Metodologia de desenvolvimento do trabalho.

4.1 Pré-projeto de agentes

A proposta do modelo teve início através da definição do pré-projeto de agentes utilizado o modelo PEAS para identificação do ambiente onde os agentes atuarão, bem como suas respectivas características. Na sequência é apresentado uma descrição dos agentes do MAS4GC, seguido do modelo PEAS de cada agente nas Tabelas 4.1, 4.3 e 4.2:

- *Patient Analyzer Agent* (PAA) - o agente analisador do paciente tem como principal objetivo coletar os dados do paciente e sua respectiva glicemia periodicamente no Banco de Dados do Glycon. O agente deve analisar os dados e fazer um relatório avaliando a situação do paciente em comparação com os dados anteriores, caso haja. Essa análise permite que o agente calcule e faça previsões de como estarão as próximas glicemias do paciente. O relatório será enviado aos agentes PTA e AMA.
- *Propose Treatment Agent* (PTA) - o agente que propõe o tratamento é responsável por analisar o relatório de avaliação gerado pelo PAA e verificar se o paciente necessita de tratamento. Esse tratamento contempla a aplicação de glicose ou insulina, bem como a dose e o tipo de aplicação. O agente pode emitir alertas ao sistema Glycon para auxiliar os profissionais de saúde. Os profissionais de saúde podem acatar ou não a sugestão enviada pelo sistema e efetuar a aplicação.
- *Adjust Monitoring Agent* (AMA) - o agente que realiza o monitoramento também recebe o relatório do paciente enviado pelo agente PAA. Com base na situação glicêmica do paciente, o agente verifica se está em um intervalo aceitável. Caso não esteja, este agente indicará o horário na qual a próxima coleta deve acontecer, com frequências de coletas mais espaçadas ou mais próximas. Assim como o PTA, irá emitir alertas ao sistema Glycon para auxiliar os profissionais de saúde.

Vale ressaltar que o profissional terá responsabilidade de aceitar as sugestões do novo plano ou manter o já existente no sistema.

Tabela 4.1: PEAS do PAA.

P (performance)	Fazer as previsões das próximas glicemias (4h) dos pacientes com acurácia de pelo menos 80%.
E (environment)	UTI hospitalar com características: parcialmente observável; estocástico; sequencial; estático; discreto; multiagente.
A (actuators)	Determinar a situação do paciente: coletar e analisar os dados (paciente e glicemia); comparar com dados anteriores e prever próximas glicemias; compartilhar o relatório de avaliação com os demais agentes (PTA e AMA).
S (sensors)	Base de dados do Glycon e/ou um módulo automatizado contendo recursos de Internet das Coisas.

Tabela 4.2: PEAS do PTA.

P (performance)	Fazer recomendações de tratamento que se assemelhem em 80% com a de um profissional da saúde humano.
E (environment)	UTI hospitalar com características: parcialmente observável; estocástico; sequencial; estático; discreto; multiagente.
A (actuators)	Analisar relatório de avaliação; calcular novo tratamento; indicar o tratamento glicêmico.
S (sensors)	Relatório de avaliação do paciente fornecido pelo PAA.

Tabela 4.3: PEAS do AMA.

P (performance)	Fazer recomendações de monitoramento que se assemelhem em 80% com a de um profissional da saúde humano.
E (environment)	UTI hospitalar com características: parcialmente observável; estocástico; sequencial; estático; discreto; multiagente.
A (actuators)	Analisar relatório de avaliação; calcular próxima coleta; indicar o horário da coleta.
S (sensors)	Relatório de avaliação do paciente fornecido pelo PAA.

O ambiente foi definido como parcialmente observável pois, no contexto da UTI onde os pacientes se encontram, os agentes só conseguem observar o que for informado no Glycon. Estocástico, pois o próximo estado do ambiente não pode ser completamente determinado pelo estado atual ou pela ação executada pelo agente. sequencial já que a decisão ou ação de um agente pode afetar uma decisão futura. Apesar do ambiente de UTI ser totalmente dinâmico, os dados serão coletados assincronamente da base de dados do sistema e dependerá da entrada do mesmo pelos profissionais de saúde, o que lhe concede a característica de ambiente estático. O ambiente inclui um número finito de valores específicos, como por exemplo número de pacientes internados na UTI, desta forma é discreto. O ambiente é multiagente porque inclui mais de um agente.

4.2 Modelagem Tropos

O SMA foi modelado usando a metodologia Tropos, por meio da elaboração dos diagramas de requisitos iniciais, requisitos finais e de projeto arquitetural, seguindo a referida ordem e contemplando a evolução do processo de modelagem da solução. Para a modelagem Tropos foi utilizado a ferramenta piStar. E para representar o projeto detalhado e a implementação foram utilizados diagramas UML.

4.2.1 Requisitos iniciais

O diagrama de requisitos iniciais apresentado na Figura 4.2 identifica as partes interessadas no domínio e as define como atores: Paciente, Profissionais de Saúde, Glycon e Agentes. O modelo aborda de maneira inicial como acontece as dependências e trocas de recursos entre as entidades. O processo inicia com o Paciente que ao ser internado

na UTI recebe os cuidados dos profissionais de saúde, mais especificamente passa pela coleta da sua glicemia, em seguida o profissional informa tais dados, além dos dados do paciente ao sistema Glycon que por sua vez os armazena na base de dados. Uma vez armazenados os dados do paciente e sua glicemia os agentes podem fazer a coleta, análise e conseqüentemente, caso haja necessidade, informar uma nova sugestão de tratamento e monitoramento. Essa sugestão é enviada ao sistema Glycon ficando disponível visualmente aos profissionais de saúde que ao consultá-lo podem efetuar o tratamento adequado no paciente.

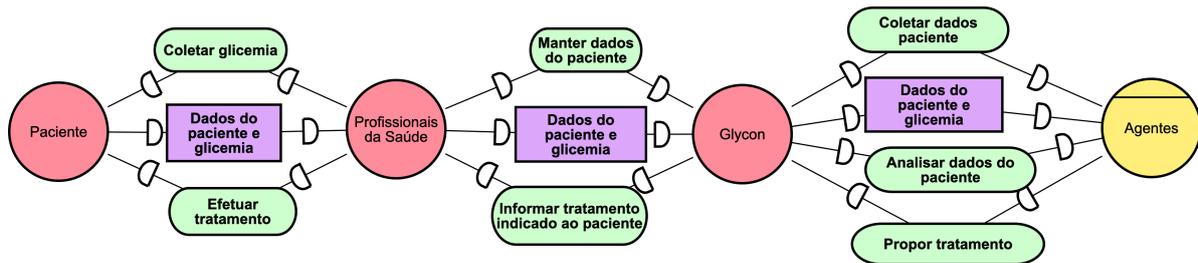


Figura 4.2: Diagrama Tropos referente a fase de requisitos iniciais.

4.2.2 Requisitos finais

Em relação ao diagrama de requisitos finais (Figura 4.3) é possível ver tal modelo expandido contendo as decomposições do ator profissionais de saúde em Médicos, Enfermeiros e outros. Em um ambiente de UTI esses atores podem acompanhar os pacientes inclusive fazer a coleta das glicemias e possíveis aplicações de glicose e insulina.

Devido ao detalhamento que este diagrama recebe em relação ao anterior agora é possível observar quais agentes comporão o modelo, bem como suas dependências em relação aos objetivos e até mesmo em relação aos recursos compartilhados. Esse diagrama permite ter uma ideia geral do sistema que será desenvolvido bem como definir quais serão seus requisitos funcionais e não funcionais.

4.2.3 Projeto arquitetural

Os requisitos finais podem ser explorados no diagrama de projeto arquitetural (Figura 4.4), onde uma visão global do sistema é apresentada. Nesse além de todos os atores envolvidos no ambiente é possível observar os agentes, suas tarefas (representando os requisitos funcionais) e desejos (requisitos não funcionais), bem como as dependências e relacionamentos para com os demais atores envolvidos no processo. Neste caso, o sistemas de agentes é representado pelo ator MAS4GC que representa os agentes que irão em conjunto coletar, analisar e fornecer informações como planos de coletas, propor tratamentos e até mesmo enviar alertas em relação a situação da saúde do paciente.

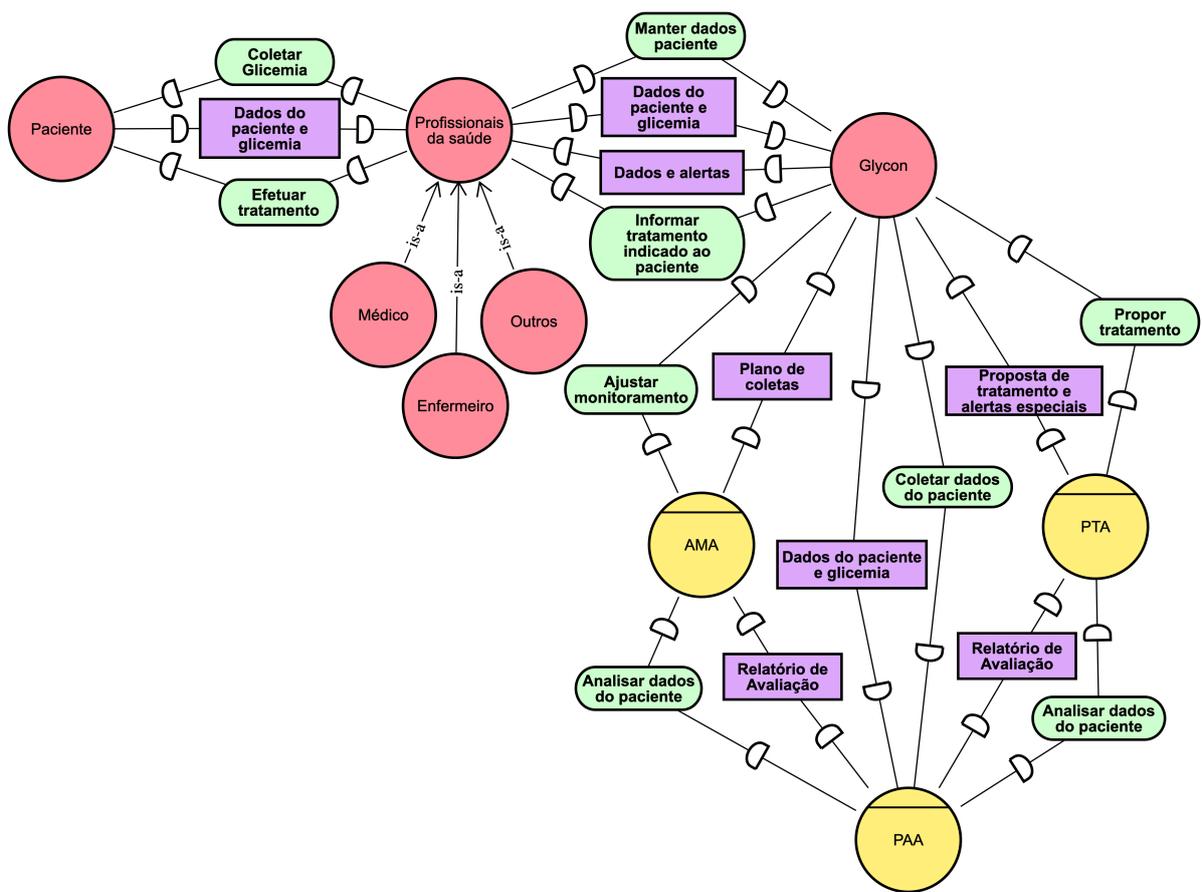


Figura 4.3: Diagrama Tropos referente a fase de requisitos finais.

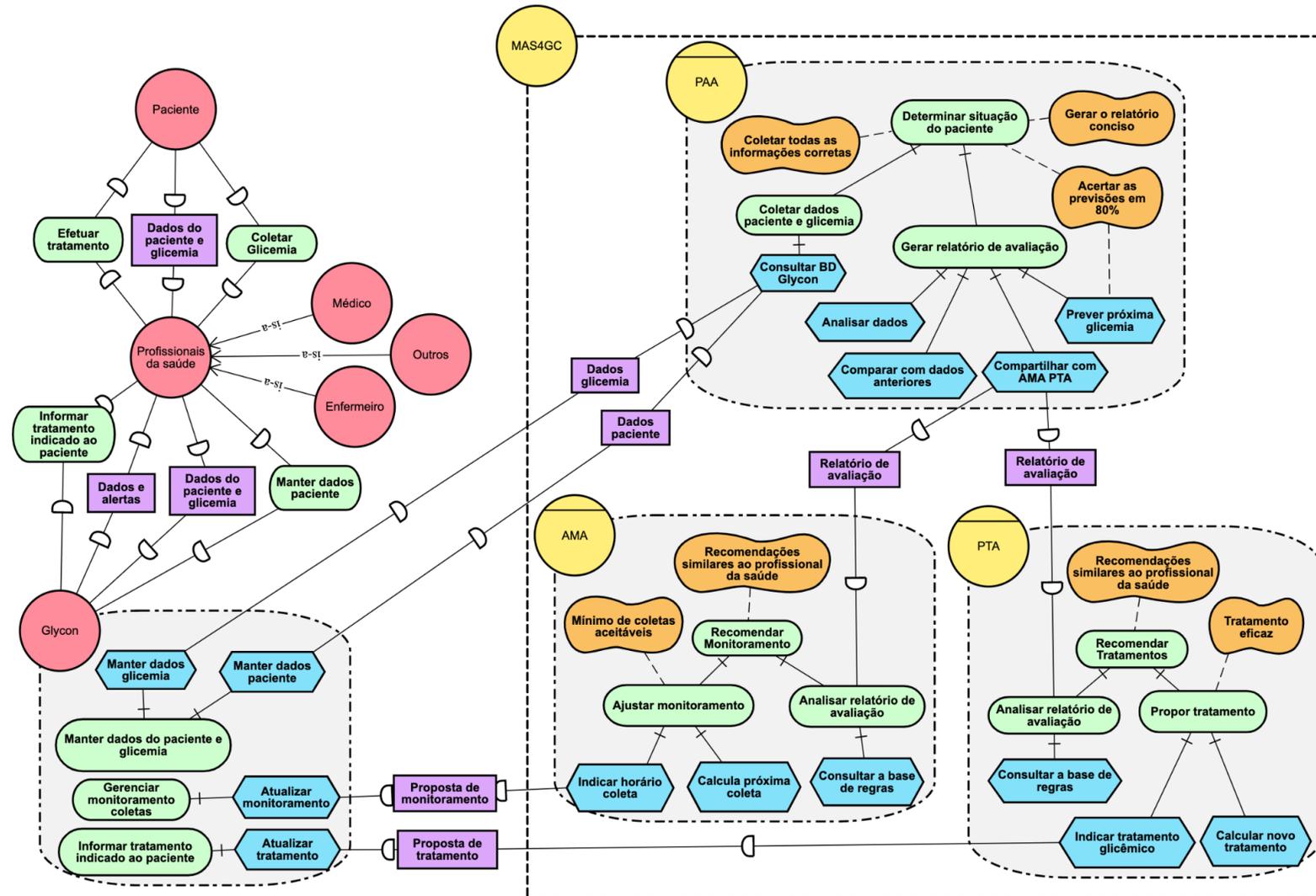


Figura 4.4: Diagrama Tropos referente a fase de projeto arquitetural.

