GABRIEL DA SILVA GONÇALVES
MATHEUS YUNJIE QIU

Aplicação de técnicas de Process Mining para análise do protocolo de tratamento de Sepsis: um estudo de caso na base do MIMIC-IV

PROJETO DE GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO
APRESENTADO AO DEPARTAMENTO DE ENGENHARIA INDUSTRIAL
DA PUC-RIO, COMO PARTE DOS REQUISITOS PARA OBTENÇÃO
DO TÍTULO DE ENGENHEIRO DE PRODUÇÃO

Orientadora: Profa. Fernanda Araujo Baião Amorim
Coorientador: Prof. Igor Tona Peres

Departamento de Engenharia Industrial
Rio de Janeiro, 12 de junho de 2023.

AGRADECIMENTOS

Gostaríamos de expressar nossa mais profunda gratidão aos nossos orientadores, Fernanda Baião e Igor Peres, por sua orientação incansável e *insights* valiosos que foram fundamentais para a realização deste trabalho. Sua paciência, suporte e críticas construtivas contribuíram de maneira decisiva para o desenvolvimento de nossa pesquisa.

Nosso agradecimento estende-se ao Dr. Jorge Ibrain Figueira Salluh, especialista na área da saúde, cuja expertise e disponibilidade foram fundamentais para a compreensão e aprofundamento de aspectos técnicos abordados neste trabalho. Seu auxílio contribuiu de maneira significativa para a qualidade e credibilidade do nosso estudo. Também agradecemos por compartilhar seu conhecimento conosco e por sua disposição em ajudar mesmo diante de uma rotina intensa. Sua contribuição foi inestimável e estamos extremamente agradecidos por sua colaboração.

Estendemos nossos agradecimentos a todos os professores do departamento de Engenharia Industrial, cujas aulas e conselhos nos inspiraram e nos guiaram por este percurso. Sua sabedoria e estímulo foram de grande valor.

Agradecemos também aos nossos familiares, tanto aqueles que estão aqui conosco quanto aqueles que, embora não mais presentes fisicamente, deixaram um legado de amor, e amigos que, mesmo indiretamente, participaram deste processo conosco. Seu apoio emocional e incentivo foram essenciais para enfrentarmos os desafios desta jornada.

Por último, mas não menos importante, somos gratos à PUC-Rio, pela estrutura fornecida e pelo ambiente acadêmico enriquecedor que nos permitiu chegar até aqui.

Este trabalho é um reflexo da ajuda e do apoio que recebemos de todos vocês. Estamos muito gratos.

RESUMO

O presente estudo foi realizado com o intuito de analisar a conformidade do tratamento de sepse na base MIMIC-IV, um banco de dados relacional disponível ao público que inclui dados sobre tratamento de pacientes em um centro médico acadêmico terciário em Boston, EUA. Nesse sentido, as principais perguntas de pesquisas feitas são, por exemplo: “Qual a conformidade geral em relação ao protocolo padrão de tratamento de sepse?”, “Quais tratamentos/atividades reduzem ou aumentam a mortalidade?” e “Quais são as principais recomendações a serem levadas em consideração?”. Para responder essas questões, ferramentas de mineração de processos e verificação de conformidade foram utilizadas. Além disso, um processo de geração de log de eventos foi realizado a fim de transformar a base do MIMIC-IV e suas tabelas relevantes em um material adequado para a análise. Os principais resultados estão relacionados a uma maior mortalidade em grupos não conformes com relação ao protocolo padrão de tratamento da sepse e outros pontos presentes, principalmente, nas análises de aderência por atividade e conformidade. Além disso, os resultados foram comparados com outro estudo, as limitações foram pontuadas, assim como possíveis trabalhos futuros.

Palavras-chave: MIMIC-IV, mineração de processos, análise de conformidade, sepse, log, Python.

ABSTRACT

The present study was undertaken to analyze sepsis treatment compliance in the MIMIC-IV database, a publicly available relational database that includes patient care data at an academic tertiary medical center in Boston, USA. In this sense, the main research questions asked are, for example: “What is the general compliance with the standard sepsis treatment protocol?”, “What treatments/activities reduce or increase mortality?” and “What are the main recommendations to take into account?”. To answer these questions, process mining and conformance checking tools were used. In addition, an event log generation process was carried out in order to transform the MIMIC-IV base and its relevant tables into suitable material for analysis. The main results are related to higher mortality in non-compliant groups with regard to the standard sepsis treatment protocol and other points present, mainly, in the analysis of adherence by activity and compliance. In addition, the results were compared with another study, the limitations were pointed out, as well as possible future work.

Keywords: MIMIC-IV, process mining, conformance checking, sepsis, log, Python.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

1 Introdução

A sepse é definida como uma disfunção orgânica com risco de vida causada por uma resposta desregulada do hospedeiro à infecção (SINGER et al., 2016). A sepse, decorrente de anomalias fisiológicas, patológicas e bioquímicas causadas por infecções, constitui um sério desafio à saúde pública. Nos Estados Unidos, a sepse é a doença que gera maiores gastos hospitalares, abrangendo mais de $38 bilhões (8,8%) dos gastos hospitalares totais nos Estados Unidos em 2017 (LIANG et al., 2020).

A ocorrência de sepse tem aumentado, provavelmente devido ao envelhecimento populacional com mais doenças concomitantes e maior identificação. Ainda que a incidência real seja incerta, projeções cautelosas indicam que a sepse é uma das principais causas de óbitos e enfermidades críticas no mundo. Ademais, percebe-se crescentemente que indivíduos que se recuperam da síndrome séptica, muitas vezes, enfrentam debilidades físicas, emocionais e cognitivas duradouras, com impactos significativos nos cuidados de saúde e na esfera social (SINGER et al., 2016).

No Brasil, o número total de pacientes com sepse e choque séptico incluídos no banco de dados do ILAS (Instituto Latino Americano de Sepse) de 2005 a 2022 é de 134.532 (ILAS, 2022). Em comparação, o total apenas para 2021 é de 14.366. Isso sugere um volume significativo de casos ao longo do tempo e indica que a sepse e o choque séptico continuam a ser um problema de saúde significativo no Brasil. O fato de haver 80 centros brasileiros contribuindo com dados para o relatório em 2022 é um sinal positivo de que há uma rede estabelecida de locais monitorando e relatando esses casos. Isso é crucial para entender a extensão do problema e para desenvolver e implementar estratégias de tratamento e prevenção eficazes.

Nos últimos anos, houve um esforço conjunto na adoção de sistemas digitais de registros de saúde nos hospitais. Nos EUA, em 2021, a adoção de Registros Eletrônicos de Saúde (EHR) certificados aumentou significativamente. Aproximadamente 78% dos médicos que trabalham em consultórios e 96% dos hospitais de cuidados intensivos não federais adotaram essa tecnologia (ONC, 2022). Isso representa um salto significativo em relação à última década, considerando que em 2011 apenas 28% dos hospitais e 34% dos médicos haviam integrado um EHR em sua prática.

Dados médicos coletados retrospectivamente têm sido cada vez mais utilizados em epidemiologia e modelagem preditiva, em parte devido à eficácia das abordagens de modelagem em grandes conjuntos de dados. Contudo, o acesso aos dados médicos para aprimorar o atendimento ao paciente ainda é um desafio, com a privacidade do paciente sendo uma das principais preocupações. Embora estudos com pacientes mostrem um consenso quase unânime de que dados médicos anonimizados devem ser usados para melhorar a prática médica, especialistas ainda discutem os melhores mecanismos para isso (MIMIC, 2023).

A base de dados MIMIC-III, antecessor do MIMIC-IV, adotou um esquema de acesso permissivo, permitindo ampla reutilização dos dados. Esse mecanismo teve sucesso no uso amplo do MIMIC-III em diversos estudos. O MIMIC-IV busca dar continuidade a esse sucesso, implementando mudanças para aprimorar a usabilidade dos dados e permitir mais aplicações em pesquisas (MIMIC, 2023).

O objetivo geral deste trabalho é ilustrar a aplicação de técnicas de mineração de processos, e em particular de verificação de conformidades em processos, para extrair informações úteis para o tratamento da sepse sob o ponto de vista de internações em um hospital nos Estados Unidos. Para isso, a base de dados MIMIC-IV foi extraída e tratada para gerar um log de eventos e, em seguida, servir para uma análise de conformidade a fim de estudar a aderência a um protocolo padrão para o tratamento da sepse. Ademais, o desenvolvimento do trabalho contou com o apoio de um especialista da área médica, que é pesquisador no tema específico do trabalho, doutor Jorge Ibrain Figueira Salluh.

O estudo está estruturado em cinco seções. O capítulo 2 possui a fundamentação teórica, apresentando os principais conceitos de sepse, indicadores médicos, process mining e gestão de processos de negócio. O capítulo 3 apresenta os materiais e métodos do trabalho. O capítulo 4 apresenta os resultados da aplicação da metodologia proposta e o capítulo 5 apresenta as considerações finais do trabalho, conclusões, contribuições, limitações e trabalhos futuros.

2 Referencial Teórico

Neste capítulo, será apresentado o referencial teórico necessário para embasamento da pesquisa aplicada na base do MIMIC-IV. Serão detalhados os conceitos relacionados ao protocolo de atendimento de sepse e a fundamentação teórica da análise dos dados por meio de mineração de processos (*Process Mining*).

2.1 Sepse e Choque Séptico

Sepse é uma resposta inadequada do organismo contra uma infecção que pode estar localizada em qualquer órgão e ser provocada por bactérias, fungos, protozoários ou vírus, como é o caso das formas graves de Covid-19 (BIBLIOTECA VIRTUAL DE SAÚDE, 2022). A definição de sepse tem evoluído ao longo dos anos. Em 2016, foi proposta uma nova definição conhecida como Sepsis-3, estabelecida por um grupo de especialistas em sepse. De acordo com a definição Sepsis-3, a sepse é definida como uma disfunção orgânica potencialmente fatal causada por uma resposta desregulada do hospedeiro à infecção (SINGER et al., 2016).

O tipo e gravidade das disfunções orgânicas é avaliada pelo escore SOFA (*Sequential Organ Failure Assessment*) que mede a função dos órgãos, incluindo respiração, circulação, função renal, função hepática, coagulação e sistema nervoso central. O choque séptico pode ser caracterizado como uma categoria específica de sepse, na qual estão presentes anormalidades excessivamente graves na circulação, função celular e metabolismo. Essas anormalidades se correlacionam com uma maior probabilidade de morte, em comparação com os casos de sepse isoladamente (SINGER et al., 2016). O choque séptico, a mais grave forma de sepse, é responsável por uma mortalidade hospitalar de 52,2% a 65,3% em pacientes que apresentam essa disfunção (FIOCRUZ, 2021).

2.2 SOFA e qSOFA

Segundo Singer et al. (2016), o SOFA é um sistema de pontuação utilizado na medicina intensiva para avaliar e quantificar a gravidade da disfunção orgânica em pacientes. É

comumente utilizado em unidades de terapia intensiva (UTIs) para avaliar o progresso e prognóstico de pacientes com diferentes tipos de falência de órgãos.

O escore SOFA avalia a função de seis sistemas orgânicos: respiratório, cardiovascular, hepático (fígado), coagulação, renal (rim) e neurológico. Cada sistema recebe uma pontuação com base em critérios específicos, como pressão arterial, oxigenação, contagem de plaquetas, níveis de bilirrubina e estado neurológico. As pontuações individuais são então somadas para calcular o escore geral do SOFA, que varia de 0 a 24.

Um escore SOFA mais alto indica um maior grau de disfunção orgânica e está associado à maior morbidade e mortalidade. O escore SOFA é útil para monitorar pacientes gravemente enfermos, avaliar a eficácia dos tratamentos e prever resultados. Ele fornece aos médicos uma ferramenta padronizada para avaliar e comunicar a gravidade da falência de órgãos em indivíduos criticamente enfermos.

O qSOFA (*Quick Sequential Organ Failure Assessment*) é um sistema de pontuação utilizado para avaliar a gravidade de pacientes com suspeita de infecção e avaliar o risco de desenvolvimento de síndrome de resposta inflamatória sistêmica (SIRS) ou sepse. É uma ferramenta rápida e simples que usa apenas três critérios clínicos (frequência respiratória, alteração do estado mental e pressão arterial sistólica) para identificar pacientes em maior risco de complicações graves. É projetado para ser aplicado rapidamente e pode ser usado como uma primeira triagem em situações de emergência.