4.2.4 Projeto detalhado

Essa etapa tem como objetivo especificar os recursos e interações entre os agentes. Considera-se que a plataforma de implementação já tenha sido escolhida para permitir um melhor nível de detalhamento. Para isso foram usados os diagramas UML de sequência de eventos e de atividades. Nesse ponto o detalhamento do modelo permitirá que os diagramas sejam mapeado diretamente para o código.

Diagrama de sequência de eventos

A Figura 4.5 apresenta as trocas de mensagens que vão ocorrer entre o paciente e profissionais de saúde, os profissionais de saúde e o sistema Glycon, bem como entre os agentes PAA, PTA e AMA.

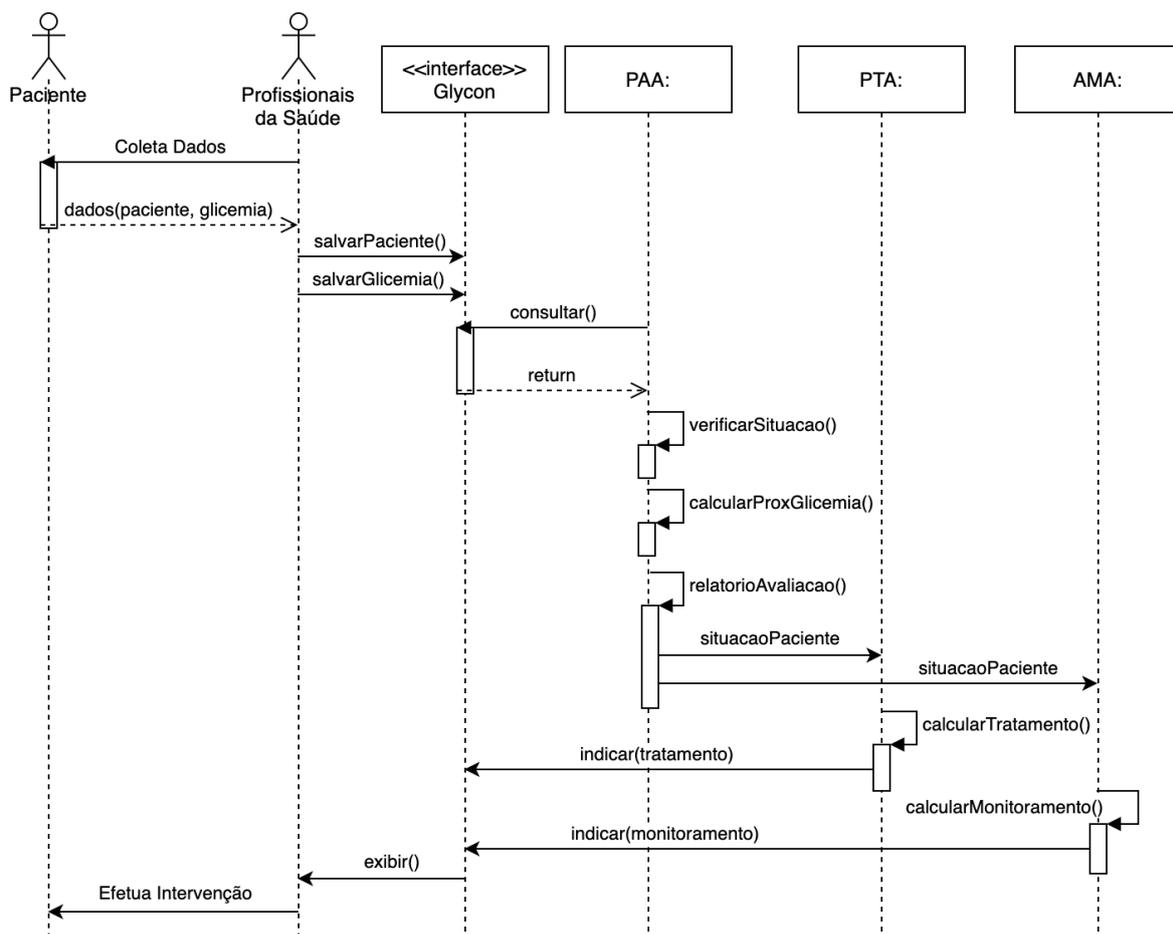


Figura 4.5: Diagrama de sequência referente a fase de projeto detalhado.

Diagrama de atividades

Foi elaborado um diagrama de atividades para cada um dos agentes PAA, PTA e AMA (Figura 4.6). O diagrama apresenta o fluxo de atividades que deve ser executado por cada um dos agentes para concluir seus respectivos objetivos: Determinar situação do paciente - PAA; Recomendar Tratamento - PTA e Recomendar Monitoramento - AMA.

Diagrama de Atividades - PAA

Diagrama de Atividades - PTA

Diagrama de Atividades - AMA

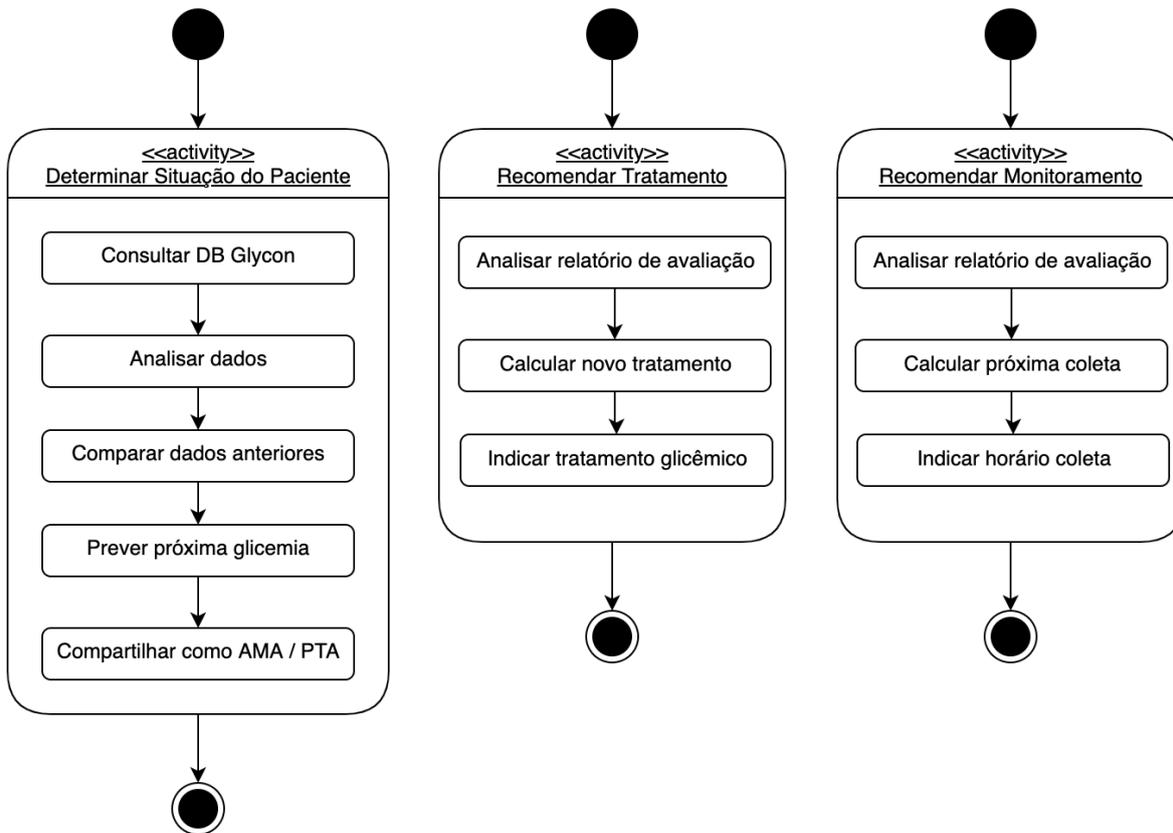


Figura 4.6: Diagrama de atividades referente a fase de projeto detalhado.

4.2.5 Diagramas de implementação

Na fase de implementação novamente os diagramas UML são necessários pois representam como a aplicação será desenvolvida. O diagrama de classes foi usado para mostrar como as classes da aplicação foram estruturadas. Será apresentada também uma representação gráfica de como a base de dados desse sistema foi implementada por meio do arquivo JSON de uma coleção.

Diagrama de Classes

A Figura 4.7 representa as classes utilizadas para classificar os objetos envolvidos no modelo bem como suas características, por meio dos atributos e métodos. O relacionamento entre as classes pode ser visto através das associações. No modelo foram utilizadas quatro classes, onde a classe *Paciente* recebe informações referentes ao paciente no momento da internação, podendo as mesmas serem atualizadas durante a permanência do paciente na UTI. Contém dados pessoais de saúde em geral e que podem interferir na sua situação glicêmica, além de um plano inicial de coletas das glicemias.

A classe *Glicemia* reúne atributos referente as coletas das glicemias, incluindo principalmente o valor da glicemia. A classe *Aplicação* mantém dados relacionados as doses de

glicose ou insulina que o paciente recebeu durante a sua internação. A classe *Agentes* é a responsável por modelar os dados que os agentes manipulam e as ações que os agentes tomam durante a internação do paciente como consultas, cálculos e análise de dados, bem como o tratamento recomendado e os horários em que as coletas de glicemia devem acontecer. Para um melhor entendimento dos atributos contidos nas classes, foi elaborado um dicionário de atributos representado pelas tabelas A.1, A.2, A.3, A.4, as quais foram incluídas no Apêndice A.

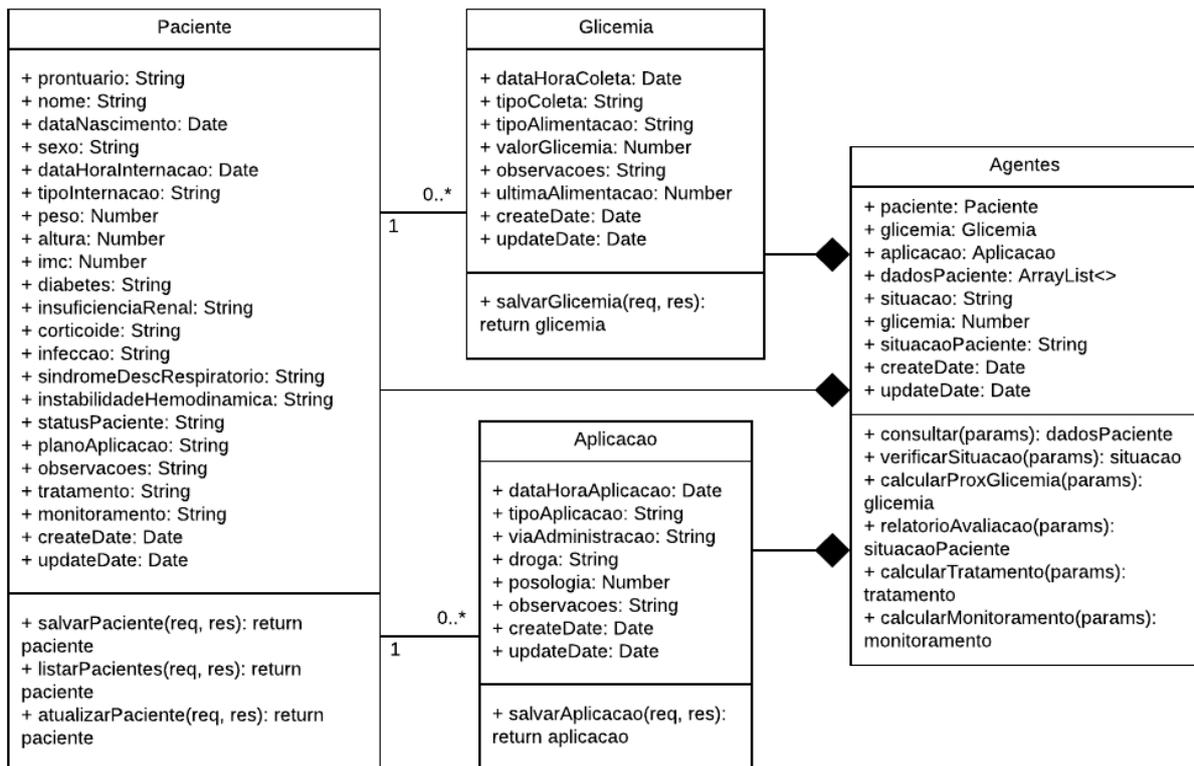


Figura 4.7: Diagrama de classes do sistema MAS4GC.

Base de Dados

Conforme revisão da literatura de Banco de Dados (BD) aplicada à área médica, realizada por Barretto and Ito (2016), os BDs não relacionais (*Not Only SQL* - NoSQL) orientados a documentos e colunas são os mais indicados para dados médicos, devido a existência de distintos tipos, diversidade, sequência temporal e a seu crescente volume de itens registrados. Além disso, os BDs NoSQL permitem produtividade no desenvolvimento de aplicações por utilizarem estruturas de dados flexíveis, serem capazes de gerenciar um conjunto de dados em larga escala em estruturas distribuídas, tornando-se uma excelente alternativa para implementação de sistemas na área da saúde.