O SOFA é frequentemente usado para monitorar a evolução da disfunção orgânica ao longo do tempo e avaliar a resposta ao tratamento em pacientes com sepse. Ele fornece uma pontuação contínua que reflete a gravidade da doença e ajuda a tomar decisões clínicas. O qSOFA é utilizado principalmente como uma ferramenta de triagem inicial para identificar pacientes com maior risco de complicações graves relacionadas à sepse e alertar os profissionais de saúde sobre a necessidade de uma avaliação mais aprofundada e intervenções imediatas.

2.3 Process Mining

A mineração de processos visa melhorar processos operacionais por meio do uso sistemático de dados de eventos. A partir de eventos e de modelos de processos, as técnicas de mineração de processos oferecem *insights*, identificam gargalos e desvios, preveem e diagnosticam problemas de desempenho e conformidade e auxiliam na automação ou remoção de tarefas repetitivas. (VAN DER AALST, 2022).

Essas técnicas podem ser voltadas ao passado (por exemplo, identificar causas de gargalos) ou ao futuro (prever tempo restante de processamento ou fornecer recomendações para reduzir taxas de falha). Ambas as análises podem desencadear ações, como contramedidas para problemas de desempenho. O foco da mineração de processos está em processos operacionais presentes em todos os setores e indústrias, incluindo produção, logística, finanças, vendas, educação, saúde e governo.

O conjunto de dados resultante é frequentemente chamado de *log de eventos* (registro de eventos), ou seja, uma coleção de eventos correspondentes ao processo selecionado. Técnicas de descoberta de processo são usadas para criar modelos de processo automaticamente. A geração de log será aprofundada no tópico “Geração de log” deste trabalho.

No entanto, existem inúmeras abordagens para aprender modelos de nível superior representados usando a notação de Modelagem de Processos de Negócio (BPMN), redes de Petri ou diagramas de atividades da Linguagem de Modelagem Unificada (UML) (VAN DER AALST, 2022).

Além desses modelos há representações gráficas de um processo ou fluxo de trabalho que mostra as interconexões e a complexidade do mesmo de forma desordenada e confusa. O nome é diagrama espaguete derivado da semelhança visual entre as linhas do diagrama e os fios de um prato de espaguete.

Exemplo de diagrama espaguete (Figura 1):

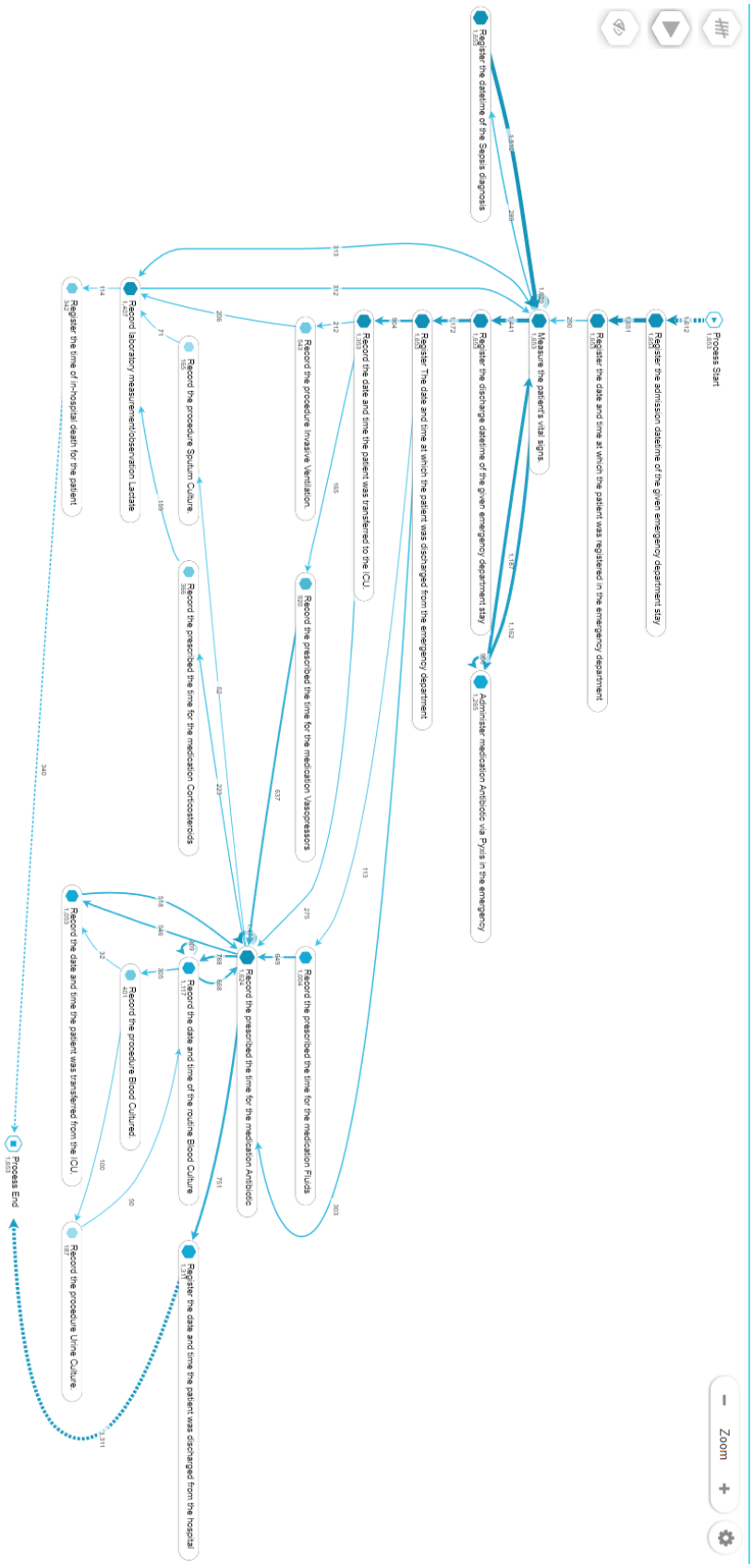

Figura 1 - Diagrama “Espaguete”.
Fonte: Elaboração própria.

2.4 BPMN

O BPMN (*Business Process Model and Notation*) é uma notação gráfica amplamente utilizada para a modelagem e documentação de processos de negócios. Desenvolvida pelo *Object Management Group* (OMG). O BPMN oferece uma linguagem padronizada para representar visualmente os processos, tornando-os mais compreensíveis e facilitando a comunicação entre diferentes partes interessadas.

O BPMN utiliza uma variedade de símbolos e elementos gráficos para representar os componentes do processo. Esses elementos incluem atividades, eventos, fluxos de sequência, gateways e objetos de dados. As atividades representam as tarefas ou ações executadas no processo, enquanto os eventos indicam os pontos de início, fim ou ocorrências intermediárias que podem acionar ações. Os fluxos de sequência conectam as atividades e eventos, mostrando a ordem em que as atividades devem ser realizadas. Os gateways são utilizados para representar decisões ou pontos de ramificação no processo, permitindo que o fluxo siga diferentes caminhos com base em condições específicas. Os objetos de dados representam as informações manipuladas ou produzidas pelo processo (OBJECT MANAGEMENT GROUP, 2010).

A notação BPMN é altamente visual e intuitiva, tornando-a acessível para usuários não técnicos e facilitando a colaboração entre diferentes equipes e departamentos. Além disso, o BPMN permite que os processos sejam documentados em diferentes níveis de detalhes, desde uma visão geral do processo até detalhes mais específicos.

A fim de tornar a explicação do BPMN mais clara e ilustrativa, um exemplo simples será utilizado baseado em van der Aalst (2022), a Figura 2, mas com elementos que aparecem no protocolo padrão na parte de materiais e métodos. Isso servirá para explicar conceitos de processo, como escolha, pular e repetir. Considere um processo envolvendo as seguintes atividades: entrar na emergência (ee), medir sinais vitais (msv), administrar antibióticos (aa), sair da emergência (se), entrar na UTI (eu), administrar vasopressores (av), registrar alta (ra), e registrar morte (rm). Esse processo fictício será chamado de “processo tratar paciente” e será usado para ilustrar os principais conceitos e notações.

O processo começa com a atividade de entrar na emergência (ee). Em seguida, duas atividades são executadas em qualquer ordem: medir sinais vitais (msv) e administrar antibióticos (aa). Os dois símbolos em forma de diamante com + dentro denotam gateways paralelos. O primeiro é o chamado AND-split iniciando as três ramificações simultâneas e o segundo é o chamado AND-join. Essas duas atividades podem ser realizadas várias vezes. Após

isso, há novamente duas atividades que ocorrem simultaneamente: sair da emergência (se) e entrar na UTI (eu); seguidas pela atividade de administrar vasopressores. Por fim, há novamente um símbolo de diamante com X dentro, sendo o primeiro chamado de XOR-split, que é o ponto onde ocorre uma decisão e o fluxo se divide em múltiplas opções, mas somente uma delas pode ser escolhida. Nesse caso, só há duas possibilidades excludentes para um paciente: alta ou morte. O segundo é chamado de XOR-join, que é o ponto onde essas opções divergentes de um XOR-Split voltam a convergir em um único caminho. Apenas o caminho que foi escolhido no XOR-Split anterior chegará a este ponto de convergência. O processo BPMN começa com um evento inicial (primeiro círculo) e termina com um evento final (último círculo).

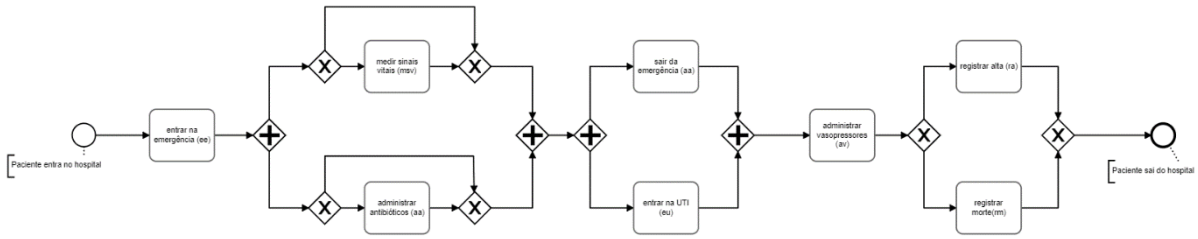

Figura 2 - Modelo BPMN do "processo tratar paciente".
Fonte: Elaboração própria.

2.5 *Conformance checking*

A verificação de conformidade relaciona eventos no log com atividades no modelo de processo (considerado como referência) e compara ambos. O objetivo é encontrar semelhanças e discrepâncias entre o comportamento previsto (no modelo) e o comportamento observado (no log). No caso deste estudo, o modelo foi descoberto automaticamente com o objetivo de mostrar o comportamento dominante, então os desvios correspondem a um comportamento excepcional (ou seja, outliers). Isso foi feito dado que a maior dificuldade de especificar o BPMN padrão é o fato de que muitas dessas atividades não possuem ordem. Além disso, montar um protocolo que refletisse o MIMIC-IV também é um desafio. De maneira geral, a verificação de conformidade requer um log de eventos e um modelo de processo como entrada. O objetivo é indicar onde o log e o modelo discordam (VAN DER AALST, 2022).

Existem várias técnicas para checar a conformidade de processos. As duas mais usadas são o *token-based replay* e *alignments*. No *replay* baseado em tokens, o modelo do processo é representado como uma Rede de Petri e os rastros no log de eventos são reproduzidos no modelo. Se o rastro indica que uma atividade deve ocorrer, a transição correspondente é

executada. Se não for possível devido a uma entrada vazia, um "token" faltante é adicionado. Tokens nunca consumidos são chamados de tokens restantes. A quantidade de tokens faltantes e restantes em relação aos consumidos e produzidos indica a gravidade do problema de conformidade. O replay baseado em tokens pode ser estendido para Redes de Petri com atividades silenciosas e duplicadas usando heurísticas. Por exemplo, se houver duas atividades com o mesmo rótulo, escolhe-se a que está habilitada. Se ambas estiverem habilitadas, escolhe-se uma delas. Da mesma forma, transições silenciosas (ou seja, transições não correspondentes a atividades registradas) são executadas quando habilitam uma transição correspondente à próxima atividade no log de eventos. Isso exige uma exploração dos estados alcançáveis a partir do estado atual e pode levar a resultados inconclusivos (VAN DER AALST, 2022). O *token-based replay* é eficiente, mas nem sempre produz caminhos válidos através do modelo.

Por outro lado, os "alignments" (alinhamentos) são considerados o melhor padrão para verificar a conformidade, pois fornecem caminhos no modelo que se aproximam ao máximo do comportamento observado (VAN DER AALST, 2022). O objetivo é mapear o comportamento observado no comportamento modelado, para fornecer melhores diagnósticos e relacionar também os casos que não se ajustam ao modelo. Os alinhamentos foram introduzidos para superar as limitações do *token-based replay*. Eles proporcionam diagnósticos mais detalhados e precisos, pois cada comportamento observado é mapeado em um comportamento de modelo que é o mais próximo possível do observado (VAN DER AALST, 2022). Os alinhamentos mostram o comportamento comum, mas também eventos pulados e inseridos que indicam desvios. Esses eventos são mais fáceis de interpretar do que os tokens ausentes ou remanescentes. No entanto, para grandes registros de eventos e processos, os cálculos de alinhamento podem ser intratáveis. Além disso, pode haver muitos alinhamentos ótimos, tornando os diagnósticos não determinísticos.

Em relação a ferramentas utilizadas para realizar esse processo, a principal e utilizada neste trabalho é o Celonis, que possui o “Conformance Checker”. Como descrito pelo próprio site: “O verificador de conformidade permite comparar automaticamente um modelo de processo de referência com os fluxos de processo reais descobertos a partir dos dados. A diferença entre o modelo e os fluxos reais é retornada na forma de uma lista de violações, que pode incluir comportamentos observados que não são permitidos pelo modelo, bem como comportamentos que estão no modelo, mas não observados na realidade” (CELONIS, 2022). Além disso, existem outras ferramentas de *conformance checking* conhecidas pelo mercado,

como ProM, Process Mining Toolkit (PMT) e Disco, mas que não foram discutidas no presente estudo.

3 Materiais e Métodos

3.1 MIMIC-IV

Conforme definido anteriormente, o presente trabalho tem como objetivo avaliar os processos de atendimento de sepse em uma base de dados real utilizando ferramentas de mineração de dados. Nesse sentido, foi selecionada como base de dados para o estudo a base do MIMIC-IV (JOHNSON et al., 2023) (GOLDBERGER et al., 2000), cujo detalhamento será apresentado em seguida. Dados médicos coletados retrospectivamente podem melhorar o atendimento ao paciente através da descoberta de conhecimentos e desenvolvimento de algoritmos. A reutilização abrangente de dados médicos é benéfica ao público, mas o compartilhamento deve proteger a privacidade do paciente. A base de dados MIMIC-III forneceu dados de cuidados intensivos para mais de 40.000 pacientes no *Beth Israel Deaconess Medical Center* (BIDMC), sendo desidentificada conforme a regulamentação HIPAA. O MIMIC-III impulsionou pesquisas em informática clínica, epidemiologia e aprendizado de máquina.

O MIMIC-IV, uma atualização do MIMIC-III, possui dados contemporâneos e melhorias. O MIMIC-IV adota uma abordagem modular na organização de dados, facilitando o uso individual e combinado de fontes distintas, e visa dar continuidade ao sucesso do MIMIC-III, apoiando diversas aplicações na área da saúde. O MIMIC-IV v2.2 foi lançado em Janeiro de 2023 e o número atualizado de pacientes, admissões e estadias na UTI são 299712 pacientes (era 315460 na v2.0), 431231 admissões (era 454324 na v2.0) e 73181 estadias na UTI (era 76943 na v2.0), respectivamente.

O MIMIC-IV é originado de dois sistemas de banco de dados hospitalares: um EHR hospitalar personalizado e um sistema de informações clínicas específico para UTI. A criação do MIMIC-IV ocorreu em três etapas:

- Aquisição: Dados de pacientes admitidos no departamento de emergência ou UTI do BIDMC foram extraídos dos respectivos bancos de dados hospitalares. Uma lista mestra de pacientes foi criada contendo todos os números de registro médico entre 2008-2019.
- Preparação: Os dados foram reorganizados para facilitar análises retrospectivas, incluindo a desnormalização de tabelas, remoção de registros de auditoria e

reorganização em menos tabelas. Não foram realizadas etapas de limpeza de dados para garantir que reflitam um conjunto de dados clínicos do mundo real.

- Desidentificação: Identificadores de pacientes conforme estipulado pela HIPAA foram removidos. Identificadores foram substituídos por números inteiros aleatórios e desidentificados. Dados estruturados foram filtrados usando listas de verificação e permissões. Um algoritmo de desidentificação de texto livre foi aplicado, se necessário. Data e hora foram deslocadas aleatoriamente para o futuro, mantendo consistência interna entre os dados de um único paciente.

O MIMIC-IV é agrupado em dois módulos: "*hosp"* e "*icu"*. O módulo "*hosp"* contém dados do EHR hospitalar, incluindo informações de pacientes, hospitalizações e transferências intra-hospitalares. Ele fornece informações temporais de pacientes, como "*anchor_year*", "*anchor_year_group*" e "*anchor_age*", permitindo inferir o ano aproximado em que o paciente recebeu cuidados. A coluna "*anchor_year*" é um ano deslocado para o paciente, é um intervalo de anos, o "*anchor_year*" do paciente ocorreu durante esse intervalo e "*anchor_age*" é a idade do paciente no "*anchor_year*", se a "*anchor_age*" de um paciente for superior a 89 anos no "*anchor_year*", sua "*anchor_age*" será definida como 91, independentemente da idade real. A data de óbito está disponível na coluna "*dod*" da tabela "*patients*", derivada de registros hospitalares e estaduais. As datas de óbito ocorridas mais de um ano após a alta hospitalar são censuradas como parte do processo de desidentificação. O módulo "*hosp"* também inclui dados de laboratório, culturas microbiológicas, pedidos de médicos, administração de medicamentos, prescrições médicas, informações de cobrança hospitalar, dados de prontuário médico eletrônico e informações relacionadas a serviços. Informações sobre provedores de cuidados estão disponíveis na tabela "*provider*". O "*provider_id*" é uma sequência de caracteres desidentificada que representa um único provedor de cuidados.

O módulo "*icu*" possui dados do sistema de informações clínicas do BIDMC: MetaVision (iMDSoft). As tabelas foram desnormalizadas, criando um esquema estrela conectando "*icustays*" e "*d_items*" a tabelas com sufixo "*events*". O módulo "*icu*" inclui entradas intravenosas e fluidas ("*inputevents*"), ingredientes ("*ingredientevents*"), saídas do paciente ("*outputevents*"), procedimentos ("*procedureevents*"), informações de data/hora ("*datetimeevents*") e outros dados registrados ("*chartevents*"). Todas as tabelas "*events*" contêm colunas "*stay_id*" e "*itemid*", identificando pacientes na UTI e conceitos documentados. A tabela "*caregiver"* possui "*caregiver_id*", um número desidentificado representando o

provedor de cuidados. Todas as tabelas "*events*" têm uma coluna "*caregiver_id*" vinculada à tabela "*caregiver*".

Além disso, existe o MIMIC-IV-ED, uma base de dados complementar ao MIMIC-IV padrão. O MIMIC-IV-ED é um vasto e acessível repositório de dados relacionados a internações no departamento de emergência (DE) do Centro Médico Beth Israel Deaconess, no período entre 2011 e 2019. Este banco de dados abrange aproximadamente 425.000 estadias no DE. Inclui informações sobre sinais vitais, triagem, reconciliação de medicamentos, administração de medicamentos e diagnósticos de alta. Todos os dados são anonimizados em conformidade com a disposição Safe Harbor da Lei de Portabilidade e Responsabilidade de Informações de Saúde (HIPAA). O MIMIC-IV-ED visa dar suporte a uma ampla variedade de iniciativas educacionais e projetos de pesquisa. O MIMIC-IV-ED é composto por uma única tabela de rastreamento de pacientes, chamada "*edstays*", e cinco tabelas de dados: "*diagnosis*", "*medrecon*", "*pyxis*", "*triage*" e "*vitalsign*".

A versão do MIMIC-IV e MIMIC-IV-ED utilizada é a v2.2, de Janeiro de 2023. Os links específicos de cada projeto são, respectivamente: https://physionet.org/content/mimiciv/2.2/ e https://physionet.org/content/mimic-iv-ed/2.2/. No total, há 32 tabelas, separadas em 3 módulos: "*hosp*", "*icu*" e "*ed*". Para averiguar a relação entre as tabelas, foi utilizado a documentação: https://mimic.mit.edu/docs/iv/. Nela, ao pesquisar por uma tabela, é possível ver a sua relação com outras, em texto como *"Links to patients on subject_id"*, por exemplo. Também existe um dicionário de dados ("Dicionário de dados MIMIC-IV.xlsx") presente no Github com as descrições detalhadas de cada tabela e seus respectivos campos. Além de um esquema relacional das tabelas utilizadas (Figura 3). Para o intuito do trabalho, foram consideradas os seguintes módulos e tabelas (módulo/tabela):

- hosp/patients: sexo, idade e data da morte do paciente, se houver informações.
- hosp/admissions: informações detalhadas sobre internações hospitalares.
- hosp/diagnoses_icd: diagnósticos faturados da CID-9/CID-10 para hospitalizações.
- hosp/d_icd_diagnoses: tabela de dimensões para "*diagnoses_icd*"; fornece uma descrição dos diagnósticos faturados da CID-9/CID-10.
- hosp/prescriptions: medicamentos prescritos.
- hosp/labevents: medições de laboratório provenientes de espécimes derivados de pacientes.
- hosp/d_labitems: a tabela de dimensões para "*labevents*" fornece uma descrição de todos os itens de laboratório.

- hosp/microbiologyevents: culturas de microbiologia.
- ed/edstays: é a tabela de rastreamento principal para visitas ao departamento de emergência.
- ed/diagnosis: fornece diagnósticos faturados para os pacientes.
- ed/vitalsign: armazena sinais vitais de rotina medidos a cada 1-4 horas.
- ed/pyxis: disponibiliza informações para dispensação de medicamentos via sistema Pyxis.
- icu/icustays: informações de rastreamento para estadias na UTI, incluindo tempos de admissão e alta.
- icu/chartevents: itens registrados ocorridos durante a internação na UTI. Contém a maioria das informações documentadas na UTI.
- icu/procedureevents: Procedimentos documentados durante a permanência na UTI (por exemplo, ventilação), embora não necessariamente conduzidos dentro da UTI (por exemplo, radiografia).
- icu/d_items: tabela de dimensões que descreve o "*itemid*". Define conceitos registrados na tabela de eventos do módulo ICU.

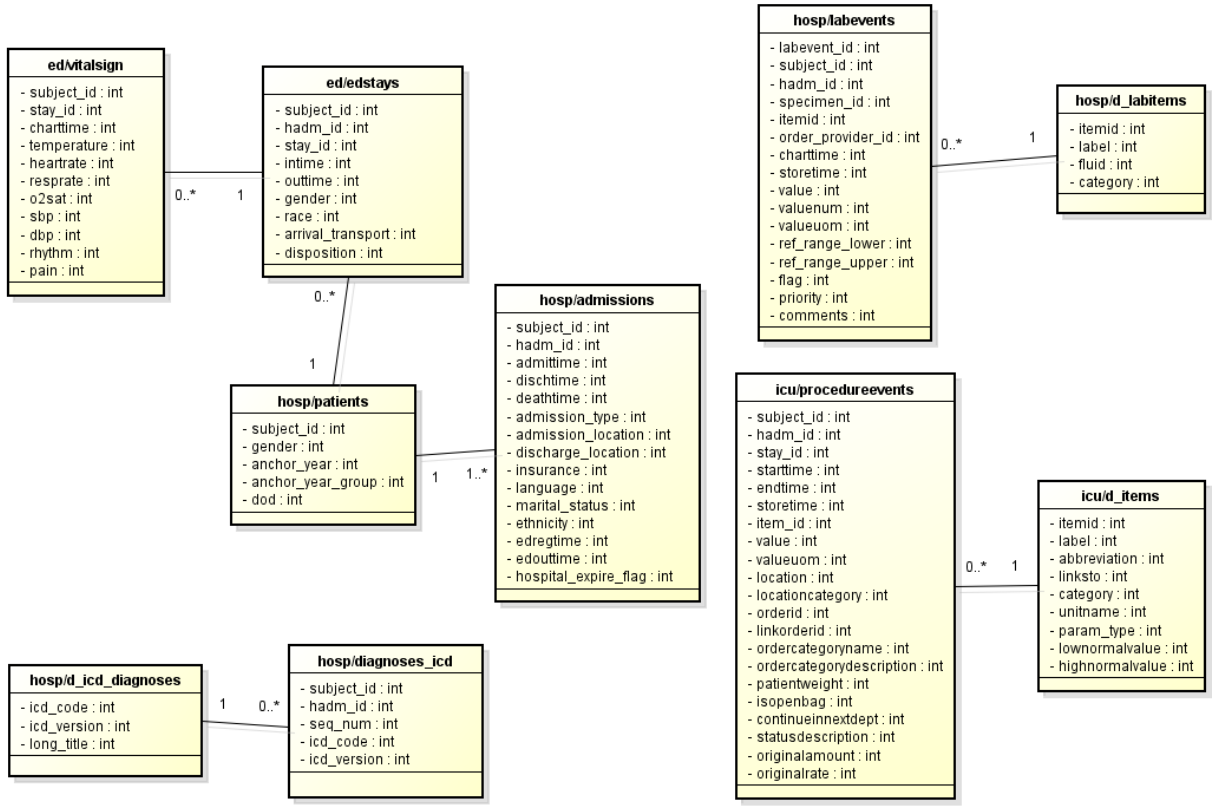

Figura 3 – Esquema relacional das tabelas utilizadas.
Fonte: Elaboração própria.