As informações manipuladas pelas classes do MAS4GC serão armazenadas em um BD NoSQL por meio do Sistema Gerenciador de Banco de Dados (SGBD) MongoDB. O MongoDB é orientado a documentos e a estrutura organizacional dos dados gera um documento do tipo JSON contendo chaves e valores onde os atributos e respectivos dados

ficam estruturados (MongoDB, 2021). A Figura 4.8 apresenta uma ilustração da estrutura do documento JSON gerado pelo MAS4GC.

```

  _id: 7
  ✓ glicemia: Array
    > 0: Object
    ✓ 1: Object
      dataHoraColeta: "2021-06-02 16:00"
      tipoColeta: "Capilar"
      tipoAlimentacao: "Zero"
      valorGlicemia: "220"
      observacoes: ""
      ultimaAlimentacao: "0"
  ✓ aplicacao: Array
    ✓ 0: Object
      dataHoraAplicacao: "2021-06-02 12:10"
      tipoAplicacao: "de horario"
      viaAdministracao: "intravenosa"
      droga: "Insulina Regular"
      posologia: "1"
      observacoes: ""
  prontuario: "MAS4GC2021"
  nome: "Tiago Segato"
  dataNascimento: 1986-12-20T00:00:00.000+00:00
  sexo: "Masculino"
  peso: 67
  altura: 171
  tipoInternacao: "Clínica"
  diabetes: "Ignorado"
  insuficienciaRenal: "Ignorado"
  corticoide: "Ignorado"
  infeccao: "Ignorado"
  sindromeDescRespiratorio: "Ignorado"
  instabilidadeHemodinamica: "Ignorado"
  planoAplicacao: "6#12#18"
  observacoes: ""
  dataHoraInternacao: 2021-06-02T10:00:00.000+00:00
  statusPaciente: "internado"
  imc: 22.91303307000445
  createDate: 2021-06-02T13:01:01.735+00:00
  updateDate: 2021-06-02T10:04:39.567+00:00
  __v: 3
  monitoramento: 2021-06-02T20:00:00.000+00:00
  tratamento: "Insulina Regular: 2 UN - SC (Prev. 4h)"

```

Figura 4.8: Estrutura da *collection* paciente.

A *Collection Paciente* representa como os dados do paciente são estruturados. Além dos seus respectivos dados pessoais, nessa coleção também ficam armazenados as suas respectivas glicemias e aplicações. Neste caso *Glicemia* e *Aplicacao* são arrays, dentro do documento *Paciente*, pois podem receber vários valores. Nessa coleção também são apresentados os dados que os agentes vão manipular, como o monitoramento e o tratamento.

4.3 Arquitetura

A Figura 4.9 apresenta a arquitetura do MAS4GC. A arquitetura está estruturada em camadas (*layers*) verticalizadas e pode ser considerada do tipo *single control*, pois possui tanto a entrada quanto a saída do mesmo lado (único sentido).

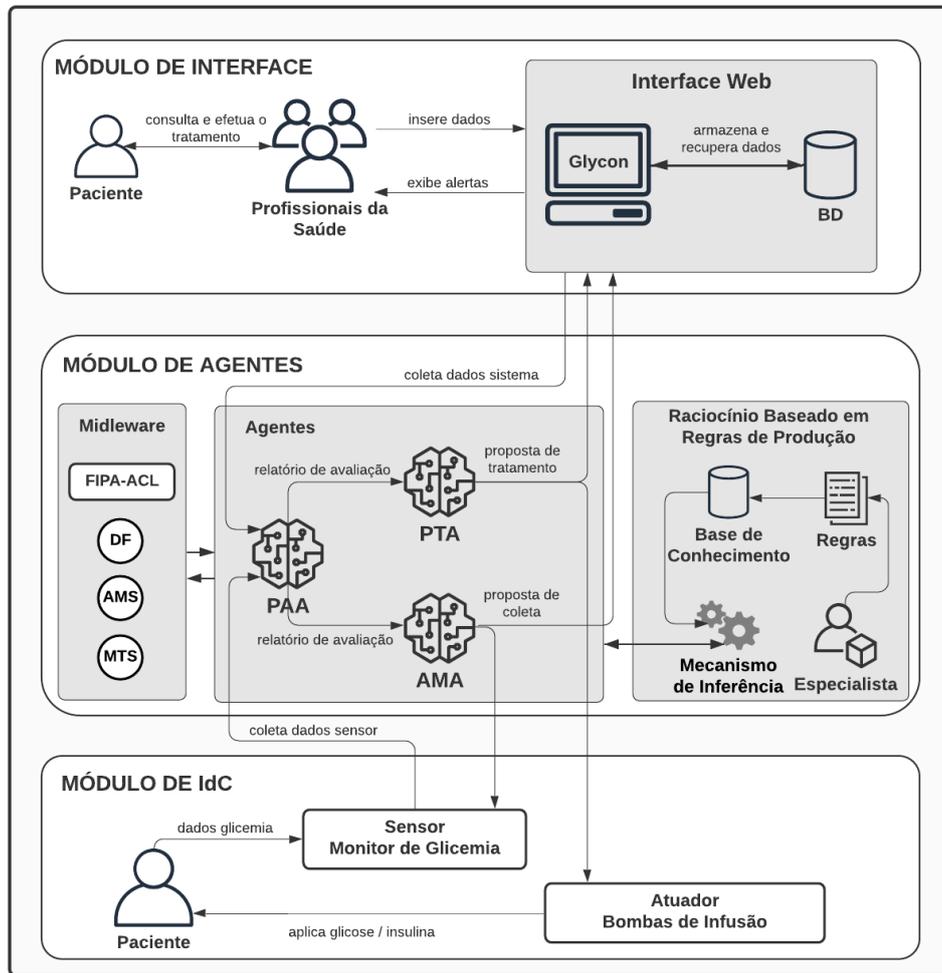


Figura 4.9: Arquitetura do MAS4GC.

No módulo de interface pode-se observar que os dados são recuperados por um profissional da saúde atuante na UTI (médico, enfermeiro, etc.) e são inseridos em um sistema Web com BD. O módulo de agentes abrange o *middleware* que contará com um *framework* de implementação de SMA (PADE) que por meio do protocolo de comunicação no padrão FIPA se comunica com os três agentes definidos.

O agente PAA será o responsável por recuperar as informações da interface Web e enviar para os demais agentes para que desempenhem suas devidas funções. Ele pode enviar os dados para o agente PTA, para que esse possa propor um tratamento adequado de aplicações de glicose e insulina, ou para o agente AMA que identificará se a frequência das coletas precisa ser mais ou menos espaçada ou até mesmo mantida na quantidade de vezes determinado pelo Glycon. Para a elaboração do relatório de avaliação, propostas

de tratamento e indicações de coletas, os agentes deve utilizar a base de conhecimento definida.

Como resposta o agente PTA pode enviar alertas contendo uma recomendação de tratamento (e.g., aplicar x mg/dL de insulina ou manter observação). Já o agente AMA enviará o alerta indicando se o plano de coletas deve ser mantido ou alterado (e.g., efetuar a próxima coleta em “data/hora”). O conhecimento desses agentes se dá por um sistema especialista contendo uma base de conhecimentos construída com auxílio de um médico intensivista e um mecanismo de inferências.

Um terceiro módulo, direcionado a abordagem de Internet das Coisas (IdC), pode ser adicionado à arquitetura mais especificamente com sensores e atuadores para os agentes trazendo maior autonomia à solução. Desta forma, o monitoramento e tratamento poderão ser realizados, por exemplo, utilizando monitores de glicemia e bombas de infusão, respectivamente.

4.4 Implementação

Nesta seção, serão apresentados os detalhes da implementação do sistema MAS4GC. Basicamente esse sistema é composto por três sistemas integrados, sendo um deles uma interface que serve para a entrada e visualização dos dados pelos profissionais de saúde (Glycon). O outro é o próprio SMA, onde, por meio do *framework* PADE foram implementados os três agentes. Por fim, o raciocínio dos agentes foi construído na forma de uma base de regras com o auxílio da biblioteca Experta.

4.4.1 Interface Web Glycon

O sistema de gerenciamento glicêmico do paciente denominado *Glicemic Control On-line* (Glycon) é um sistema web que tem como principal objetivo receber os dados do paciente e sua glicemia (Rodrigues et al., 2021). Além disso, os profissionais de saúde tem a possibilidade de fazer a manutenção desses dados (inserir, atualizar e listar). Porém as duas tarefas mais relevantes do sistema são a de registrar as glicemias e as possíveis aplicações de glicose ou insulina que o paciente pode receber. Tais informações são listadas e apresentadas por meio de gráficos em uma espécie de *dashboard* do paciente conforme apresentado na Figura 4.10. Outra funcionalidade importante é receber e exibir as recomendações de tratamento e monitoramento para os pacientes, servindo como sugestões de tratamento aos profissionais de saúde. Tais informações são apresentadas na tela inicial da aplicação, onde estão listados todos os pacientes internados na UTI (Figura 4.11)

O Glycon foi desenvolvido por alunos do curso superior de Tecnologia em Sistemas para Internet, do Instituto Federal de Brasília, com participação e sob orientação do autor deste trabalho. No desenvolvimento foi utilizado a linguagem Javascript, mais especificamente os *frameworks* React.js (*front-end*) e Node.js (*back-end*). Sua base de dados utiliza o SGBD MongoDB disponível pela ferramenta Web Atlas. Atualmente está hospedado no *Heroku Cloud Application Platform*¹.

¹Disponível em <https://glycon.herokuapp.com/>.

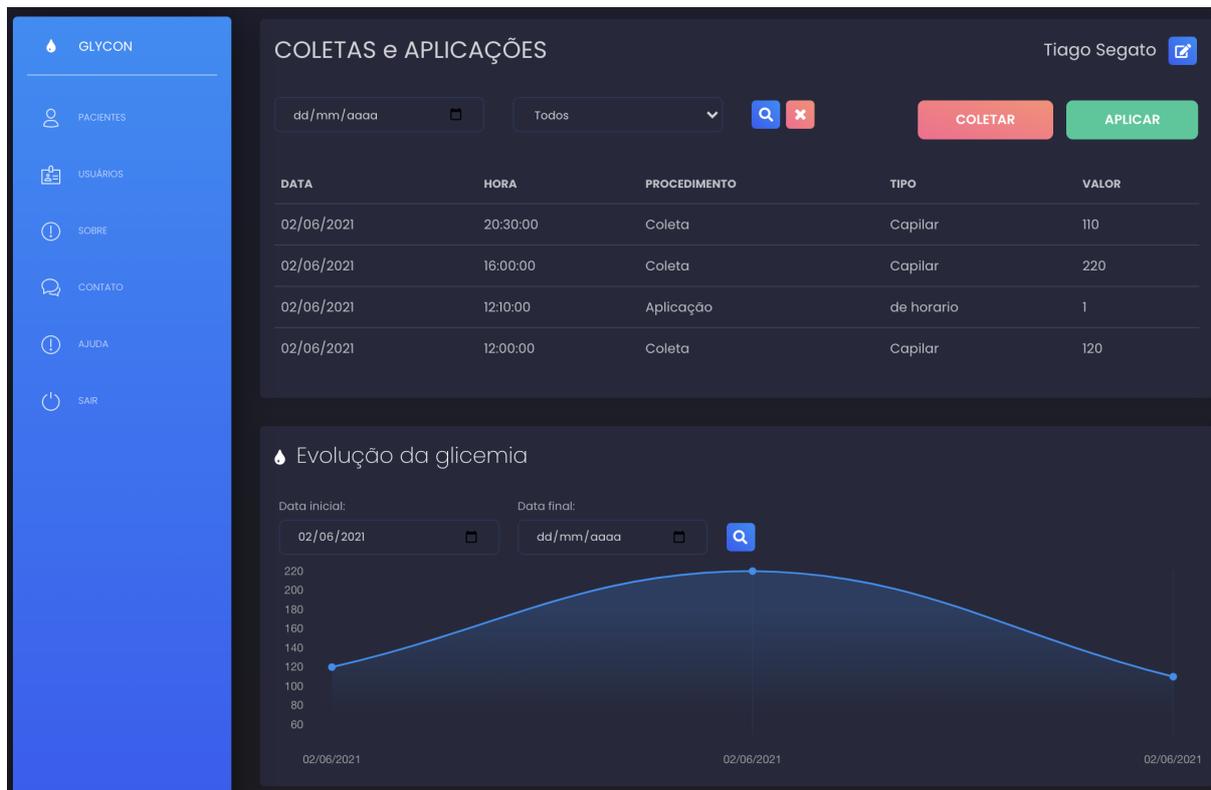


Figura 4.10: *Dashboard* de gerenciamento dos pacientes - Glycon.



Figura 4.11: Tela inicial com a lista de pacientes e respectivas recomendações - Glycon.

Simulador de pacientes e dados glicêmicos

Em complemento ao Glycon foi desenvolvido um simulador com o intuito de ter informações de pacientes e suas respectivas glicemias o mais próximo possível de pacientes reais. A princípio se pretendia implantar o Glycon em uma UTI real (possivelmente na UTI do Hospital do Paranoá) e com auxílio dos médicos intensivistas analisar seu comportamento e resultados. Porém, devido à pandemia da COVID-19 e conseqüentemente a superlotação das UTI e sobrecarga de trabalho dos profissionais de saúde, tal proposta se tornou inviável.

Desta forma, foi gerado um simulador de pacientes sob a supervisão e orientação do médico intensivista Dr. Sérgio Fernandes. Os pacientes são gerados através das informações do desvio padrão de cidadãos brasileiros (IBGE, 2020). Para simular um paciente primeiro é sorteado um grupo, os grupos possuem as informações contidas na Tabela 4.4.

Tabela 4.4: Dados do grupo usados pelo simulador.

Dados	Descrição	Exemplo
Sexo	sexo do paciente	feminino
Faixa de início	valor inicial da faixa de idade	20
Faixa de fim	valor final da faixa de idade	24
Frequência	frequência em que o grupo pode ser selecionado	0.059
Frequência Cumulativa	soma da frequência do grupo com os grupos anteriores	0.059
Média de IMC	média índice de massa corporal	23
Desvio Padrão do IMC	desvio padrão do índice de massa corporal	2.5
Altura média	média das alturas	162
Desvio Padrão da altura	desvio padrão das alturas	6.4
Diabetes	probabilidade do paciente ser portador de diabetes	0.005

Após sortear o grupo são calculados os principais atributos do paciente, sendo eles altura (Bastos et al. (2009); Silva and Monteiro (2009)), *Índice de Massa corporal* (IMC) (Lima et al. (2015); Cervi et al. (1998); Anjos et al. (1998)), peso, idade, sexo. (IBGE (2020)), e se o paciente é ou não portador de diabetes ((Melmed et al., 2020)). Os atributos que possuem média e desvio padrão foram calculados com ajuda da biblioteca SciPy. Já os atributos que possuem apenas a frequência foram calculados através de uma randomização simples, por meio da função *random()* da linguagem Python.

A partir dos principais atributos do paciente são calculados outros que são fundamentais para a simulação do paciente com semelhança a um paciente real: volemia (volume sanguíneo), temperatura e a glicemia inicial do paciente. Também são definidos outros atributos úteis para a manipulação dos dados do paciente como, por exemplo, data de criação e atualização do paciente no banco de dados. E para caracterização do paciente outras informações necessárias para o sistema são criadas como nome, prontuário e tipo de internação. A Figura 4.12 apresenta um exemplo dos dados de um paciente gerado pelo simulador.

```

    _id: ObjectId("60ddd466186111cc9e816a2a")
    nome: "Nete Simas"
    sexo: "M"
    dataNascimento: 1948-10-04T10:29:41.353+00:00
    altura: 183.54
    imc: 24.24
    peso: 81.66
    diabetes: false
  > glicemia: Array
    dataHoraInternacao: 2021-06-30T14:42:46.964+00:00
    createDate: 2021-07-01T14:42:46.964+00:00
    updateDate: 2021-07-01T14:42:46.964+00:00
    prontuario: "OHURXR"
    tipoInternacao: "Clínica"
    insuficienciaRenal: "Ignorado"
    corticoide: "Ignorado"
    infeccao: "Ignorado"
    sindromeDescRespiratorio: "Ignorado"
    instabilidadeHemodinamica: "Ignorado"
    statusPaciente: "internado"
    planoAplicacao: "6#12#18"
    observacoes: ""
  > aplicacao: Array

```

Figura 4.12: Dados de paciente gerado pelo simulador.

4.4.2 Framework PADE

Para o desenvolvimento do SMA foi utilizado o *framework* PADE, que além de facilitar o desenvolvimento, execução e gerenciamento dos agentes em ambientes de computação distribuída, utiliza o protocolo de comunicação FIPA auxiliando na troca de mensagem entre os agentes. A Listagem 4.1 apresenta um trecho de código onde o agente PTA cria e envia uma requisição de dados para o agente PAA. A Listagem 4.2 apresenta o PAA recebendo tal solicitação e desempenha a função que gera o relatório de avaliação do paciente.

Listing 4.1: Requisição de dados do agente PTA para o PAA

```

class PTAgent(Agent):
    def __init__(self, aid, paa_name):
        super(PTAgent, self).__init__(aid=aid)

        message = ACLMessage(ACLMessage.REQUEST)
        message.set_protocol(ACLMessage.FIPA_REQUEST_PROTOCOL)
        message.add_receiver(AID(name=paa_name))
        message.set_content('Possui_Nova_Coleta?')

        self.comport_request = CompRequest(self, message)
        self.comport_temp = ComportTemporal(self, 10.0, message)

        self.behaviours.append(self.comport_request)
        self.behaviours.append(self.comport_temp)

```

Listing 4.2: PAA recebe a solicitação e execução a função relatorioAvaliacao()

```

def handle_request(self, message):
    super(CompRequest, self).handle_request(message)
    display_message(self.agent.aid.localname,
                    message.content)

    situacaoPaciente = relatorioAvaliacao(self)

    reply = message.create_reply()
    reply.set_performative(ACLMessage.INFORM)
    reply.set_content(situacaoPaciente)
    self.agent.send(reply)

```

Um dos fatores que contribuíram na escolha do PADE como *framework* foi ele ser escrito em Python, já que o restante da aplicação também foi codificando usando esta linguagem. Python é uma linguagem simples e de fácil utilização, permite desenvolvimento orientado a objetos e possui uma vasta gama de bibliotecas de códigos, principalmente na área estatística. Tais bibliotecas contribuíram muito no desenvolvimento das funcionalidades do agente PAA, no que se refere a implementação da regressão linear e o cálculo preditivo. Para esse fim, foram utilizadas as bibliotecas numpy, pandas e scikit-learn. Outra biblioteca utilizada foi o pymongo, que atua na conexão entre o Python e o banco de dados MongoDB, além da biblioteca json que permitiu manipular os arquivos do banco de dados trazendo produtividade a implementação.

A Figura 4.13 apresenta a troca de mensagens entre os três agentes em tempo de execução, bem como a resolução dos cálculos do modelo de predição e as recomendações de tratamento e monitoramento. Em vermelho foram enumeradas algumas linhas que serão detalhadas em seguida:

```

1 [PAA_20000] 02/06/2021 16:45:33.522 -> Possui Nova Coleta?

```

	Paciente	Sexo	IMC	Diabetes	Tempo	Hora	Alimentacao	UltimaGlicemia	Glicemia
0	1	0	26	0	6	18	2	83	99
1	1	0	26	0	12	0	2	99	118
2	1	0	26	0	18	6	8	118	76
3	1	0	26	0	24	12	2	76	103
4	1	0	26	0	30	18	2	103	95
...
1445	50	0	21	0	150	18	2	96	82
1446	50	0	21	0	156	0	2	82	89
1447	50	0	21	0	162	6	8	89	94
1448	50	0	21	0	168	12	2	94	96
1449	50	0	21	0	174	18	2	96	100

```

[1450 rows x 9 columns]
3 Previsão de Glicemia para 4 hs é de: [140.36544458] com acurácia de: 0.90
4 [AMA_30000] 02/06/2021 16:45:33.990 -> {'ID': 7, 'Paciente': 'Tiago Segato', 'Situacao': 'prevgAlvo',
'DataHora': datetime.datetime(2021, 6, 2, 20, 30)}
5 Próxima coleta: 2021-06-03 04:30:00
6 [PTA_10000] 02/06/2021 16:45:34.197 -> {'ID': 7, 'Paciente': 'Tiago Segato', 'Situacao': 'prevgAlvo',
'DataHora': datetime.datetime(2021, 6, 2, 20, 30)}
7 Manter Observação (Prev. 4h)

```

Figura 4.13: Execução da troca de mensagem ente os agentes.

- A linha indicada pelo número 1 mostra o momento em que o agente PAA recebe uma mensagem do agente PTA questionando se há pacientes com novas coletas de

glicemias. Como já foi descrito o agente PAA faz a consulta na base de dados, faz os cálculos necessários e apresenta-os na linha 3.

- O número 2 exibe o *dataset* que contém dados de outros pacientes que são usados para aumentar a base de conhecimento dos agentes e tornar o modelo preditivo melhor.
- O número 3 apresenta a predição do valor da glicemia para daqui a 4 horas (140.36 mg/dL) e a acurácia, calculado com base no modelo preditivo com 90% de acerto na predição.
- Após os cálculos, o Agente PAA envia o relatório de avaliação para o agente AMA e para o PTA, itens 4 e 6 respectivamente.
- O item 5 apresenta a recomendação de monitoramento feita pelo agente AMA, indicando o momento (data e hora) recomendado para que seja feita a próxima glicemia do paciente em questão.
- E por fim, o item 7 apresenta a recomendação de tratamento feita pelo agente PTA, onde nesse caso é apenas manter o paciente em observação, pois considera a glicemia do paciente em 4 horas como estável, ou seja, dentro da faixa alvo estipulada.

4.4.3 Biblioteca Experta

Para um melhor entendimento das regras com base no conhecimento médico foram elaborados três diagramas contendo os fluxos das decisões que são tomadas pelo especialista dependendo das ações de cada agente.

A Figura 4.14 representa o conhecimento do agente PAA que tem como principal objetivo determinar a situação do paciente. O PAA envia os dados referente a situação do paciente para os agentes AMA e PTA para que estes ajustem o monitoramento ou proponham um tratamento, respectivamente. Primeiramente é verificado se a glicemia coletada é a primeira, caso seja, é verificado a sua faixa e o procedimento padrão com base na regra básica (Tabelas 2.1 e 2.2). Caso já existam outros valores de glicemia coletados, o agente será responsável por fazer a comparação (por meio de uma RLM) e gerar uma predição de qual será a próxima glicemia podendo então encaminhar esse dado aos demais agentes. Esses agentes vão fazer os ajustes de melhor tratamento e frequência de coletas para o paciente.

Como dito anteriormente, esse agente não faz recomendações de tratamento ou monitoramento, ou seja, ele apenas verifica os dados do paciente e identifica a sua situação glicêmica, portanto não foi necessário implementar uma base de regras para tal.

Nos casos onde houve mais de uma coleta e há necessidade de um ajuste na periodicidade das coletas das glicemias ou que haja necessidade de se elaborar um tratamento adequado ao paciente, a situação atual do paciente será enviada para os agentes PTA e AMA, para que eles possam desempenhar seus respectivos papéis.

A Figura 4.15 representa as ações do agente PTA. Neste caso, primeiramente ele deve verificar qual a situação glicêmica do paciente enviada pelo PAA, analisar qual é a tendência e de acordo com o seu valor calcular qual o tratamento mais indicado entre as opções: aplicar glicose, aplicar insulina ou simplesmente manter o paciente sob observação.

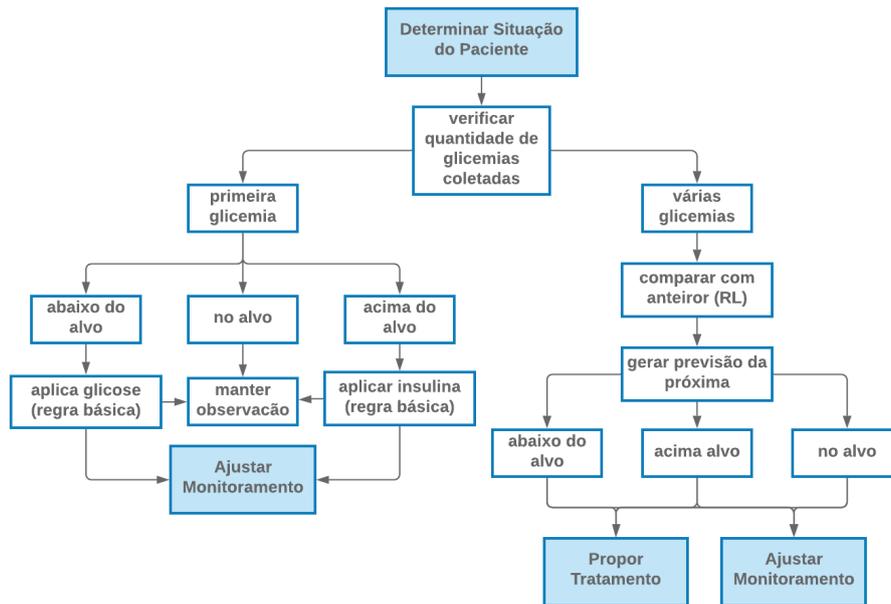


Figura 4.14: Diagrama de representação do conhecimento do agente PAA.

Por exemplo, em um caso onde o paciente esteja internado há alguns dias e consequentemente outras glicemias tenham sido coletadas, o agente deve efetuar uma comparação com a glicemia anterior e calcular qual é a previsão para sua próxima glicemia, para que seja indicado o tratamento mais adequado. Pode acontecer, por exemplo, que mesmo a glicemia estando fora da faixa alvo (hipoL) a indicação seja apenas manter observação, pois em um momento anterior a glicemia poderia estar em uma faixa mais grave (hipoG). Neste caso, a previsão indica que a glicemia pode estar voltando a normalidade e se aproximando da faixa alvo (hAlvo).

A Listagem 4.3 apresenta a regra construída por meio do *framework* Experta que seria executada nessa situação.

Listing 4.3: Exemplo de regra para uma recomendação de tratamento

```

@Rule(AND(BloodGlucose( situacao='gAlvo' ),
BloodGlucose(idPaciente=MATCH.idPaciente)))
def bg_gAlvo(self , idPaciente):
    tratamento = "Manter_Observacao"
    response = connection.collection.update_one({ "_id": idPaciente },
    { "$set": { "tratamento": tratamento } })
  
```

Em suma a regra é composta por uma condição e uma função. Onde, a anotação @Rule, contém as condições necessárias, nesse caso é verificado a situação do paciente que é igual a 'hAlvo'. A função é uma função Python padrão que recebe as informações necessárias e contempla o tratamento adequado. Além disso, o agente executa a inserção ou atualização da recomendação de tratamento na base de dados compartilhada pelo Glycon, de modo que essa informação apareça visualmente para os profissionais de saúde em forma de um alerta.

A Figura 4.16 ilustra o procedimento que o agente AMA deve desempenhar, que neste caso é basicamente com base na situação determinado pelo PAA. Verificar em qual faixa

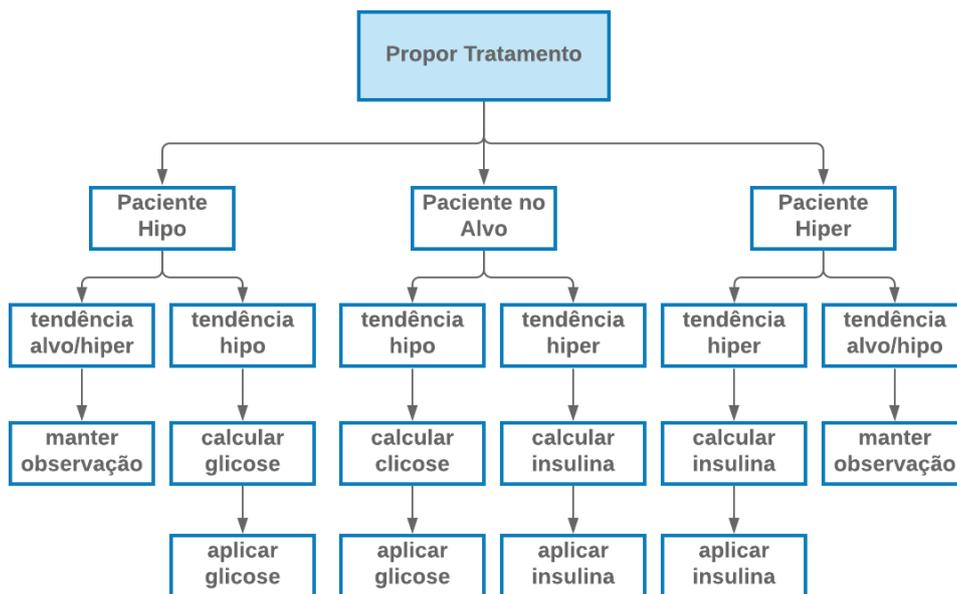


Figura 4.15: Diagrama de representação do conhecimento do agente PTA.

a glicemia do paciente se encontra, fazer o cálculo e sugerir ao sistema que a frequência das coletas seja aumentada ou diminuída.