3.1.1 Obtenção/extração dos dados

Primeiramente, vale ressaltar que um descritivo de palavras técnicas da área está presente no Apêndice A. Para ter acesso aos dados, foi necessário tornar-se um usuário credenciado da PhysioNet (https://physionet.org/), uma plataforma online que oferece acesso gratuito a uma grande quantidade de dados fisiológicos e software associado. Para isso, bastou fornecer informações pessoais e relativas à pesquisa acadêmica e aguardar a aprovação. Além disso, foi requisitado completar um treinamento chamado *"CITI Data or Specimens Only Research*", um curso que cobre aspectos importantes da pesquisa com dados de participantes humanos. Por fim, para ter acesso aos dados é obrigatório assinar o contrato de uso de dados para o projeto em questão. Os dados estão em formato ".zip", em pastas e com arquivos ".csv" de cada uma das tabelas mencionadas anteriormente. É importante ter em mente que, para o MIMIC-IV, o tamanho total do arquivo zipado é de 7,1 GB e do totalmente descompactado é 65,3 GB. Em relação ao MIMIC-IV-ED, o tamanho total do arquivo zipado é de 116,3 MB e do totalmente descompactado é 703 MB.

3.1.2 Instrumento de procedimento

Antes do tratamento dos dados, é importante definir alguns passos para a melhor execução de processo. Com base no método para extrair logs de eventos do MIMIC-IV de Cremerius et al. (2023), derivado de Jans et al. (2019), seis passos foram definidos previamente, são eles:

1. Definir o principal objetivo comercial do projeto: para uma preparação útil do log de eventos, o objetivo do projeto de mineração de processos precisa ser definido. No caso deste estudo, este é analisar a conformidade com as diretrizes clínicas em pacientes com sepse.
2. Definir coorte[1](s) de pacientes: pacientes com mais de 18 anos com sepse.
3. Definir a noção de caso: deve ser selecionado um atributo que determina a instância do processo, ou seja, o "*case_id"* em *Process Mining*. No MIMIC-IV, existem duas possíveis noções de casos: o identificador do sujeito (o paciente com seu

[1] Grupo de indivíduos que compartilham uma característica em comum e são acompanhados ao longo do tempo para fins de pesquisa ou estudo. Essa característica pode ser uma condição médica, uma exposição a um fator de risco, uma idade específica, entre outros critérios.

"*subject_id*") ou o identificador da administração hospitalar ("*hadm_id*"). Com o "*subject_id*", o histórico completo do paciente, incluindo várias admissões, pode ser analisado. Com "*hadm_id"*, cada admissão de paciente é representada como um traço individual no registro de eventos. No nosso caso, definimos o "*hadm_id*", para considerar as internações separadamente.

4. Selecionar os atributos do caso: depois que as coortes de pacientes e a noção de caso tiverem sido selecionadas, na próxima etapa, atributos adicionais de casos precisam ser selecionados. Os atributos de caso podem ser usados para filtrar e agrupar no projeto de mineração de processo. No caso do presente estudo, a principal forma de filtro foi o diagnóstico presente em "*hosp/diagnoses_icd*".
5. Selecionar os tipos de eventos e seus atributos: tabelas chave e seus relacionamentos precisam ser identificados. De todas as tabelas presentes no MIMIC-IV, 16 foram selecionadas como mencionado no tópico "MIMIC-IV" deste capítulo e mais exploradas no subtópico "Geração do log".
6. Enriquecer os atributos dos eventos: opcionalmente, os eventos podem ser enriquecidos por atributos de eventos adicionais de qualquer outra tabela no MIMIC-IV se os eventos tiverem vários *timestamps*. Isso foi feito, por exemplo, na tabela de admissões, "*hosp/admissions"*. O uso de eventos adicionais é melhor explicitado no subtópico "Geração do log".

3.1.3 Estratificação de gravidade e identificação da população de interesse

Após a extração, foi utilizado como ferramenta principal o Python a fim de manipular os dados e obter *insights* sobre estes. Como ponto de partida, o objetivo era identificar a população de interesse, ou seja, pacientes com diagnóstico de sepse e com qSOFA maior ou igual a 2. Para isso, foram utilizados diversos métodos do Python para a obtenção de uma tabela em que fosse possível a análise. Mais detalhes técnicos sobre esse processo estão disponíveis no Apêndice B.

Após o processo descrito, obtemos uma tabela com informações mais detalhadas sobre os pacientes. Dado que, para o cálculo do qSOFA (SINGER et al., 2016), somente a frequência respiratória e a pressão arterial sistólica são relevantes, foram removidas linhas que apresentavam valores inexistentes para qualquer uma dessas duas medições. De acordo com o critério do qSOFA, para o paciente com frequência respiratória maior ou igual a 22/min um

ponto é adicionado ao score. Isso é aplicável, da mesma forma, para quando a pressão arterial sistólica for menor ou igual a 100 mm Hg. No caso do MIMIC, foi considerado que, de acordo com os sinais vitais, caso o paciente apresentasse esses critérios, em qualquer momento das medições, o score seria considerado e permaneceria inalterado, mesmo que outras medições fossem normais. Com o cálculo do qSOFA realizado, descrito também no Apêndice B, O *DataFrame* resultante foi exportado para excel, por meio do método "*to_excel"*, com o nome de "'pacientes_sepse_qSOFA_2.xlsx'", com o intuito de ser utilizado posteriormente nas análises.

3.1.4 Partição tabela de *icu/chartevents* e *hosp/labevents*

Dado que a tabela **"***icu/chartevents"* possui 28,1 GB de tamanho e 313.645.063 de linhas, foi necessário particioná-la para melhorar o tempo de execução. Por meio de uma função chamada "particionar_csv" que particiona um arquivo CSV grande em vários arquivos menores. Essa função possui como parâmetros o nome do arquivo grande, o tamanho de cada partição e o diretório onde serão salvos os arquivos menores. Esses arquivos menores são salvos em um diretório específico. Essa função faz esse processo em 4 etapas:

1. Primeiro, verifica se o diretório onde guardar as partes do arquivo existe. Se não existir, é criado um novo. Isso é feito a partir dos métodos *path.exists* e *makedirs* da biblioteca *os* do Python.
2. Lê o arquivo grande, mas um pedaço por vez, por meio do método *read_csv* do pandas, que é usado para ler um arquivo csv e com argumento *chunksize* igual ao tamanho de partição, para conseguir ler por "blocos".
3. Para cada pedaço, este é salvo como um novo arquivo pequeno no diretório escolhido.
4. Por fim, exibe que o pedaço foi salvo como um novo arquivo e onde ele foi salvo. Isso é feito por meio do método "*print"* do Python, que é usado para exibir uma *string* no console. A mensagem exibida é: "Partição X salva como Y", sendo X o número da partição e Y o nome do arquivo, que segue o padrão de "particao_X".
5. Como parâmetros, o nome do arquivo é "*hosp/labevents.csv*", o tamanho da partição é 500.000, ou seja, 500.000 linhas por vez, e o diretório de saída escolhido foi "particoes_chartevents". No total, o diretório possui 628 partições e um tamanho de

29,7 GB. O nome do arquivo com o código completo é “particionar.py” e pode ser encontrado no Github desse trabalho.

Esse mesmo processo foi realizado para a tabela “*hosp/labevents”,* por também ter um tamanho elevado (12,7 GB) e 118.171.367 linhas. Nesse caso, o diretório de saída é chamado de “particoes_labevents”, tendo 237 partições e um tamanho de 13,3 GB.

3.1.5 Filtro pacientes com sepse e qSOFA igual a 2 em “*icu/chartevents”* e *“hosp/labevents”*

Tendo em vista o grande tamanho das tabelas “*icu/chartevents” e “hosp/labevents”,* além do fato de somente os pacientes com diagnóstico de sepse e qSOFA igual a 2 serem relevantes para o trabalho, foi feito um filtro nos diretórios “particoes_chartevents” e “particoes_labevents” a fim de obter os dados somente dos pacientes pertinentes. Por meio de uma função chamada “filtrar_pacientes”, sem parâmetros, um arquivo Excel é lido e conta a quantidade de arquivos em um diretório específico sendo que, para cada arquivo, combina, por meio do método *merge*, o conteúdo dele com o do arquivo Excel com base em duas colunas específicas (“*subject_id*” e “*hadm_id*”). Dessa forma, a partição só contém os dados de pacientes presentes no arquivo “pacientes_sepse_qSOFA_2.xlsx”. Por fim, salva o resultado da combinação como um novo arquivo CSV em outro diretório. E ao final de cada combinação, imprime o número do arquivo processado. Um dos diretórios finais é chamado de “particoes_chartevents_filtradas”. Com esse processo foi possível reduzir o tamanho da pasta de 29,7 GB para 1,70 GB. O outro, é chamado de “particoes_labevents_filtradas” e, com isso, foi possível reduzir o tamanho da pasta de 12,7 GB para 198 MB.

3.1.6 Tabela de medicamentos

Com auxílio de um médico intensivista especialista da área, definiu-se os medicamentos que seriam importantes para o tratamento de sepse. Nessa tabela existem 4 campos: “Generic name”, “Brand names”, “Tipo esp” e “Tipo”. Uma linha é, por exemplo: Amikacin, Amikin, Aminoglycoside e Antibiotic. Com isso, além do nome do medicamento, temos o seu tipo, que pode ser: *Fluids*, *Antibiotic*, *Vasopressors*, *Corticosteroids*. Nessa tabela existem 144 medicamentos únicos. Para conferir quais remédios dessa tabela estavam na base

(“*hosp/prescriptions”*), primeiro foram obtidos os medicamentos únicos da tabela “*prescriptions”.* Isso foi feito utilizando o método *unique* do pandas, para pegar somente os registros de medicamentos somente uma vez na coluna “*drug*” e, assim uma tabela chamada “df_presc_unicas” foi gerada. Após isso, por meio da biblioteca FuzzyWuzzy do Python, que usa Distância Levenshtein (RISTAD et al., 1998, p. 7) para calcular as diferenças entre as *strings*, o nome de cada remédio da tabela “remedios” foi verificado na tabela “*prescriptions”* e determinado se havia uma correspondência de pelo menos 80%.

A biblioteca FuzzyWuzzy utiliza essa métrica, no nosso caso, por meio do método *process.extractOne.* Isso retorna a melhor correspondência da coluna "*Generic name*" do *DataFrame* para a palavra, que no caso é o medicamento, usando o “*fuzz.token_sort_ratio”* como métrica de similaridade. O uso desse método implica que a ordem das *strings* não importa. Por exemplo, as *strings* "O gato perseguiu o rato" e "O rato perseguiu o gato" teriam uma pontuação de similaridade de 100% usando “*fuzz.token_sort_ratio*”, porque ambas contêm exatamente os mesmos tokens, apenas em ordens diferentes. Para informações mais detalhadas dessa biblioteca, segue o link: https://github.com/seatgeek/thefuzz.

Com isso, foi criada uma coluna chamada “Correspondência” na tabela “df_presc_unicas” para mostrar qual nome corresponde ao visto na tabela “remedios”. Por exemplo, uma linha (Tabela 1) é:

Tabela 1 - Exemplo de uso da correspondência.

Nome remédio	Correspondência
Vancomycin Intraventricular	Vancomycin

Fonte: Elaboração própria.

“Nome remédio” é o nome que está em “*prescriptions”* e “Correspondência” o que está em “remedios”. Para facilitar o processo de correspondência, foi utilizado o método “*split”* para obter somente a primeira parte da *string* a ser comparada. No exemplo acima, ao utilizar “*split”* e pegar o primeiro elemento, este seria “Vancomycin” e “Intraventricular” seria desconsiderado, já que não é importante para o processo de correspondência.