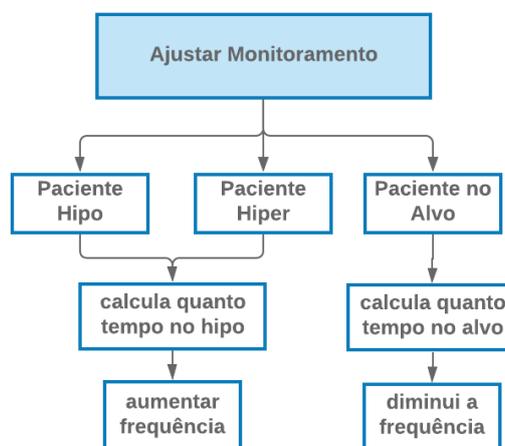


Figura 4.16: Diagrama de representação do conhecimento do agente AMA.

Como exemplo considere um paciente que possui várias coletas e a previsão é que sua glicemia continue estável pelas próximas 4 horas, neste caso a regra apresentada na Listagem 4.4 será executada.

Listing 4.4: Exemplo de regra para uma recomendação de monitoramento

```

@Rule(AND(BloodGlucose(glicemia=MATCH.glicemia),
TEST(lambda glicemia: glicemia > 70 and glicemia <= 140),
BloodGlucose(idPaciente=MATCH.idPaciente),
BloodGlucose(dataHora=MATCH.dataHora)))
def cm_rule5(self, idPaciente, dataHora):
    monitoramento = dataHora + timedelta(hours=8)
  
```

```
response = connection.collection.update_one({ "_id": idPaciente },  
{ "$set": { "monitoramento": monitoramento } })
```

Na anotação da regra na Listagem 4.4, além de receber os dados é feito uma comparação para identificar em qual intervalo a glicemia se encontra e em que hora foi coletada. Nesse caso, a glicemia está estável e a indicação de nova coleta pode ser espaçada para 8 horas. Essa informação é acrescido da data e hora atual e inserido na base de dados do Glycon para que seja visualizado pelos profissionais de saúde na interface gráfica. A princípio 25 regras foram implementadas usando o Experta e representam o conhecimento dos agentes PTA e AMA.

Capítulo 5

Experimentos

Com o intuito de validar a proposta do MAS4GC foram realizados dois experimentos. Visando manter as glicemias dos pacientes na faixa alvo pelo maior tempo durante a internação do paciente na UTI, o tratamento preventivo e as recomendação dos agentes foram considerados. Para validar o modelo de predição do MAS4GC foi implementado um modelo baseado em regressão linear simples e múltipla (RLS e RLM). Para validar as recomendações dos agentes baseados em RBR foi utilizado uma base de regras de produção definida utilizando o conhecimento do médico intensivista Dr. Sérgio Fernandes.

O Experimento 1 visa definir qual o modelo de regressão (RLS ou RLM) é mais indicado para prever a glicemia dos pacientes internados em UTI, além de avaliar a acurácia das predições. Sendo assim, o Experimento 1 está relacionado à questão de pesquisa (QP1) apresentada no Capítulo 1. Já o Experimento 2 avalia a performance dos agentes em fazer recomendação de tratamento e monitoramento glicêmico dos pacientes internados em UTI. Mais especificamente avalia o quanto esse resultado é semelhante a diferentes profissionais de saúde, bem como a um médico intensivista, especialista em UTI. Sendo assim, o Experimento 2 está relacionado à questão de pesquisa (QP2) (vide Capítulo 1).

A Seção 5.1 apresenta o *dataset* utilizado nos experimentos. O detalhamento dos dois experimentos, bem como seus resultados serão apresentados nas Seções 5.2 e 5.3, respectivamente.

5.1 Definição do *dataset*

Embora tenha sido implementado um simulador de pacientes, junto a interface Glycon, apresentado no Capítulo 4, este gera uma única glicemia por paciente o que viabiliza apenas testes de predição em tempo de execução. Ou seja, sempre que for necessário inserir um paciente novo no Glycon para testes, o simulador fornecerá dados sintéticos acerca de um paciente, permitindo que não se insiram dados totalmente aleatórios e a predição seja mais próxima possível de uma situação real. Surgiu então a necessidade de se ter uma base contendo diversos pacientes com vários dados glicêmicos representando a glicemia do paciente ao longo do dia, além de outros dados específicos de pacientes para uso no MAS4GC.

Alguns *datasets* foram encontrados, porém não os consideramos como ideais para a elaboração dos testes. Um deles foi o *Medical Information Mart for Intensive Care* (MI-

MIC) ¹, apesar de conter diversos dados acerca de pacientes em UTI, em sua versão gratuita, à qual tivemos acesso, não continha os dados necessários para nossos experimentos como os dados das glicemias do paciente ao longo da sua internação.

Desta forma, com o auxílio do médico especialista foram aproveitadas algumas das fórmulas de geração de dados de pacientes usados no simulador presente no Glycon e adicionadas em uma planilha eletrônica (Excel). Mediante alguns ajustes (i.e., adição de informações como data e hora inicial e frequência de coletas), foi possível gerar randomicamente dados de pacientes e diversas glicemias, com base nas distribuições esperadas para a população local das UTIs de Brasília, simulando episódios glicêmicos próximos a pacientes reais. Foram usadas fórmulas de predição baseadas em distribuições fisiológicas gerando pacientes como o ilustrado na Figura 5.1.

	A	B	C
1	Nome	Micael Damiano	
2	Sexo	M	
3	Idade	71	
4	Altura	173,7	
5	IMC	22,92	
6	Peso	69,13	
7	Diabetes	Não	
8	Glicemia Basal(med)	80,0	
9	Glicemia Basal(DP)	8,0	
10	Freq Glicemias(h)	6,0	4
11	Data e hora inicial	01/01/2021 12:00	
12	Carga Alimentar(Pico)	21,68	
13	Carga Alimentar(DP)	4,61	
14	Carga DM	1	

Figura 5.1: Dados de um paciente gerados na planilha eletrônica.

Com o simulador é possível gerar dados mais gerais como nome, sexo, idade, altura, IMC, peso e diabetes, por mais que pareçam ser simples dados gerados aleatoriamente (por meio das funções ALEATÓRIO() ou ALEATÓRIOENTRE()), devem fazer sentido no contexto que serão aplicados. Por exemplo, o sexo e a idade são obtidos pela pirâmide etária da população brasileira de adultos (IBGE, 2020). E a altura usa uma distribuição normal da população brasileira adulta (Bastos et al. (2009); Silva and Monteiro (2009)). O IMC usa uma distribuição beta e o peso é obtido pelo IMC e altura (Lima et al. (2015); Cervi et al. (1998); Anjos et al. (1998)). Para definir o diabetes foi usada uma prevalência próxima de 20%, que é o esperado pela população em geral, porém com mais chances de se ter o diabetes conforme maior a idade (Melmed et al., 2020).

Além desses dados mais gerais, dados acerca da glicemia do paciente também são gerados e assim como os demais, seguem uma lógica para que faça sentido aos dados do paciente gerados anteriormente (sexo, idade, altura, etc.). Tais glicemias são obtidas aleatoriamente dentro de faixas normais ou não normais conforme são ou não diabéticos (Melmed et al., 2020).

¹<https://physionet.org/content/mimiciiii-demo/1.4/>

Entre esses são gerados dados como média (Med) e desvio padrão (DP) da glicemia basal (em jejum), a frequência de glicemias geradas em horas, data e hora inicial da glicemia gerada, a carga alimentar (Pico, quanto de glicose é aumentado na circulação após uma alimentação) e seu desvio padrão, além da carga do *Data Mining* (DM), que representa o quanto a mais de glicose é mantido na circulação devido a reação da insulina para o paciente portador de diabetes (Melmed et al., 2020). Os itens descritos foram inseridos em uma curva de distribuição normal, simulando a distribuição de uma UTI mediana. Os dados sintéticos de pacientes gerados neste trabalho se apresentaram suficientes para os experimentos realizados, no entanto, novos dados podem ser adicionados ao *dataset* conforme sugestões de médicos intensivistas para investigações futuras.

A título de exemplo a fórmula a seguir apresenta como é gerada a média da glicemia basal do paciente, onde a célula B7 representa se o paciente tem ou não diabetes. Todas as fórmulas usadas nessa planilha podem ser encontradas no Apêndice B.

$SE(B7="Não";ALEATÓRIOENTRE(80;90);ALEATÓRIOENTRE(90;130))$

A cada nova execução das fórmulas que contém as funções randômicas um novo paciente é gerado contendo as suas respectivas glicemias e demais valores. A Figura 5.2 apresenta os dados de um paciente simulado na planilha eletrônica.

	A	B	C	D	E	F	G	H	I	J	K
1	Nome	Sexo	Idade	Altura	IMC	Peso	Diabetes	Data hora	Hora do dia	Alimentação	Glicemia
2	Micael Damiano	M	71	174	22,9	69,1	Não	01/01/2021 12:00	12	2	89,3
3	Micael Damiano	M	71	174	22,9	69,1	Não	01/01/2021 18:00	18	2	96,2
4	Micael Damiano	M	71	174	22,9	69,1	Não	02/01/2021 00:00	0	2	84,8
5	Micael Damiano	M	71	174	22,9	69,1	Não	02/01/2021 06:00	6	8	85,2
6	Micael Damiano	M	71	174	22,9	69,1	Não	02/01/2021 12:00	12	2	102,8
7	Micael Damiano	M	71	174	22,9	69,1	Não	02/01/2021 18:00	18	2	96,8
8	Micael Damiano	M	71	174	22,9	69,1	Não	03/01/2021 00:00	0	2	87,4
9	Micael Damiano	M	71	174	22,9	69,1	Não	03/01/2021 06:00	6	8	97,7
10	Micael Damiano	M	71	174	22,9	69,1	Não	03/01/2021 12:00	12	2	86,8

Figura 5.2: Ilustração de dados de um paciente simulado.

Para a efetiva construção do *dataset* o processo de gerar os pacientes foi executado 50 vezes e as primeiras 30 linhas de registro foram copiadas e coladas em uma nova planilha que consolidou a base para ser utilizada como o *dataset* dos experimentos. Os dados usados nesse simulador são base de um artigo em desenvolvimento pelo Dr. Sérgio Fernandes para publicação futura.

5.2 Experimento 1

O Experimento 1 tem como objetivo identificar qual tipo de regressão é mais apropriado para prever o próximo evento glicêmico de um paciente, bem como identificar quais as variáveis independentes são mais relevantes para o modelo. Definidos o modelo de regressão adequado e as variáveis independentes, objetiva-se aferir se o modelo apresenta um resultado satisfatório para predição do próximo evento glicêmico de pacientes. Os

modelos de regressão testados foram a RLS e RLM. O *workflow* da Figura 5.3 ilustra a sistematização metodológica do experimento, o qual foi descrito por meio de tópicos numerados e apresentados na sequência.

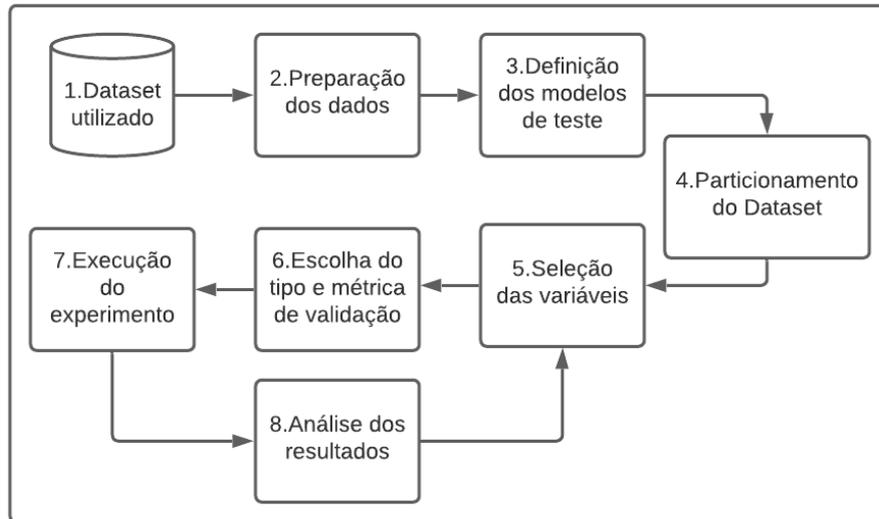


Figura 5.3: *Workflow* metodológico do experimento.

1. *Dataset* utilizado: contém 50 pacientes simulados, com 30 coletas de glicemia para cada paciente. As informações presentes nesse *dataset* são: paciente, sexo, idade, altura, peso, IMC, diabetes (se tem, não tem, ou se foi ignorado), tempo (baseado na data e hora do dia em que a coleta da glicemia foi efetuada), hora do dia, última alimentação (realizada a quantas horas) e glicemia (valor da glicemia atual).
2. Preparação dos dados: alguns valores disponíveis no *dataset* são variáveis categóricas, e estes não são ideais para a regressão, por isso foi necessário realizar alguns ajustes. Do paciente foi utilizado um código numérico sequencial iniciado em 1, para sexo foi definido 0 para feminino e 1 para masculino. Os valores 0, 1 e 2 foram usados para representar quem não tem diabetes, teve essa informação ignorada ou tem diabetes, respectivamente. O campo tempo recebeu uma escala de valores de acordo com a diferença em horas na qual ocorreram as coletas e o campo hora do dia, recebeu apenas o valor que representa as horas, descartando os minutos e segundos. Os demais valores são numéricos por padrão e foram mantidos, exceto em caso de números com casas decimais que foram aproximados para seus correspondentes em inteiro, por exemplo: IMC de 25,92, passou a ser 26.
3. Definição dos modelos utilizados: serão utilizadas a regressão linear simples e múltipla (RLS e RLM).
4. Particionamento do *dataset*: em ambos os casos, na RLS e na RLM a regressão foi baseada nos valores das 29 primeiras coletas e a predição apontou qual deveria ser o valor da glicemia na 30ª coleta. Então os valores referente a 30ª glicemia dos 50 pacientes foram alocados em uma nova tabela para conferência posterior. Ambas as regressões foram submetidas a uma validação cruzada utilizando o método K -

fold, onde o *dataset* foi dividido em 10 grupos e a acurácia será medida 10 vezes intercalando os conjuntos de treinamento e teste (*10-fold cross validation*).