Em seguida, as correspondências vazias foram descartadas. No caso específico do lactato, como o nome estava diferente da base para a tabela “remedios”, foi feita uma função “lactato” para fazer uma correspondência manual. Com um merge entre essa tabela resultante e a tabela “remedios” usando o “*Generic Name*”, a tabela decorrente dessa operação foi

exportada com o nome “remedios_de_para.xlsx”, sendo usada posteriormente para a confecção do *log* de eventos.

Além disso, havia um erro na tabela “remedios” em que um dos medicamentos era “*Nor-adrenaline”.* Contudo, na tabela “*prescriptions”,* havia somente “*norepinephrine*”, ou seja, não houve correspondência, sendo que norepinefrina é um outro termo para noradrenalina. Da mesma forma que a correspondência de lactato foi feita manualmente, a de noradrenalina também foi. No final, a tabela “remedios_de_para.xlsx” possui 221 medicamentos diferentes. Contudo, muitos dos registros são um medicamento em diferentes formas, por exemplo: albumina 5% e albumina 25%. Na base, ainda existem erros de grafia, mesmo com esse empecilho, utilizando a biblioteca FuzzyWuzzy foi possível obter a correspondência, que foi o elemento principal para análise. Um exemplo de erro de grafia é no caso da albumina, que em alguns momentos aparece sem o “n”, com parênteses faltando ou sem o símbolo de “%”.

O mesmo processo foi realizado para a tabela “*ed/pyxis”* a fim de obter correspondência com a tabela “remedios.xlsx”. A tabela resultante desse procedimento é chamada de “remedios_de_para_pyxis”, possuindo 111 medicamentos únicos. Por fim, a coluna que foi utilizada, de fato, para análise é a “Tipo”, tendo 6 possibilidades: *Fluids, Antibiotic, Vasopressors* e *Corticosteroids*. Isso foi feito dado que, no protocolo e orientações sobre a sepse, na maioria dos casos, o nome do remédio em si não importa, somente a qual classe pertence é relevante. Vale ressaltar que, na base *“ed/pyxis”*, não há nenhum fluido presente em “remedios.xlsx”. Isso não quer dizer que não há fluido nenhum, mas os presentes na tabela de remédios não existem nessa base. Além disso, com o auxílio do especialista da área médica, foi considerado que antibióticos e antimicrobianos são o mesmo tipo de medicamento, a fim de facilitar a busca e seu ajuste às recomendações de tratamento da sepse.

3.1.7 Geração do log

A fim de obter-se o log de eventos (uma coleção de eventos correspondentes ao processo selecionado) a ser analisado, foram utilizadas as 16 tabelas mais relevantes já mencionadas. De maneira geral, o objetivo era obter um log em que cada linha refere-se a um evento com sete atributos principais (“*subject_id”*, “*hadm_id”*, “*activity”*, “*column”*, “*resource”*, “*timestamp”*, “*add. Info”*), incluindo os três obrigatórios: *case* (“*hadm_id”*), *activity* e *timestamp*. Com a importação das bases já realizada, foi utilizado o método “*merge”* entre *“hosp/admissions”* e *“hosp/patients”* na coluna *“subject_id”* para obter informações detalhadas sobre os pacientes

e suas admissões. Em seguida foi feito outro *"merge"*, mas agora com a tabela oriunda do arquivo "pacientes_sepse_qSOFA_2.xlsx", para filtrar somente os pacientes relevantes. Com essa tabela resultante, foi criada uma função para iterar sobre as linhas e, dependendo da coluna, transformá-la em uma nova linha na tabela nova, com o uso do método "*apply"* do pandas. O nome dessa função é *"transform_row"*, com a linha como parâmetro e foi usada, de forma semelhante, em todas as tabelas. A tabela resultante é chamada de "df_transformedX", sendo X um número de 1 a 8. Um exemplo de um uso dela é (Figura 4):

```
def transform_row(row):
    return {
        'subject_id': row['subject_id'],
        'hadm_id': row['hadm_id'],
        'stay_id': row['stay_id'],
        'Activity': "Measure the patient's vital signs.",
        'Coluna': 'charttime',
        'Resource': 'vitalsign/ed',
        'Timestamp': row['charttime'],
        'Info adc': f" Resprate {row['resprate']} - SBP {row['sbp']}"
    }

df_transformed = df_final.apply(transform_row, axis=1)
df_transformed = pd.DataFrame(df_transformed.tolist())
df_transformed
```

Figura 4 - Exemplo de função utilizada para gerar o log.
Fonte: Elaboração própria.

A fim de deixar mais claro como funciona o processo, por exemplo, em uma linha com a coluna *"admittime"*, esse registro torna-se uma nova linha com: *"subject_id"*, *"hadm_id"*, *"stay_id"*; o evento que, nesse caso, é *"Register the date and time the patient was admitted to the hospital"*; *"admittime"*, "*admissions/hosp*" e o *timestamp* desse evento. Após o uso da função a tabela resultante (Tabela 2) ficou da seguinte forma, em um exemplo fictício:

Tabela 2 – Exemplo fictício do log.

subject_id	hadm_id	stay_id	Activity	Coluna	Resource	Timestamp
10000000	21111111	11111111	Register the date and time the …	admittime	admissions /hosp	2980-06-25 12:55:00

Fonte: Elaboração própria.

A mesma ideia foi seguida para as outras tabelas. No caso de *"ed/vitalsign"*, foi feito um merge entre a tabela antes do log, usando o *"subject_id"* e o *"stay_id"*, para obter o *"hadm_id"* nos registros dos sinais vitais. Com isso, seguindo a mesma lógica da função anterior, cada linha virou um registro para o log, em que a atividade é *"Measure the patient's vital signs"* e na coluna de informações adicionais (Info adc) há o registro da frequência respiratória e a pressão arterial sistólica, as medidas importantes para o qSOFA.

Para a tabela de remédios na emergência via Pyxis, primeiro foi feito um filtro, usando "*merge*", para pegar os registros somente dos pacientes pertinentes, da mesma forma que foi feito na tabela de admissões e de pacientes. Além disso, a tabela oriunda do processo de correspondência, "remedios_de_para_pyxis.xlsx", foi usada nessa etapa para filtrar os medicamentos relevantes presentes na base de remédios via Pyxis. Esse filtro foi feito por meio de um *"merge"* entre essas duas tabelas, usando a coluna "*name*". Assim como para as outras tabelas, foi criada uma função para transformar cada registro em um log, sendo a atividade "*Administer medication X via Pyxis in the emergency*" e "X" o medicamento em questão.

Em relação a tabela "*hosp/labevents",* dado que, anteriormente, foi realizado um filtro em suas partições, foi necessário ter um meio para juntar essas partes e gerar um *DataFrame* único. Para isso, a biblioteca Dask foi utilizada. Primeiramente, o diretório "particoes_labevents_filtradas" foi lido com o método "*read_csv"* dessa biblioteca. No entanto, diferente do método de mesmo nome do Pandas, esse consegue ler vários arquivos, de maneira paralela e distribuída. Ao ler o diretório, o seguinte formato é passado: "particoes_labevents_filtradas\particao_*.csv". O asterisco representa um caractere qualquer, ou seja, qualquer arquivo que comece com "particao_" e um, nesse caso, número, por exemplo "particao_1". Dessa forma é possível ler todo o diretório de maneira eficiente. Para facilitar a leitura, foi considerado que o tipo de todas as colunas é "*object*". Após isso, foi utilizado o método "*compute"* do Dask que executa todos os cálculos que foram definidos no *DataFrame* Dask ddf, e então converte o resultado desses cálculos em um *DataFrame* do Pandas, que foi armazenado como "pandas_labeventos", sendo o todo o resultado trazido para a memória local e, por isso, é importante ter memória suficiente no computador. Em outras palavras, "*compute"* transforma um Dask *DataFrame* em um Pandas *Dataframe*. Então, foi utilizado o método "*astype"* para transformar o tipo da coluna "*itemid*" para "*int64*", isso é importante para o próximo passo. Com a importação da tabela *"hosp/d_labitems"*, foi realizado um *"merge"* com "pandas_labeventos", usando a coluna "*itemid*", a fim de obter mais informações sobre as medições. Dado que somente o lactato é relevante, foi feito um filtro para somente extrair a

medição "*Lactate*". Para gerar o log desta tabela, a função "*transform_row*" foi utilizada pelo "*apply"* do Pandas.

Sobre a tabela "*icu/icustays",* foi realizado um merge com a tabela "*hosp/patients"*, usando as colunas "*subject_id*" e "*hadm_id*" para obter informações mais detalhadas dos pacientes e suas estadias na UTI. Um ponto a ser levado em consideração nesta etapa é que, apesar de não estar explicitamente documentado na documentação do MIMIC, a coluna "*stay_id*" na tabela "*icu/icustays"* é diferente da coluna "*stay_id*" em "*ed/edstays".* Apesar do nome ser igual, na primeira tabela, refere-se à estadia na UTI e, na segunda, à estadia na emergência. Para resolver esse problema, a coluna "*stay_id*" foi renomeada para "*stay_id_icu*" na tabela da estadia na UTI, utilizando-se o método "*rename*" do pandas. Em seguida, uma função foi criada para transformar os registros em um *log*. A particularidade nesse caso é que, para cada linha, existe o registro da coluna "*intime*" e "*outtime*", isto é, há dois tipos de atividades na mesma linha. Para solucionar essa questão, cada linha da tabela original se tornou duas na tabela de *logs*. Uma linha para o "*intime*", com atividade descrita como "'*Record the date and time the patient was transferred to the ICU.*'" e uma para o "*outtime*", com atividade "*Record the date and time the patient was transferred from the ICU*".

A tabela "*hosp/prescriptions"*, primeiramente, foi utilizada em um "*merge"* com "pacientes_sepse_qSOFA_2.xlsx", usando as colunas "*subject_id*" e "hadm_id", a fim de filtrar somente os eventos de pacientes importantes para a análise. Em seguida, foi utilizada a tabela "remedios_de_para" para filtrar somente os medicamentos relevantes para o estudo. Isso foi feito por meio de um "*merge*" da tabela resultante anterior e o *DataFrame* "df_remedios_de_para", usando a coluna "*drug*". Por fim, uma função do tipo "*transform_row*" foi utilizada e o log da tabela "*hosp/prescriptions"* foi gerado.

Para a tabela de procedimentos, *"icu/procedureevents",* inicialmente, foram feitos dois "*merge*", um com a sua dimensão "*icu/d_items",* usando a coluna "*itemid*" e outro com a tabela "pacientes_sepse_qSOFA_2.xlsx". Com isso, foi possível obter mais informações sobre os procedimentos, principalmente a sua descrição e filtrar somente os pacientes relevantes. Como citado anteriormente, uma lista de procedimentos importantes foi fornecida pelo especialista da área médica. Essa lista foi transformada em um *DataFrame* no Python, usando "*pd.DataFrame*". Dessa forma, foi feito um "*merge"* entre a tabela resultante anterior e essa de procedimentos relevantes, usando a coluna "*label*", com o objetivo de filtrar somente dados importantes. No final, uma função do tipo "*transform_row*" foi utilizada e o *log* da tabela "*icu/procedureevents"* foi gerado.

A fim de obter dados relativos ao procedimento de hemocultura, a tabela "*hosp/microbiologyevents*" foi usada. Primeiramente, foi realizado um filtro para extrair somente as linhas que continham a descrição como "*BLOOD CULTURE*". Em seguida, foi feito um "*merge"* com a tabela "pacientes_sepse_qSOFA_2.xlsx", usando as colunas "*subject_id*" e "*hadm_id*". Para transformar os dados dessa tabela em um *log*, novamente a função do tipo "*transform_row*" foi usada. A tabela resultante foi chamada de "*df_transformed8*".

No total, foram geradas 8 tabelas transformadas em *log*. Por meio do método "*concat*" do Pandas, usado para concatenar ou juntar duas ou mais estruturas de dados Pandas ao longo de um determinado eixo, as tabelas foram concatenadas de forma que, de maneira simplória, uma ficou "empilhada" na outra. Em seguida, como havia somente a coluna "*Timestamp*", avaliamos que seria melhor separá-la em duas, "*Init Timestamp*" e "*End Timestamp*", a fim de obter a duração de um evento. Em tabelas que havia esse tipo de estrutura, a função do tipo "*transform_row*" já havia tratado isso. Contudo, para tabelas em que só havia um *timestamp*, após a concatenação, caso o valor da coluna "*Timestamp*" existisse, ele seria passado, também para as colunas "*Init Timestamp*" e "*End Timestamp*", dado que consideramos que, para eventos instantâneos, o tempo inicial e final eram iguais. Isso foi feito por meio de uma função anônima com "*apply"* do Pandas.

Assim como citado em Cremerius et al. (2023) e presente em Kusuma et al. (2020), anexamos, manualmente, um registro de data e hora à tabela "*diagnoses_icd"*. Foi considerado, com o auxílio do especialista da área médica, que o momento em que é feito o diagnóstico é quando o paciente entra na emergência, ou seja, o mesmo *timestamp* da atividade "*Register the admission datetime of the given emergency department stay*". Para adicionar o diagnóstico na tabela final, foi feito um script que manipula dados de pacientes com diagnóstico de sepse, oriundos do arquivo "pacientes_diag_sepse.xlsx". Primeiro, os registros de admissão de emergência do DataFrame chamado df_paciente são filtrados. Em seguida, esses dados são combinados com o *DataFrame* df_pacientes_sepse, baseado em dois identificadores de pacientes ("*subject_id*" e "*hadm_id*"). Finalmente, adiciona essas novas informações à base dos pacientes, fazendo uma base maior com esta e as novas informações sobre sepse.

Após analisar os logs iniciais, foi percebido que, em pacientes que morreram, o evento de alta e morte tinham o mesmo *timestamp*. Para facilitar as análises posteriores, foi considerado que o evento final deveria ser morte ou alta, não ambos juntos. Para resolver essa questão, foi criado um processo de quatro passos:

1. Identificar os pacientes que morreram (aqueles com "*deathtime*" na coluna "*Column*").
2. Criar um conjunto de identificações únicas para esses pacientes (usando "*subject_id*" e "*hadm_id*").
3. Definir uma função auxiliar ("*keep_row*") que retorna Falso para as linhas que correspondem à alta hospitalar de pacientes que morreram.
4. Aplicar essa função ao *DataFrame* original para criar um novo *DataFrame* que contém apenas as linhas que devem ser mantidas.

Foi percebido, também, que após a morte, eventos estavam acontecendo, muito provavelmente por engano. Então, foi necessário excluir eventos pós alta ou morte. Para isso, os seguintes passos foram realizados:

1. Identificar as "atividades importantes" (alta ou morte).
2. Criar um *DataFrame* com apenas essas atividades.
3. Encontrar o tempo da primeira "atividade importante" para cada paciente e internação.
4. Juntar essa informação ao *DataFrame* original.
5. Filtrar o *DataFrame* para manter apenas as atividades que ocorreram até a primeira "atividade importante".
6. Remover a coluna auxiliar usada para essa filtragem.

Por fim, a atividade "*Register the date and time the patient was admitted to the hospital*" se comportava de maneira inconsistente. Pela documentação do MIMIC-IV, na tabela "*hosp/admissions*", coluna "*admittime*", a sua descrição é a mesma da atividade acima, mas, em tradução livre, seria "Registrar a data e a hora em que o paciente foi internado no hospital". Dessa forma, não é explícito se esse *timestamp* referencia a hora em que o paciente foi internado no hospital ou na UTI. Além disso, após observar o *log*, essa atividade aparecia em momentos diversos da internação, às vezes no início, outras após ou durante a UTI, etc. Para remover esse evento, foi realizado um filtro no *DataFrame* final para obter qualquer evento diferente desse indesejado. Para exportar essa tabela, foi utilizado o método "*to_csv"* e, antes, o *DataFrame* foi ordenado pelo "*subject_id*", "*hadm_id*" e "*Init Timestamp*", para facilitar sua observação.

3.1.8 BPMN padrão

O BPMN padrão foi feito com base em 3 fontes: Quintano (2019, p. 43); Kalimouttou et al. (2023) e Evans et al. (2021). A primeira fonte sugere que as principais atividades para o tratamento da sepse são: antibióticos, hemocultura, expansão de volume e lactato. Já na segunda, para o MIMIC-IV, foram consideradas, inicialmente, 22 recomendações. Dentre estas, os autores identificaram 6 itens associados a uma redução na mortalidade, sendo eles: corticosteroides, vasopressina, antimicrobiano, ringer com lactato, bicarbonato e insulina. A terceira fonte contém todas as recomendações possíveis para o tratamento da sepse, sendo 93 no total. Com o auxílio do especialista da área médica, foi possível chegar a um consenso de quais atividades eram, de fato, relevantes a serem analisadas. Seis atividades principais foram consideradas: antibiótico, vasopressor, fluidos, lactato, hemocultura e corticosteroides. Suas respectivas recomendações, presentes em Evans et al. (2021) e Kalimouttou et al. (2023) são:

- Para adultos com possível sepse sem choque, sugerimos um curso de investigação rápida por tempo limitado e, se a preocupação de infecção persistir, a administração de antimicrobianos dentro de 3 horas a partir do momento em que a sepse foi reconhecida pela primeira vez.
- Para adultos com possível choque séptico ou alta probabilidade de sepse, recomendamos a administração de antimicrobianos imediatamente, de preferência dentro de uma hora após o reconhecimento.
- Para adultos com choque séptico, sugerimos iniciar vasopressores perifericamente para restaurar a pressão arterial média, ao invés de atrasar o início até que um acesso venoso central esteja garantido.
- Para adultos com suspeita de sepse ou com sepse, sugerimos medir o lactato sanguíneo.
- Para adultos com suspeita de sepse ou choque séptico, recomendamos a realização da hemocultura em até uma hora.
- Para adultos com choque séptico e necessidade contínua de terapia vasopressora, sugerimos o uso de corticosteroides IV.

A maior dificuldade de especificar o BPMN padrão é a questão de que muitas dessas atividades não possuem ordem. Por exemplo, medir o lactato e realizar a hemocultura podem ser feitas em qualquer ordem, dado que o tempo de 1 hora seja respeitado. Além das seis atividades destacadas, foram incluídas mais 11 que são referentes a processos inerentes ao tratamento da sepse e a um hospital, como: registrar hora de alta, registrar hora em que um paciente entrou ou saiu da UTI, etc.

Sobre o modelo, após importarmos o log de eventos no Celonis, geramos na própria ferramenta um diagrama em BPMN. Contudo, nem todos os eventos relevantes estavam presentes, então foi necessário adicionar o lactato, a hemocultura e os corticosteroides. Com isso, esse modelo é utilizado na parte de conformidade. Foi usada a mesma nomenclatura do MIMIC-IV no nome das atividades do diagrama em BPMN, a fim de facilitar a análise de conformidade na etapa final do estudo. Como citado anteriormente, muitas atividades não possuem ordem específica, isso é discutido com mais profundidade no capítulo de Resultados. O diagrama em BPMN (Figura 5) e suas raias (Figura 6 e 7) ficaram da seguinte forma:

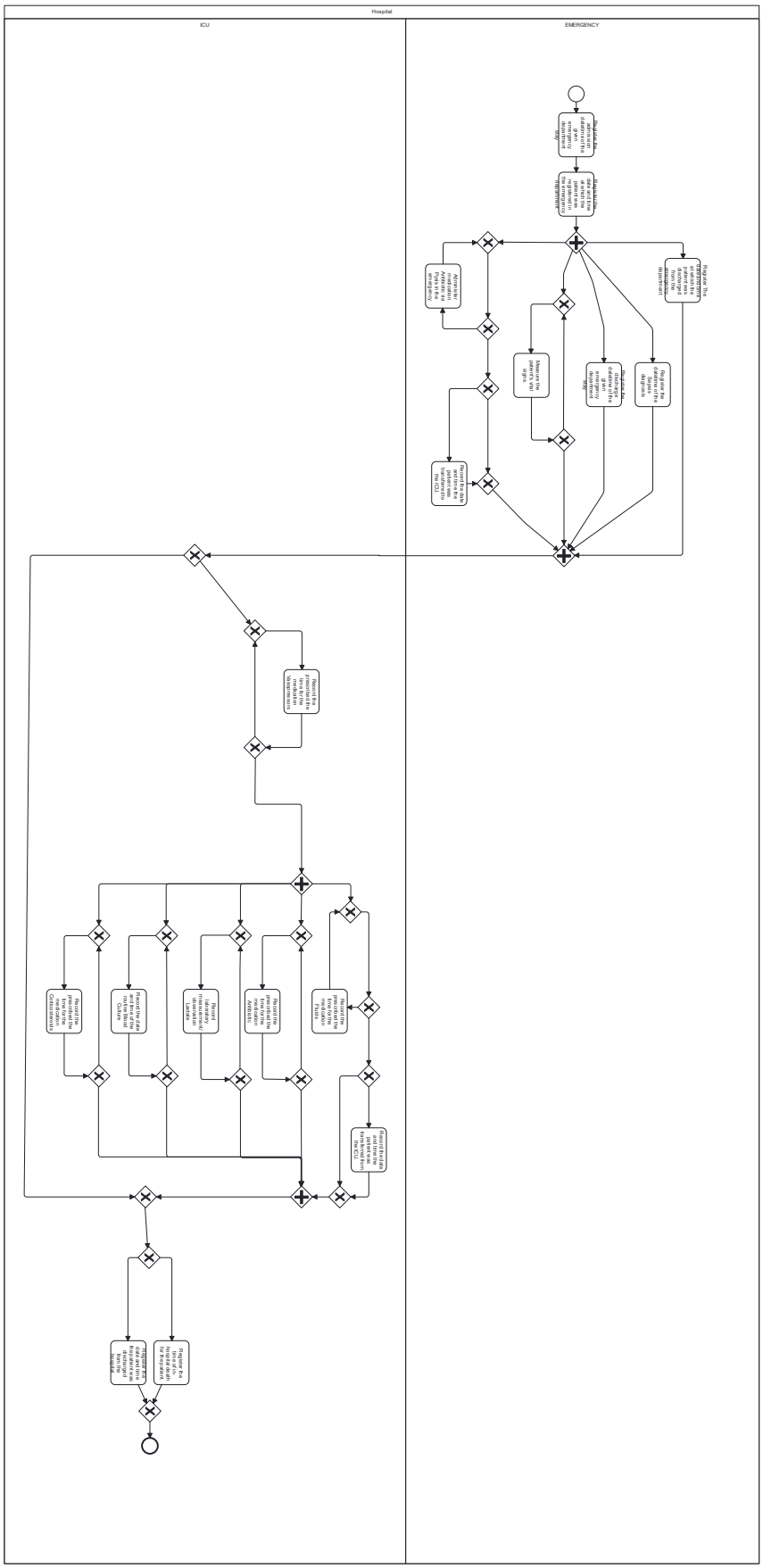

Figura 5 - BPMN padrão seguindo o protocolo de tratamento da sepse.
Fonte: Elaboração própria.

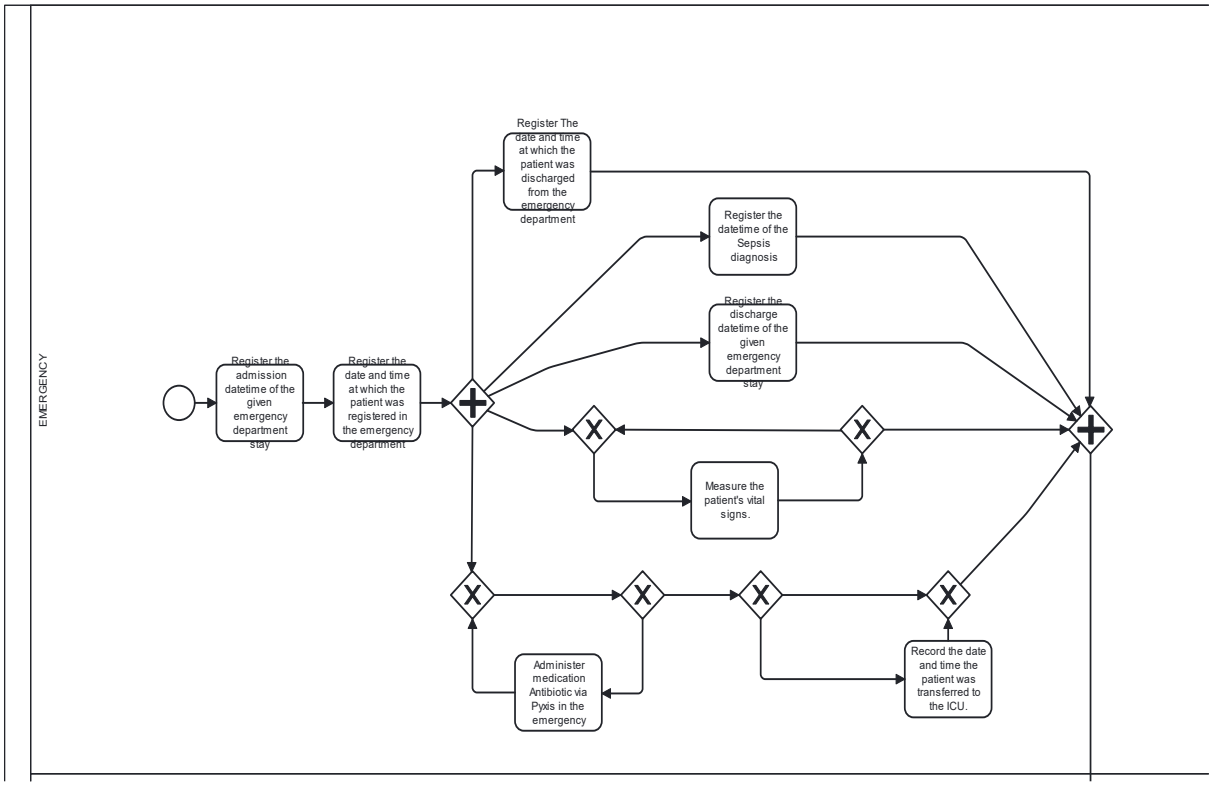

Figura 6 - BPMN padrão na emergência.
Fonte: Elaboração própria.