5. Seleção das variáveis: no caso da RLS serão utilizados apenas as variáveis tempo, como sendo a variável independente e glicemia com a dependente. Na RLM glicemia também é a variável dependente, pois é esse o valor que pretende-se prever e todas as demais variáveis foram usadas como independentes. Posteriormente um novo valor foi acrescentado, a última glicemia (glicemia anterior a que se deseja prever).
6. Escolha do tipo e métrica de validação: após a aplicação das regressões serão calculados os resíduos, a média dos resíduos e desvio padrão para todas as coletas. Será considerado uma margem de segurança de duas vezes o desvio padrão para cima e para baixo (limites inferior e superior). Com base nesses valores estipula-se os limites inferior e superior em ml/dL, de acordo com as predição feitas e verificando quantas vezes o modelo acertou (dentro dessa margem de erro). Também será aplicada a validação cruzada em ambas as regressões com o intuito em aumentar a carga de testes na base de dados utilizada. Novamente o método *K-fold* será aplicado criando-se 10 subgrupos a partir da base atual. Os resultados serão validados por meio das seguintes medidas: coeficiente R^2 , MAE, MAPE e RMSE, conforme apresentado na Seção 2.3.
7. Execução do experimento: todo o processo foi executado utilizando primeiro a RLS e depois a RLM. Iniciou-se calculando a regressão linear e na sequência foi calculada a predição da 30ª glicemia para os 50 pacientes. Na sequência o teste de validação cruzada foi executado para ambas as regressões.
8. Reexecução do experimento: com as regressões e predições executadas os resultados foram analisados e constatou-se que a RLM seria mais viável no contexto aplicado, porém ainda havia como melhorar o modelo adicionando variáveis independentes (no caso foi adicionado o valor da última glicemia). Como a RLM possui diversas variáveis independentes foi feita uma análise nos resultados da regressão, mais especificamente nas variáveis com menor valor de p (valor-p) e foram feitos alguns testes para a escolha dos melhores parâmetros para o modelo. Todo o processo descrito será discutido na análise dos resultados.

5.2.1 Experimento 1: análise dos resultados

Conforme apresentado, após executar a regressão linear simples e múltipla e analisar os resultados notou-se que a RLM teve um resultado melhor. O principal ponto que foi levado em consideração e que inviabilizou o uso da RLS foi o seu desvio padrão alto de 49,63. Fator esse que implica diretamente nos limites inferior e superior a ser considerado de glicemia prevista que seria de 99,60 mg/dL. Isso indica que uma predição por meio desse modelo ficaria longe o bastante do aceitável para o tratamento adequado. Ou seja, não é aceitável que se erre em quase 100 mg/dL do valor da glicemia de um paciente, pois certamente o tratamento não apresentará um resultado adequado.

De acordo com Rath et al. (2020), as limitações da regressão linear incluem o fato de que frequentemente ela explora uma relação entre a média das variáveis de entrada e as de saída. Assim como a média não é uma descrição completa de uma única variável, a RLS

não permite um entendimento claro das relações das variáveis. Portanto, uma análise dos vários fatores (diversas variáveis) por meio da RLM apresenta melhores resultados em alguns contextos, inclusive nesse experimento, onde a variável dependente (glicemia) baseada em muitas variáveis independentes (paciente, sexo, imc, diabetes, alimentação, etc.) tiveram uma correlação maior, diferente da RLS, que leva apenas a variável independente “tempo” em consideração.

A RLM utilizando 10 variáveis independentes apresentou um desvio padrão de 28,65 com o limite de dois pontos de 57,3. Tal valor ainda não é indicado, então buscou-se outra variável que pudesse em conjunto com as demais trazer melhor resultado para o modelo. Foi então que o valor da última glicemia foi adicionado ao modelo. O *insight* que levou a considerá-la foi que na maioria dos casos a glicemia atual, dependendo do horário da sua coleta, não se distancia muito da última coletada.

Como foram executadas várias regressões nesse teste elas foram enumeradas, a RLS e RLM usadas até o momento serão citadas como RLS1 e RLM1, respectivamente. As demais RLM criadas serão apresentadas como RLM2, RLM3 e RLM4.

Todo o processo de regressão, predição e análise foi executado possibilitando uma melhora nos resultados (RLM2). O desvio padrão baixou para 17,2 e os limites ficaram em 34,4, que de acordo com o especialista é um valor aceitável para prosseguir com o tratamento (aplicação de glicose ou insulina). Após ter alcançado um valor aceitável de limite foi feito a contagem de acertos do modelo, verificando quantas das 50 predições realizadas ficaram no intervalo composto pelo limite de 34,4. O resultado foi de 46 vezes, totalizando 92% das vezes.

Com o intuito de melhorar ainda mais o modelo, os mesmos testes foram executados mais duas vezes (RLM3 e RLM4). Neste caso, os valores com menor valor-p foram descartados. No teste RLM3 foram desconsideradas as variáveis idade, altura e peso, mas o resultado foi similar ao anterior de 17,2 e 34,4 para o desvio padrão e limite, respectivamente, com 92% de acerto. Embora as variáveis removidas do teste anterior terem sido as com menor valor de p, um novo teste foi feito (RLM4), considerando, segundo o especialista variáveis que pudessem não ter forte correlação com o modelo. Agora foram desconsideradas sexo e IMC, porém os resultados foram praticamente iguais aos dois anteriores, como pode ser visto na Figura 5.4.

Para prosseguir com os testes foi escolhido o modelo de RLM3, pois obteve o mesmo resultado dos modelos RLM2 e RLM4, porém utilizando menos variáveis independentes. O modelo RLM3 foi comparado novamente com a RLS1, porém usando uma carga de dados maior (10 vezes), por meio de uma validação cruzada (*10-fold*). Os resultados obtidos podem ser observados nos gráficos da Figura 5.5.

Concluindo a análise do Experimento 1, pode-se dizer que o modelo de regressão mais apropriado para prever o próximo evento glicêmico de um paciente foi o RLM3. Esse modelo de RLM apresenta erros considerados baixos pelo especialista alcançando uma acurácia de acertos de 90% das vezes que o MAS4GC recomenda o tratamento ao profissional de saúde para aplicar no paciente.

5.3 Experimento 2

O objetivo do Experimento 2 é comparar as recomendações de tratamento e monitoramento feitas pelo MAS4GC com as dos profissionais de saúde. Seis profissionais de saúde

TESTES				
RLS 1	RLM 1	RLM 2	RLM 3	RLM 4
	Paciente	Paciente	Paciente	Paciente
	Sexo	Sexo	Sexo	
	Idade	Idade		Idade
	Altura	Altura		Altura
	IMC	IMC	IMC	
	Peso	Peso		Peso
	Diabetes	Diabetes	Diabetes	Diabetes
Tempo	Tempo	Tempo	Tempo	Tempo
	Hora do dia	Hora do dia	Hora do dia	Hora do dia
	Alimentacao	Alimentacao	Alimentacao	Alimentacao
		Ultima Glic.	Ultima Glic.	Ultima Glic.

Variáveis	1	10	11	8	9
Desv. Pad	49,63	28,65	17,2	17,2	17,3
Limite	99,26	57,3	34,4	34,4	34,6
Porcent.	--	--	92%	92%	92%

Figura 5.4: Tabela comparativa entre os modelos de regressão utilizados.



Figura 5.5: Resultados da comparação dos modelos de RLS1 e RLM3 utilizando o coeficiente R^2 e medidas de avaliação de erro.

contribuíram para o experimento. Cinco profissionais eram de diferentes especialidades, além de um médico intensivista especialista. Além de diferentes especialidades, tais profissionais também se diferem em relação a tempo de atuação e experiência. Apesar de alguns profissionais não terem formação acadêmica direcionada especificamente ao controle glicêmico de pacientes, os profissionais utilizados neste experimento possuem conhecimento adquirido através de experiência. Além disso, os profissionais se voluntariaram para participar do experimento auxiliando na realização do mesmo, tendo em vista a dificuldade na disponibilidade de médicos especialistas em UTI durante a pandemia. Os profissionais voluntariados foram:

- endocrinologia - residente com atuação de seis anos, além de ter contato diário com estudos relacionados a glicemia.
- nefrologia - residente com atuação de quatro anos e meio.

- neurologia - atua na área há quatro anos, além de um ano em UTI.
- gastroenterologia - atua há 20 anos na área e 22 anos em UTI.
- fisioterapia - tem experiência profissional de 12 anos, bem como um ano atuando em UTI.
- intensivista - atua na área médica há 22 anos, sendo 19 em UTI, com experiência em rotina e chefia de UTI.

A sistematização do experimento contém as mesmas sete etapas do Experimento 1. As Etapas 1, 2 e 5 de *dataset* utilizado, preparação de dados e seleção de variáveis são as mesmas. As demais etapas serão apresentadas na sequência:

- Definição dos modelos utilizados - o conhecimento dos agentes utilizados para fazer as recomendações foi baseado em regras de inferência, enquanto os profissionais de saúde humana utilizaram seu conhecimento, baseado em estudos realizados e experiências adquiridas.
- Particionamento do *dataset* - foram utilizadas as coletas de número 29 (29° das 30 coletas) de cada um dos 50 pacientes, tanto pelo médico intensivista, que analisou os 50 casos, quanto pelos profissionais de saúde de diferentes áreas. Sendo que cada um dos cinco profissionais analisou dez casos.
- Escolha do tipo e métrica de validação - (1) comparação entre as respostas do MAS4GC em relação a dos profissionais de saúde e do especialista; e (2) uma validação de teste diagnóstico.
- Execução do experimento - foram criadas cinco planilhas contendo a explicação do experimento e 10 dos 50 casos. Entende-se por caso os dados do paciente que incluem glicemia entre outras informações, descritos na etapa Seleção de variáveis (item 5 do Experimento 1). Para cada caso, o profissional de saúde foi solicitado a informar: (1) qual o tratamento indicado (com base no Tratamento da Tabela 2.2); e (2) quantas horas a próxima coleta deve acontecer (com base em Monitoramento na Tabela 2.2). O especialista da UTI recebeu planilha semelhante, mas contendo todos os 50 casos.

5.3.1 Experimento 2: análise dos resultados

Após o preenchimento das planilhas pelos profissionais de saúde, os dados foram tabulados contendo as recomendações do MAS4GC, as recomendações dos profissionais de saúde e as recomendações do especialista da UTI para tratamento e monitoramento, conforme apresentado na Tabela 5.1 e descritos na sequência:

- As recomendações de tratamento do MAS4GC foram analisadas de acordo com as recomendações dos profissionais de saúde, as quais coincidiram em 80% dos casos.
- Ainda acerca do tratamento, comparando as recomendações do MAS4GC com as do médico intensivista especialista houve coincidência de 84%.
- Em relação as recomendações de monitoramento, os profissionais de saúde coincidiram com as do MAS4GC em apenas 48% das vezes.

- Ainda acerca do monitoramento, as recomendações do MAS4GC coincidiram com as do especialista em 78% das vezes.

Tabela 5.1: Comparação de recomendações de tratamento e monitoramento.

Comparação	Tratamento	Monitoramento
Profissionais de Saúde x SMA	80%	48%
Especialista em UTI x SMA	84%	78%

Segundo o médico especialista em UTI, uma taxa de acerto aceitável seria próxima a 80% dos casos. Esse experimento mostra que as recomendações do MAS4GC foram satisfatórias na maioria dos casos, exceto quanto à recomendação de acompanhamento por profissionais de saúde. Diante dessa questão específica, com o auxílio do especialista em UTI, percebeu-se que a divergência das recomendações pode ter sido pelos seguintes fatos:

- alguns profissionais de saúde utilizam seus parâmetros de frequência para medir a glicemia, ignorando a tabela de recomendação fornecida no experimento (Tabela 2.2).
- os profissionais de saúde utilizaram parâmetros específicos das UTI que atuam, como coletas de hora em hora, por exemplo.
- os profissionais de saúde seguem padrões de monitoramento contínuo, onde a frequência das coletas é constante.

Embora os resultados apresentados sejam satisfatórios, outra análise acerca desses dados foi feita, buscando melhorar os resultados do experimento.

5.3.2 Validação de teste diagnóstico

Foi realizado uma validação de teste diagnóstico para mensurar a sensibilidade e especificidade dos diagnósticos do MAS4GC em comparação com o médico intensivista. Sendo o médico intensivista um especialista em UTI suas recomendações foram definidas como padrão ouro.

Conforme apresentado na validação de teste diagnóstico na Seção 2.3, a sensibilidade está relacionada a probabilidade de um desfecho esperado positivo e especificidade é a probabilidade de que um paciente que não necessita de tratamento tenha um desfecho esperado negativo. Aplicando esses conceitos a proposta um desfecho positivo (sensibilidade) seria o MAS4GC indicar um tratamento quando o paciente realmente tenha que receber um tratamento, como por exemplo a aplicação de insulina. E a especificidade seria o agente recomendar *Manter Observação*, ou seja, não efetuar nenhum tratamento, caso o paciente realmente não necessite de aplicação de glicose ou insulina.