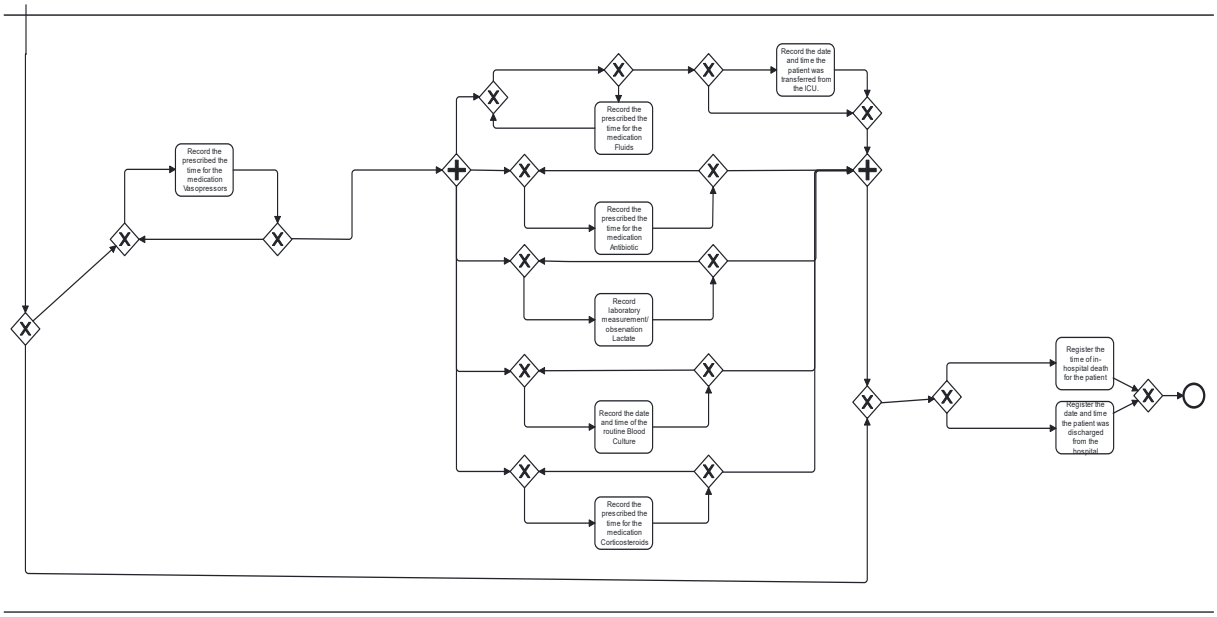

Figura 7 - BPMN padrão na UTI.
Fonte: Elaboração própria.

3.1.9 CELONIS

Segundo Vogelgesang et al. (2022), Celonis é uma ferramenta de análise de processos de negócios que utiliza técnicas de mineração de processos e inteligência artificial para obter *insights* valiosos a partir dos dados operacionais de uma organização. Com o Celonis, é possível visualizar e entender em detalhes como os processos de negócios estão sendo executados, identificar gargalos, desvios e oportunidades de melhoria, e tomar decisões informadas para otimizar a eficiência operacional.

A plataforma do Celonis permite que o usuário importe dados de várias fontes, como logs de sistemas, registros de transações, interações de clientes e outros eventos relevantes. Esses dados são processados e analisados em tempo real, fornecendo uma visão em tempo real dos processos de negócios. O Celonis usa algoritmos avançados para identificar padrões, tendências e anomalias nos dados, revelando *insights* ocultos e possibilitando uma compreensão profunda do desempenho operacional.

Com base nos *insights* gerados pelo Celonis, é possível identificar áreas de melhoria, automatizar tarefas manuais, otimizar fluxos de trabalho e implementar estratégias de melhoria contínua. A ferramenta também oferece recursos de monitoramento em tempo real, alertas e painéis interativos que permitem que os usuários acompanhem o desempenho dos processos e tomem medidas imediatas quando necessário. Em suma, o Celonis capacita as organizações a impulsionarem a eficiência operacional, reduzir custos, aumentarem a produtividade e melhorarem a experiência do cliente.

4 Resultados

Neste capítulo são discutidos e analisados os resultados da extração dos logs de eventos do MIMIC-IV através do Python e sua conexão com a ferramenta de *process mining* do Celonis. Antes, é importante ressaltar os resultados obtidos da subseção "Estratificação de gravidade e identificação da população de interesse**".** O arquivo final gerado possui 3358 internações e 3023 pacientes únicos, ou seja, um mesmo paciente pode ter mais de uma internação. De maneira ilustrativa (Figura 8), o processo de obtenção da população de interesse foi realizado da seguinte forma, sendo "n" o número de pacientes:

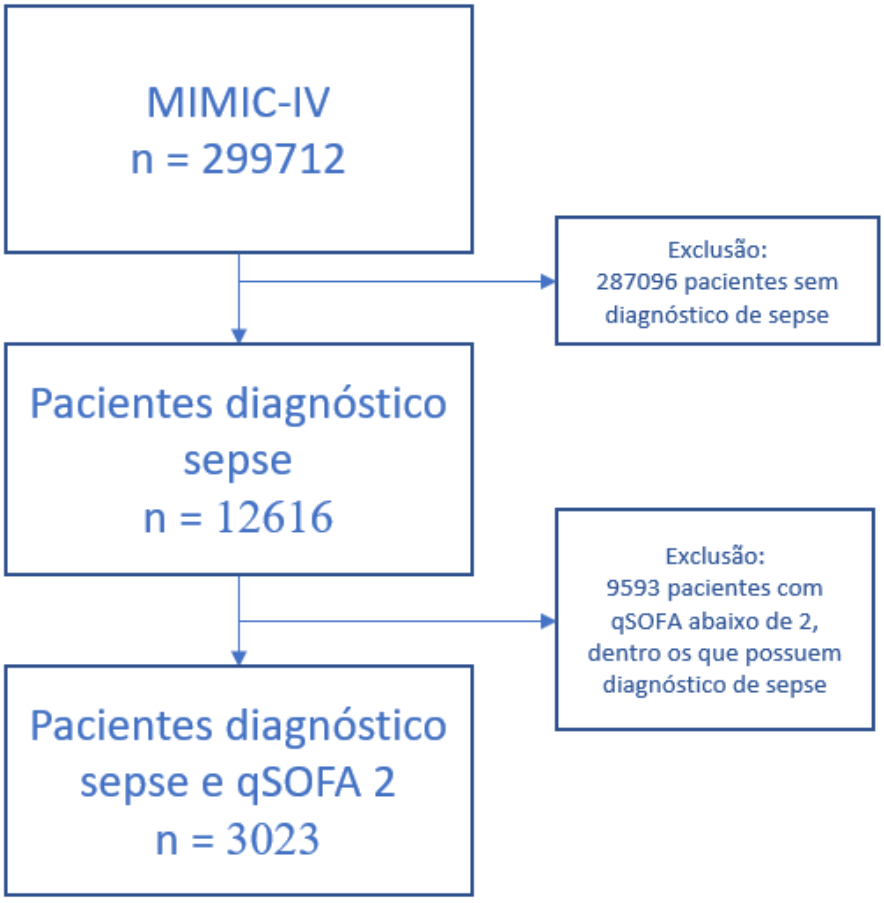

Figura 8 - Número de pacientes após cada filtro aplicado.
Fonte: Elaboração própria.

Inicialmente, com a integração de dados, foi necessário a repartição da base de dados em 2, devido a limitação de *upload* do Celonis para acesso universitário de 100.000 linhas e 1 GB, isso pode ser visto em mais detalhes na tabela (Tabela 3) a seguir.

Tabela 3 – Preparação de dados do Celonis.

	K1	K2
Quantidade de internações	1654,00	1704,00
Atividades	76362,00	76339,00
Taxa de mortalidade	20,62%	19,95%
Tempo médio de internação	260 horas	253 horas
Intervalo das internações (anos fictícios, para garantir privacidade dos dados)	2110 - 2160	2161 - 2211

Fonte: Elaboração própria.

Na Figura 9 pode-se observar uma base de dados que inclui 3358 internações e 152.701 atividades. Ela foi repartida em dois *data pools*, K1 e K2. No arquivo csv original, o número de atividades foi dividido pela metade, mas, para não perder atividades importantes de cada internação, o critério de divisão se estende até a última atividade da última internação. K1 possuindo 1654 internações com 76.362 atividades e K2 possuindo 1704 internações com 76.339.

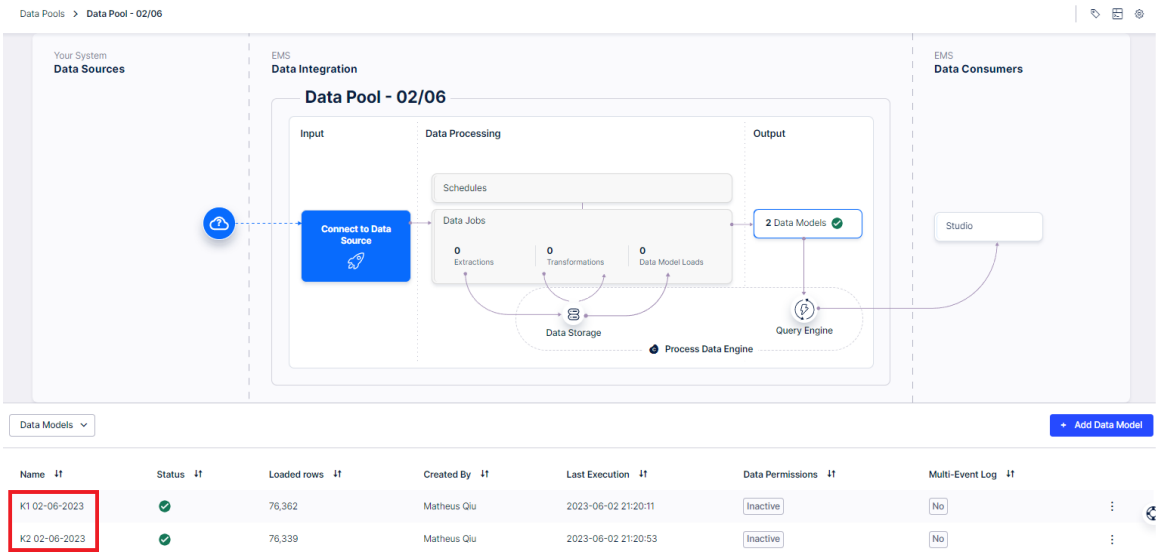

Figura 9 - Data Integration.
Fonte: Elaboração própria.

A análise de dados dos processos do MIMIC-IV revela informações sobre a aderência geral ao protocolo padrão, aderência por atividade, desvios e aderência do tempo dos processos. Esses resultados fornecem uma visão aprofundada dos cuidados e dos desfechos dos pacientes, permitindo identificar oportunidades de melhoria e otimização dos recursos.

4.1 Análise de aderência geral

Para avaliar a aderência geral ao protocolo padrão, analisamos o percentual de pacientes que passaram por todos os processos recomendados, pelo menos, de acordo com o BPMN padrão. Para chegar a esse resultado, foi feita uma média ponderada entre a conformidade de K1 e K2 (Figura 10 e 11). Além disso, na aba de desvios, colocamos na *allowlist* os que se tratavam, na teoria, de desvios na questão de ordem. Por exemplo, realizar a medição do lactato antes ou depois de verificar os sinais vitais que, neste caso, está correto. Os incorretos são tratados na parte de análise de desvios.