A Figura 5.6 apresenta os resultados desse teste em relação ao tratamento. É possível ver que o MAS4GC agiu nos 8 casos onde o intensivista disse que deveria agir, alcançando 100% de sensibilidade para detectar casos onde devesse agir (indicando um tratamento). E dos 42 casos onde o intensivista indicou que não era necessário agir o MAS4GC não agiu em 36 casos, o que resulta em uma especificidade de 85,71%. Ou seja, o agente disse que deveria agir em situações onde o especialista disse que não era necessário agir. No

geral, a acurácia foi de 88%, considerando o total de acertos e o modelo como um todo obteve a pontuação de 0,92 no *F1-score*.

Apesar de ser uma especificidade relativamente boa, esses 6 casos de discordância foram analisados e foi possível notar que todos os casos são limítrofes, ou seja, os valores das glicemias estavam próximos do valor superior estabelecido. Inclusive o tratamento recomendado pelo MAS4GC nesses casos foi de aplicar a menor quantidade de insulina (2 unidades de insulina regular SC). De acordo com o intensivista essa é uma decisão aceitável, pois não oferece riscos a saúde do paciente.

Tratamento			Intensivista	
			Agir	Não Agir
		50	8	42
MAS4GC	Agir	14	8	6
	Não Agir	36	0	36
Sensibilidade			100,00%	
Especificidade			85,71%	
Acurácia			88,00%	
F1-score			0,92	

Figura 5.6: Resultado da validação de teste diagnóstico para tratamento.

Em suma, em termos de tomada de decisão o MAS4GC foi seguro, pois em todos os casos onde houveram discordância, o especialista considerou que a decisão tomada pelos agentes era aceitável.

A Figura 5.7 apresenta os resultados desse teste em relação ao monitoramento. Diferentemente do tratamento, onde verificamos se as indicações eram iguais ou diferentes, no monitoramento tem-se variáveis numéricas, que representam as horas em que devem acontecer as coletas, então a comparação se dá pela proximidade dos valores. Foi definido um ponto de corte de duas horas, ou seja, serão verificados os casos em que tanto o MAS4GC quanto o profissional disseram que havia necessidade de se coletar a glicemia em até duas horas (levando em consideração os casos mais críticos).

Considerando o ponto de corte de duas horas, foi possível verificar que o MAS4GC fez indicações de coletas próximas ao horário recomendado pelo especialista. O MAS4GC acertou todos os casos que não precisavam fazer coleta em até duas horas e também acertou em 4 dos 6 casos na qual o especialista disse que precisava. Ou seja, o MAS4GC errou em apenas dois casos, resultando em uma sensibilidade de 66,67% (apesar de serem apenas dois erros, foram 2 de 6) e 100% de especificidade.

Embora os resultados tenham sido satisfatórios alcançando uma acurácia de 96% e um *F-score* de 0,80, uma análise específica nos dois casos onde o MAS4GC “errou” foi elaborada. Foi constatado que em ambos os casos os valores eram limítrofes, com erro de apenas duas horas e dentro das regras que foram consideradas seguras perante o especialista. Segundo o especialista, esse erro pode ser minimizado por calibrações periódicas.

Monitoramento			Intensivista	
			Prox.	Não Prox.
		50	6	44
MAS4GC	Prox.	4	4	0
	Não Prox.	46	2	44
Sensibilidade			66,67%	
Especificidade			100,00%	
Acurácia			96,00%	
F1-score			0,80	

Figura 5.7: Resultados do validação de teste diagnóstico para Monitoramentos.

Ao final do Experimento 2, conforme os resultados apresentados e as métricas estabelecidas, conclui-se que o MAS4GC apresenta resultados promissores comparados ao especialista humano. Embora tal resultado mostre que os agentes recomendem tratamentos equivalentes a um médico especialista, pode se assumir que há um ganho em escala com isso, onde um médico especialista pode analisar um único paciente por vez, ao contrário de um SMA que pode fazer recomendações para vários pacientes no mesmo intervalo de tempo. Outro fator que vale ser ressaltado é que os agentes possuem o conhecimento semelhante ao intensivista e esse conhecimento pode ser consultado pelos demais profissionais da saúde que atuam na UTI como enfermeiros, fisioterapeutas entre outros. Permitindo que os profissionais possam combinar o seu conhecimento com o de um especialista em terapia intensiva e efetuar o melhor tratamento possível em seus pacientes.

Capítulo 6

Conclusões

O presente trabalho apresenta uma solução baseada em agentes inteligentes para o controle glicêmico em pacientes internados em UTI. Foi levado em consideração o problema de controle glicêmico de pacientes em UTI visando a automatização de procedimentos dos profissionais de saúde, viabilizando o compartilhamento do conhecimento de um especialista em cuidados intensivos.

A literatura evidencia que a combinação de uma abordagem com sistemas baseados em agentes inteligentes utilizando regras de inferência lógica e modelos de predição são promissores na área de saúde. Os agentes de um SMA podem cooperar e agir de maneira autônoma em ambientes complexos tal como uma UTI hospitalar. Os modelos preditivos podem contribuir com predições de glicemias permitindo o acompanhamento preventivo, enquanto o conhecimento baseado em regras formaliza o conhecimentos de médicos intensivistas e outros profissionais especializados. Não foram detectados trabalhos com foco no controle glicêmico de pacientes internados em UTI utilizando SMA como neste trabalho.

Com base na solução proposta e implementada nesta dissertação foram realizados dois experimentos como prova de conceito, utilizando pacientes com dados sintéticos. O Experimento 1 mostrou que o modelo de predição com regressão linear múltipla (RLM3) tem uma taxa de acerto de 90% da glicemia para as próximas 4 horas, permitindo que os profissionais de saúde antecipem e realizem o tratamento preventivo nos pacientes. No Experimento 2, as recomendações de tratamento feitas pelo MAS4GC coincidiram em 88% dos casos com as do especialista e 96% sobre as recomendações para monitoramento (coleta de glicemia) dos pacientes, mostrando-se uma solução viável. Conforme sugerido pelo especialista, a utilização do MAS4GC pode analisar informações de um maior número de pacientes que um profissional humano durante o mesmo período de tempo, podendo compartilhar o conhecimento de um médico intensivista com os demais profissionais que atuam em UTI.

6.1 Publicações

Alguns resultados de publicação científica foram alcançados durante a realização deste trabalho, conforme segue:

- Segato, T. H. F., Ralha, C. G., Fernandes, S. E. S. (2021). *Development process of multiagent system for glycemic control of intensive care unit patients. Artificial*

Intelligence Research. International Peer-reviewed and Open Access Journal for the Artificial Intelligence Specialists., Sciedu Press, 10(1): 43-56. DOI: <https://doi.org/10.5430/air.v10n1p43>

- Segato, T. H. F., Serafim, R. M. da S., Fernandes, S. E. S., Ralha, C. G. (2021). *MAS4GC: Multi-Agent System for Glycemic Control of Intensive Care Unit Patients*. 10th Brazilian Conference on Intelligent Systems, BRACIS 2021, Online Event, Brazil, November 29—December 03, 2021, Proceedings.

6.2 Trabalhos futuros

Apesar dos resultados promissores a investigação precisa ser aprofundada. Como trabalhos futuros, pretende-se expandir a base de regras para melhorar a capacidade dos agentes quanto ao processo de recomendação, principalmente em termos de monitoramento glicêmico de pacientes em diferentes contextos, analisando novos valores de taxas glicêmicas e respectivas escalas de tempos de monitoramento de paciente. Considerar novos ambientes, tais como de monitoramento contínuo ou com pacientes específicos com diabetes ou infectados com COVID-19.

Embora o modelo preditivo tenha apresentado bons resultados, a aplicação de outras técnicas de IA com aprendizado de máquina (e.g., SVM, NN, DL) para melhorar o modelo preditivo com dados de pacientes reais, o uso de lógica difusa (FL) para tratar intervalos de dados glicêmicos de forma adequada, entre outras técnicas, pode resultar em investigações interessantes.

Em complemento ao sistema de coleta de dados e tratamento manual, seja utilizando o sistema Glycon ou os profissionais de saúde, uma opção de investigação interessante seria estender o MAS4GC para incluir um módulo inicial de IdC com sensores e atuadores para os agentes autônomos (Seção 4.3). Desta forma, o monitoramento e tratamento poderão ser realizados por meio de monitores de glicemia e bombas de infusão. No entanto, novos requisitos poderão ser adicionados a proposta de solução apresentada neste trabalho para ampliar a potencialidade de uso da tecnologia de IdC. Estender a solução desenvolvida neste trabalho incluindo os demais itens do FAST HUG também será uma contribuição interessante.

Considerando os resultados alcançados até o momento, pode-se dizer que o MAS4GC apresenta uma boa capacidade de predição e recomendação com dados simulados de pacientes. No entanto, seria muito recomendável testes em ambientes clínicos envolvendo pacientes reais. Certamente, esse caminho requer a aprovação por um Comitê de Ética da UnB (e.g., Comitê de Ética em Pesquisa da Faculdade de Medicina - CEP/FM/UnB), o que durante o tempo de execução de um mestrado acadêmico, e principalmente em época de pandemia não se mostrou viável.

Referências

- Anjos, L. A. d., da, V. G. V., and Castro, I. R. R. d. (1998). Distribuição dos valores do índice de massa corporal da população brasileira até 25 anos. *Rev Panam Salud Publica/Pan Am J Public Health*, 3(3):164–173. 47, 55
- Anvisa, A. N. d. V. S. (2010). Resolução de Diretoria Colegiada – RDC N^o 7, de 24 de fevereiro de 2010. Available at http://portal.anvisa.gov.br/documents/10181/3231546/RDC_07_2010_COMP.pdf access on July 2020. 4, 5
- Barbetta, P. A., Bornia, A. C., and Reis, M. M. (2010). *Estatística Para Cursos De Engenharia E Informática*. Atlas. 3a edição. 17, 22
- Barretto, E. H. S. and Ito, M. (2016). Restrições de integridade em prontuários eletrônicos nosql - uma revisão sistemática. *XV Congresso Brasileiro de Informática em Saúde - CBIS*, pages 395–404. 42
- Bastos, S. F., Sabara, F. G., Rosa, R., and Felipe, L. (2009). Sizebr – o estudo antropométrico brasileiro. Available at http://arquivos.portaldaindustria.com.br/app/conteudo_18/2014/07/10/6822/SizeBR_Por.pdf?r=0.562248437169 access on April 2021. 47, 55
- Bellifemine, F. L., Caire, G., and Greenwood, D. (2007). *Developing Multi-Agent Systems with JADE (Wiley Series in Agent Technology)*. John Wiley & Sons, Inc., Hoboken, NJ, USA. 19
- Berghe, V. D., G., Wouters, P., Weekers, F., Verwaest, C., Bruyninckx, F., Schetz, M., Vlasselaers, D., Ferdinande, P., Lauwers, P., and Bouillon, R. (2001). Intensive insulin therapy in critically ill patients. *New England journal of medicine*, 345(19):1359–1367. 7
- Bertachi, A., Viñals, C., Biagi, L., Contreras, I., Vehí, J., Conget, I., and Giménez, M. (2020). Prediction of nocturnal hypoglycemia in adults with type 1 diabetes under multiple daily injections using continuous glucose monitoring and physical activity monitor. *Sensors (Basel)*, 20(6):1705. PMID: 32204318. 2, 29, 31, 32, 33
- Braga, A. A., Fernandes, M. C. C., Madeira, M. P., and Júnior, A. A. P. (2015). Associação entre Hiperglicemia e Morbidade em Pacientes Críticos na Unidade de Terapia Intensiva de um Hospital Terciário de Fortaleza - CE. *Journal of Health and Biological Sciences*, 3(3):132–136. 2, 8

- Bresciani, P. and Perini, A. (2004). Tropos : An Agent-Oriented Software Development Methodology. *Autonomous Agents and Multi-Agent Systems*, 8:203–236. 12, 15
- Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J. and Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122. 22, 23
- Cervi, A., do C. C., F. S., and Priore, S. E. (1998). Análise crítica do uso do índice de massa corporal para idosos. *Revista de Nutrição*, 18(6):765–775. 47, 55
- Chase, J. G., Benyo, B., and Desai, T. (2019). Glycemic control in the intensive care unit: a control systems perspective. *Annual Reviews in Control*, 48:359–368. 2, 5, 7
- Contreras, I. and Vehi, J. (2018). Artificial intelligence for diabetes management and decision support: Literature review. *Journal of Medical Internet Research*, 20(5):1–24. x, 1, 18, 25, 26, 30, 31, 32
- Cooksley, T., McAvoy, T., and Haji-Michael, P. (2018). Glucose control in critical Care. *Journal of the Intensive Care Society*, 13(4):289–292. 1, 2, 8, 9, 10
- Dalpiaz, F., Franch, X., and Horkoff, J. (2016). istar 2.0 language guide. Available at <https://arxiv.org/pdf/1605.07767.pdf> access on June 2020. 13, 14, 15
- Dankwa-Mullan, I., Rivo, M., Sepulveda, M., Park, Y., Snowdon, J., and Rhee, K. (2019). Transforming Diabetes Care Through Artificial Intelligence: The Future Is Here. *Population Health Management*, 22(3):229–242. 1, 25, 26
- Darabi, Z., Zarandi, M. H., Solgi, S. S., and Turksen, I. B. (2015). An intelligent multi-agent system architecture for enhancing self-management of type 2 diabetic patients. In *IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. IEEE Inc. 2, 8, 27, 30, 33
- DeJournett, J. and DeJournett, L. (2017). Comparative Simulation Study of Glucose Control Methods Designed for Use in the Intensive Care Unit Setting via a Novel Controller Scoring Metric. *Journal of Diabetes Science and Technology*, 11(6):1207–1217. 1, 2, 10, 28, 30, 33
- DeJournett, J., Nekludov, M., DeJournett, L., and Wallin, M. (2020). Performance of a closed-loop glucose control system, comprising a continuous glucose monitoring system and an AI-based controller in swine during severe hypo- and hyperglycemic provocations. *Journal of Clinical Monitoring and Computing*, pages 1–9. 2, 29, 30, 33
- DeJournett, L. and DeJournett, J. (2016). In Silico Testing of an Artificial-Intelligence-Based Artificial Pancreas Designed for Use in the Intensive Care Unit Setting. *Journal of Diabetes Science and Technology*, 10(6):1360–1371. 2, 8, 18, 27, 29, 30, 33
- GREI-UFC (2019). Python Agent DEvelopment framework. Available at <https://pade.readthedocs.io/en/latest/>. Access on January 2021. 20

- IBGE (2020). *Pesquisa nacional de saúde : 2019 : atenção primária à saúde e informações antropométricas : Brasil / IBGE*. Coordenação de Trabalho e Rendimento. 47, 55
- Jemal, H., Kechaou, Z., and Ben Ayed, M. (2017). Enhanced decision support systems in intensive care unit based on intuitionistic fuzzy sets. *Advances in Fuzzy Systems*, 2017:1–8. 28, 29, 31, 33
- Jemal, H., Kechaou, Z., and Ben Ayed, M. (2019). Multi-agent based intuitionistic fuzzy logic healthcare decision support system. *Journal of Intelligent and Fuzzy Systems*, 37(2):2697–2712. 2, 29, 31, 33
- Kim, D.-Y., Choi, D.-S., Kim, J., Chun, S. W., Gil, H.-W., Cho, N.-J., Kang, A. R., and Woo, J. (2020). Developing an individual glucose prediction model using recurrent neural network. *Sensors (Basel)*, 20(11):6460. PMID: 33198170. 2, 30, 31, 32, 33
- Knottnerus, J. A. and Buntinx, F. (2011). *The evidence base of clinical diagnosis : theory and methods of diagnostic research*. BMJ Books. 2a edição. x, 23, 24
- Krinsley, J. S. (2004). Effect of an intensive glucose management protocol on the mortality of critically ill adult patients. *Mayo Clinic Proceedings*, 79(8):992–1000. 7
- Kumar, A., Dey, D., Sabharwal, M., Sarkar, D., and Kumar, A. (2020). One fast hugs bid covid: Covid care bundle modified mnemonic to facilitate covid 19 critical care medicine. *Journal of Medical Science and Critical Research*, 8(11):66–70. 6
- Lima, N. P., Horta, B. L., Motta, J. V. d. S., Valença, M. S., Oliveira, V., Santos, T. V. d., Gigante, D. P., and Barros, F. C. (2015). Evolução do excesso de peso e obesidade até a idade adulta, pelotas, rio grande do sul, brasil, 1982-2012. *Cadernos de saúde pública*, 31(9):2017–2025. 47, 55
- Liu, S.-p., c Qin, Wang, W., Zhang, M., Liu, C., Xiao, X., Liu, Z., Hu, W.-m., and Jin, P. (2020). Hyperglycemia is a strong predictor of poor prognosis in covid-19. *Diabetes Research and Clinical Practice*, 167(108338):1–6. 2
- Malak, J. S., Safdari, R., Zeraati, H., Nayeri, F. S., Mohammadzadeh, N., and Seied Farajollah, S. S. (2018). An agent based architecture for high-risk neonate management at neonatal intensive care unit. *Electronic Physician*, 10(1):6193–6200. 28, 31, 33
- Malak, J. S., Zeraati, H., Nayeri, F. S., Safdari, R., and Shahraki, A. D. (2019). Neonatal intensive care decision support systems using artificial intelligence techniques: a systematic review. *Artificial Intelligence Review*, 52(4):2685–2704. 2, 25, 26
- Melmed, S., Auchus, R. J., Goldfine, A. B., Koenig, R. J., and Rosen, C. J. (2020). *Williams Textbook of Endocrinology*. Elsevier. 14a edição. 47, 55, 56
- Melo, L. S., Sampaio, R. F., Leão, R. P. S., Barroso, G. C., and Bezerra, J. R. (2019). Python-based multi-agent platform for application on power grids. *International Transactions on Electrical Energy Systems*, 29(6):e12012. e12012 ITEES-18-0867.R2. 19

- Mendes, A. H. D. (2017). Agentes autônomos para monitoramento e alocação dinâmica de recursos para nuvem computacional. Mestrado em informática, Universidade de Brasília - UnB, Brasília. 16, 22
- MongoDB (2021). O que é o mongodb? Available at <https://www.mongodb.com/pt-br/what-is-mongodb> access on Ago 2021. 43
- Moraes, P. L. (2020). Glicose: Brasil escola. Available at <https://brasilecola.uol.com.br/saude/glicose.htm> access on January 2021. 7
- Noronha, K. V. M. d. S., Guedes, G. R., Turra, C. M., Andrade, M. V., Botega, L., Nogueira, D., Calazans, J. A., Carvalho, L., and Servo, L. (2020). Pandemia por covid-19 no brasil: análise da demanda e da oferta de leitos hospitalares e equipamentos de ventilação assistida segundo diferentes cenários. Available at <https://doi.org/10.1590/0102-311X00115320> access on April 2021. 2
- Pimentel, J. and Castro, J. (2018). piStar Tool-A Pluggable Online Tool for Goal Modeling. In *Proceeding of the IEEE 26th International Requirements Engineering Conference*. 13, 16
- Pérez, R. A. M. (2019). Experta. Available at <https://experta.readthedocs.io/en/latest/index.html> access on January 2021. 19, 21
- Rath, S., Tripathy, A., and Tripathy, A. R. (2020). Prediction of new active cases of coronavirus disease (covid-19) pandemic using multiple linear regression model. *Diabetes & Metabolic Syndrome: Clinical Research & Reviews*, 14(1):1467–1474. 17, 58
- Rodrigues, V. R., Serafim, R. M. d. S., and Segato, T. H. F. (2021). Gerenciamento do Índice glicêmico de pacientes internados em uti por meio de sistemas informatizados. Trabalho de conclusão de curso (graduação) – tecnólogo em sistemas para internet., Instituto Federal de Brasília - IFB, Brasília. 45
- Russell, S. and Norvig, P. (2010). *Artificial Intelligence A Modern Approach*. Pearson Education, Inc., 3 edition. 12, 17, 18
- SDB, S. B. d. D. (2015). Métodos para avaliação do controle glicêmico. Available at <https://www.diabetes.org.br/profissionais/images/pdf/diabetes-tipo-2/010-Diretrizes-SBD-Metodos-para-Avaliacao-pg110.pdf> access on April 2021. 9
- Silva, S. C. d. and Monteiro, W. D. (2009). Levantamento do perfil antropométrico da população brasileira usuária do transporte aéreo nacional – projeto conhecer. Available at https://www2.anac.gov.br/arquivos/pdf/Relatorio_Final_Projeto_Conhecer.pdf access on April 2021. 47, 55
- The Drools Team (2020). Drools documentation - version 7.40.0.final. Available at https://docs.jboss.org/drools/release/7.40.0.Final/drools-docs/html_single/index.html access on july 2020. 19, 20
- Vehí, J., Contreras, I., Oviedo, S., Biagi, L., and Bertachi, A. (2020). Prediction and prevention of hypoglycaemic events in type-1 diabetic patients using machine learning. *Health Informatics Journal*, 26(1):703–718. PMID: 31195880. 2, 29, 31, 32, 33

- Vincent, J. L. (2005). Give your patient a fast hug (at least) once a day. *Critical Care Medicine*, 33(6):1225–1229. 1, 5, 7
- Wooldridge, M. (2009). *Introduction to Multiagent Systems*. John Wiley & Son, 2 edition. 12
- Zhang, Z. (2015). A mathematical model for predicting glucose levels in critically-ill patients: the pignoli model. *PeerJ*, pages 1–11. 17

Apêndice A

Tabela de atributos das classes do MAS4GC

As Tabelas A.1, A.2, A.3 e A.4 apresentam os atributos contidos nas classes *Paciente*, *Glicemia*, *Aplicacao* e *Agentes* referentes ao diagrama de classes do MAS4GC apresentado na Figura 4.7.

Tabela A.1: Atributos da classe *Paciente*.

Atributo	Significado
prontuario	número atribuído a cada paciente para um melhor gerenciamento de seus dados pelos profissionais de saúde durante a sua internação
nome	nome do paciente
dataNascimento	data de nascimento do paciente
sexo	sexo do paciente
dataHoraInternacao	data e hora da internação do paciente na UTI
tipoInternacao	tipo da internação que pode ser: clínica, cirúrgica de urgência, cirúrgica eletiva, síndrome coronária aguda, acidente vascular encefálico, trauma ou oncológica
peso	peso do paciente em Kg
altura	altura do paciente em cm
imc	resultado do cálculo do índice de massa corporal
diabetes	pode receber os valores: ignorado, controle domiciliar dietético, controle domiciliar com hipoglicemiantes oral, controle domiciliar com insulina, controle domiciliar medicamentoso misto ou não tem
insuficienciaRenal	pode receber os valores: ignorado, crônica dialítica, crônica não dialítica, aguda dialítica, aguda não dialítica ou não tem
corticoide	pode receber os valores: ignorado, menos de 7 dias, a mais de 7 dias ou não tem (refere-se ao uso em dias)

infeccao	pode receber os valores: ignorado, infecção simples, sepse ou não tem
sindromeDesc-Respiratorio	pode receber os valores: ignorado, possui ou não tem
instabilidade-Hemodinamica	pode receber os valores: ignorado, sim - controlado sem drogas vasoativas, sim - controlado com drogas, sim - descontrolado apesar das drogas ou não tem
statusPaciente	pode receber os valores: internado ou alta
planoAplicacao	recebe um vetor de valores referente aos horários em que deverá ser coletado o sangue do paciente para a medição da glicemia
observacoes	campo aberto para entrada de observações em relação ao paciente
tratamento	campo onde são armazenadas as recomendações de tratamento (aplicação de glicose/insulina) advindas do agente PTA
monitoramento	campo onde são armazenadas as recomendações de monitoramento (data e hora na qual deve acontecer a próxima coleta de glicemia) advindas do agente AMA
createDate	data e hora de criação do registro no sistema
updateDate	data e hora de alteração dos dados do registro no sistema

Tabela A.2: Atributos da classe *Glicemia*.

Atributo	Significado
dataHoraColeta	data e hora em que a coleta foi efetuada
tipoColeta	pode receber os valores: capilar e bioquímica
tipoAlimentação	pode receber os valores: zero, oral líquida, oral pastosa, oral completa, interal intermitente, interal contínua ou parenteral
valorGlicemia	valor da glicemia em mg/dL
observacoes	campo aberto para entrada de observações em relação a glicemia do paciente
ultimaAlimentacao	há quantas horas o paciente recebeu a última alimentação
createDate	data e hora de criação deste registro no sistema
updateDate	data e hora de alteração dos dados deste registro no sistema

Tabela A.3: Atributos da classe *Aplicacao*.

Atributo	Significado
dataHoraAplicacao	data e hora em que a aplicação foi efetuada
tipoAplicacao	pode receber os valores: de resgate ou de horário
viaAdministracao	pode receber os valores: intravenoso, subcutâneo ou via oral
droga	pode receber os valores: insulina de ultra rápida, insulina regular, insulina NPH, insulina lenta – Demerol, insulina ultra lenta – Glargina, Glicose a 25% ou Glicose a 50%
posologia	indica a quantidade da droga utilizada na aplicação (seguido de "UI" se for insulina e "ml" de for glicose)
observacoes	campo aberto para entrada de observações em relação a aplicação da glicose ou insulina no paciente
createDate	data e hora de criação deste registro no sistema
updateDate	data e hora de alteração dos dados deste registro no sistema

Tabela A.4: Atributos da classe *Agentes*.

Atributo	Significado
paciente	representa os dados da classe Paciente que compõem esta classe
glicemia	representa os dados da classe Glicemia que compõem esta classe
aplicacao	representa os dados da classe Aplicacao que compõem esta classe
dadosPaciente	array criado apenas com os dados do paciente e glicemia que serão utilizados para determinar a situação do paciente
situacao	valor retornado pela função que faz a verificação do estado glicêmico do paciente, e.g., gAlvo (glicemia está na faixa alvo)
glicemia	valor da glicemia atual do paciente (caso tenha apenas uma coleta) ou da glicemia prevista (caso tenha várias glicemias coletadas)
situacaoPaciente	uma string (JSON) que contém as informações necessárias da situação do paciente (id, nome, glicemia, situacao, dataHoraColeta) a ser enviadas aos demais agentes
createDate	data e hora de criação deste registro no sistema
updateDate	data e hora de alteração dos dados deste registro no sistema

Apêndice B

Lista de Fórmulas usadas no Simulador

A Figura B.1 apresenta as fórmulas utilizadas nos cálculos para a geração de cada um dos dados do paciente simulado.

	A	B	C	D
1	Nome	Marciele La Rosa	SE(B2="F";PROCV(ALEATÓRIOENTRE(1;1485);Nomes!A:D;3;FALSO); - se repete para sobrenome	
2	Sexo	F	SE(ALEATÓRIOENTRE(0;1)=0;"F";"M")	
3	Idade	76	SE(ALEATÓRIOENTRE(1;40)<=4;INV.NORM.N(ALEATÓRIO());35;6);INV.NORM.N(ALEATÓRIO());70;12))	
4	Altura	172,8	SE(B2="F";INV.NORM.N(ALEATÓRIO());162;6,8);INV.NORM.N(ALEATÓRIO());172;7,2))	
5	IMC	28,32	SE(B3<=60; 15 + (INV.BETA(ALEATÓRIO()); 3;7)*((35+(B3-15)*(1/3)) - 15));	
6	Peso	84,58	B5*((B4/100)^2)	
7	Diabetes	Não	SE(ALEATÓRIO()<= (B3-15)*(0,48/95); "Sim"; "Não")	
8	Glicemia Basal(med)	89,0	SE(B7="Não"; ALEATÓRIOENTRE(80;90); ALEATÓRIOENTRE(90;130))	
9	Glicemia Basal(DP)	8,0	SE(B7="Não"; ALEATÓRIOENTRE(B8*0,088; B8*0,089); ALEATÓRIOENTRE(B8*0,085; B8*0,11))	
10	Freq Glicemias(h)	6,0	SE(B7="Não";SE(C10=4;6;C10*2);8)	
11	Data e hora inicial	01/01/2021 12:00	Dara e hora inicial definidas a mão	
12	Carga Alimentar(Pico)	20,75	ALEATÓRIOENTRE(1800;2200)/100	
13	Carga Alimentar(DP)	4,62	ALEATÓRIOENTRE(400;600)/100	
14	Carga DM	1	SE(B7="Sim"; 1+ INV.BETA(ALEATÓRIO());2;7)*2,5;1)	

Figura B.1: Detalhamento das fórmulas usadas no simulador.