Conformidade em K1:

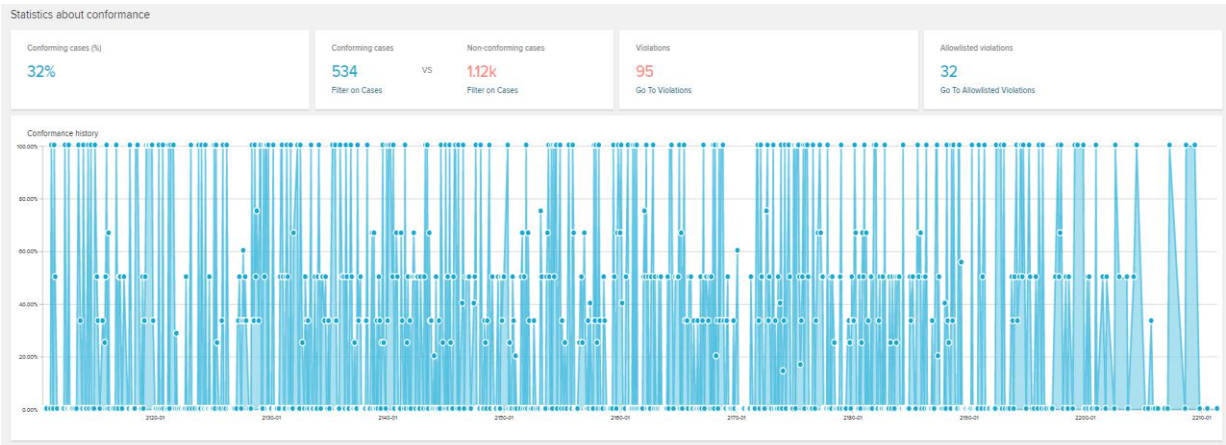

Figura 10 - Conformidade na base de dados K1.
Fonte: Elaboração própria.

Conformidade em K2:

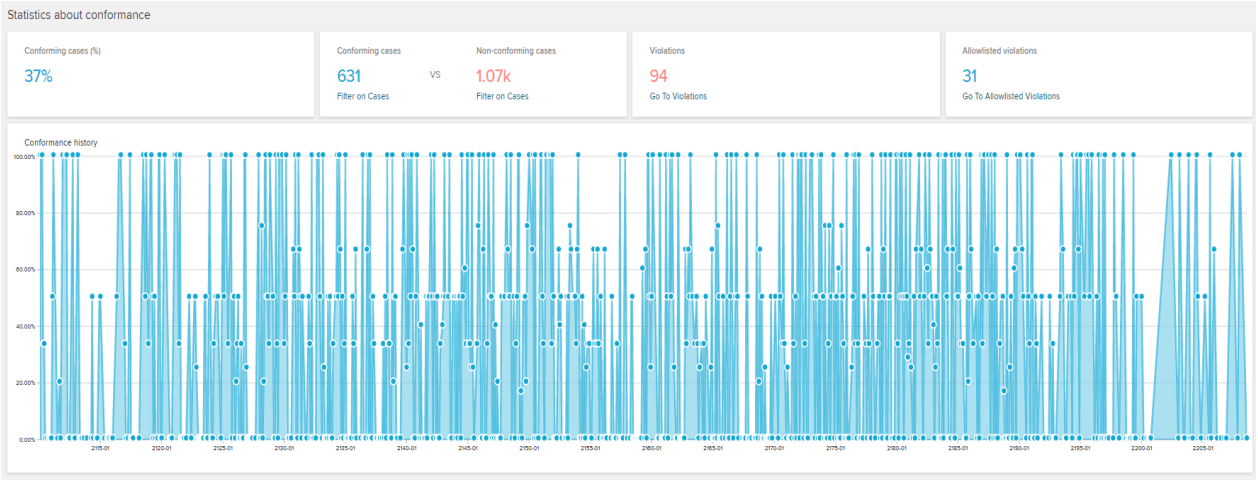

Figura 11 - Conformidade na base de dados K2.
Fonte: Elaboração própria.

Nesse sentido, a primeira análise foi verificar a aderência geral ao BPMN padrão, o resultado obtido foi de 34,69%, ou seja, nesses pacientes, pelo menos, o protocolo foi seguido. Isso significa que, por exemplo, das 17 atividades do padrão, essas ou mais foram seguidas. Contudo, é importante analisar, dentro desses 34,69% aderentes, qual foi a mortalidade, que, no caso, o valor obtido foi de 9,10%. Como comparação, essa mesma observação foi feita para o grupo de não aderentes e a mortalidade foi de 26,22%. Com esse resultado, já é possível perceber que, efetivamente, seguir o protocolo padrão resulta em uma menor mortalidade. Para as análises a seguir foram agrupadas as tabelas K1 e K2.

4.2 Análise de aderência por atividade

Outro indicador relevante para o estudo é analisar a aderência por atividade, isto é, para cada atividade do protocolo verificar qual o percentual de pacientes que passou por ele e verificar, para os grupos aderentes e não aderentes, qual a mortalidade. Isso foi feito por meio da ferramenta de “Activity selection” do Celonis. No total, o log de eventos possui 17 atividades. Desse modo, foi elaborada uma tabela (Tabela 4) para avaliar a aderência de cada atividade e a mortalidade de cada grupo. A mortalidade é calculada com base em cada grupo individualmente. Por exemplo, na atividade 7, ocorreram 2580 casos aderentes e 504 mortes dentro desse grupo, a mortalidade, portanto é 504 dividido por 2580. Nessa tabela não foram excluídas 2 atividades, alta do hospital e declaração de óbito, pois são atividades de desfecho do processo e teriam mortalidade de 0% e 100%, respectivamente, não agregando valor para análise. A seguir, a tabela (Tabela 4) de aderência por atividade:

Tabela 4 - Aderência por atividade.

	Atividade	Aderência	Qtd. Casos c/Atividade	Percentual de mortes	Qtd Mortes	Não Aderência	Qtd. Casos s/Atividade	Percentual de mortes não aderentes	Qtd Mortes não aderentes
1	Register the admission datetime of the given emergency department stay	100,00%	3358	20,28%	681	0,00%	0	0,00%	0
2	Register the date and time at which the patient was registered in the emergency department	100,00%	3358	20,28%	681	0,00%	0	0,00%	0
3	Register The date and time at which the patient was discharged from the emergency department	100,00%	3358	20,28%	681	0,00%	0	0,00%	0
4	Register the discharge datetime of the given emergency department stay	100,00%	3358	20,28%	681	0,00%	0	0,00%	0
5	Register the datetime of the Sepsis diagnosis	100,00%	3358	20,28%	681	0,00%	0	0,00%	0
6	Measure the patient`s vital signs.	100,00%	3358	20,28%	681	0,00%	0	0,00%	0
7	Administer medication Antibiotic via Pyxis in the emergency	76,83%	2580	19,53%	504	23,17%	778	22,75%	177
8	Record laboratory measurement/observation Lactate	84,81%	2848	22,40%	638	15,19%	510	8,43%	43
9	Record the date and time the patient was transferred from the ICU.	63,16%	2121	9,85%	209	36,84%	1237	38,16%	472
10	Record the date and time the patient was transferred to the ICU.	81,03%	2721	23,81%	648	18,97%	637	5,18%	33
11	Record the prescribed time for the medication Vasopressors	54,62%	1834	29,88%	548	45,38%	1524	8,73%	133
12	Record the prescribed time for the medication Antibiotic	97,59%	3277	19,56%	641	2,41%	81	49,38%	40
13	Record the prescribed time for the medication Fluids	59,80%	2008	20,52%	412	40,20%	1350	19,93%	269
14	Record the date and time of the routine Blood Culture	67,09%	2253	20,24%	456	32,91%	1105	20,36%	225
15	Record the prescribed time for the medication Corticosteroids	22,01%	739	32,61%	241	77,99%	2619	16,80%	440

Fonte: Elaboração própria.

Ao observar essa tabela, é possível perceber que há atividades inerentes ao processo possuindo 100% de aderência, as de 1 a 6. Dentre as atividades que estão presentes no protocolo, as mais seguidas são: administração de antibióticos, medição/observação laboratorial do lactato, transferência do paciente para UTI e administração de antibiótico via Pyxis. De maneira geral, as 9 recomendações principais, de 7 a 15 da Tabela 4, estão sendo seguidas de maneira aceitável. Em alguns casos, a não aderência possui uma menor mortalidade. Isso é explicado, possivelmente, pelo fato de que, se um paciente está levemente enfermo, não é necessário realizar determinadas atividades. Tem-se em destaque a atividade 12 de medicação de antibióticos na UTI que possui a maior taxa de aderência de atividades não inerentes e a maior taxa de mortalidade do grupo não aderente à atividade. Isso demonstra que essa recomendação é, de fato, relevante por diminuir, significativamente, a mortalidade.

4.3 Análise de desvios

A tabela com todos os desvios permitidos (Tabela 5) encontra-se nesta subseção como *allowlist*. Os desvios com *"undesired activity"* foram permitidos já que, com o auxílio do

especialista da área médica, foram consideradas apenas as atividades relevantes presentes no MIMIC-IV. Foram considerados desvios permitidos os casos com um percentual acima de 5% e atividades de procedimentos médicos que a ferramenta determinou como indesejadas, mas são atividades complementares ao BPMN padrão.

Tabela 5 -Tabela de desvios permitidos

	Lista de desvio permitidos	K1	K2	Total	Percentual de casos totais
1	Record the procedure Invasive Ventilation is an undesired activity	543	516	1059	31,54%
2	Record the procedure Blood Cultured is an undesired activity	401	404	805	23,97%
3	Record the procedure Dialysis Catheter is an undesired activity	125	135	260	7,74%
4	Record the procedure Stool Culture is an undesired activity	100	75	175	5,21%
5	Record the procedure Pan Culture is an undesired activity	37	30	67	2,00%
6	Record the procedure BAL Fluid Culture is an undesired activity	22	17	39	1,16%
7	Record the procedure Wound Culture is an undesired activity	19	15	34	1,01%
8	Record the procedure Sputum Culture is an undesired activity	165	144	309	9,20%
9	Record the procedure Urine Culture is an undesired activity	187	216	403	12,00%
10	Administer medication Corticosteroids via Pyxis in the emergency is an undesired activity	146	167	313	9,32%
11	Administer medication Vasopressors via Pyxis in the emergency is an undesired activity	106	98	204	6,08%
12	Record the procedure Peritoneal Dialysis is an undesired activity	4	7	11	0,33%
13	Record the procedure CSF Culture is an undesired activity	2	0	2	0,06%
14	Measure the patient's vital signs. is followed by Record laboratory measurement/observation Lactate	301	317	618	18,40%
15	Register The date and time at which the patient was discharged from the emergency department is followed by Record the prescribed the time for the medication Antibiotic	292	327	619	18,43%
16	Record the prescribed the time for the medication Antibiotic is followed by Register the date and time the patient was discharged from the hospital	283	286	569	16,94%
17	Record laboratory measurement/observation Lactate is followed by Record the prescribed the time for the medication Vasopressors	282	272	554	16,50%
18	Record the prescribed the time for the medication Antibiotic is followed by Record the date and time of the routine Blood Culture	277	314	591	17,60%
19	Record the date and time the patient was transferred to the ICU is followed by Record laboratory measurement/observation Lactate	260	275	535	15,93%
20	Record the date and time the patient was transferred to the ICU is followed by Record the prescribed the time for the medication Antibiotic	254	248	502	14,95%
21	Record the prescribed the time for the medication Antibiotic is followed by Record laboratory measurement/observation Lactate	223	231	454	13,52%
22	Record the prescribed the time for the medication Antibiotic is followed by Record the prescribed the time for the medication Vasopressors	223	174	397	11,82%
23	Record laboratory measurement/observation Lactate is followed by Record the prescribed the time for the medication Antibiotic	216	206	422	12,57%
24	Record laboratory measurement/observation Lactate is followed by Record the prescribed the time for the medication Fluids	210	198	408	12,15%
25	Record the prescribed the time for the medication Antibiotic is followed by Record the prescribed the time for the medication Fluids	205	221	426	12,69%
26	Record the prescribed the time for the medication Fluids is followed by Record the prescribed the time for the medication Antibiotic	178	187	365	10,87%
27	Record the prescribed the time for the medication Antibiotic is followed by Record the date and time the patient was transferred from the ICU.	160	183	343	10,21%
28	Record the date and time the patient was transferred to the ICU is followed by Record the prescribed the time for the medication Fluids	121	125	246	7,33%
29	Register The date and time at which the patient was discharged from the emergency department is followed by Record the prescribed the time for the medication Fluids	114	121	235	7,00%
30	Record the date and time of the routine Blood Culture is followed by Record laboratory measurement/observation Lactate	110	108	218	6,49%
31	Record the prescribed the time for the medication Fluids is followed by Record the prescribed the time for the medication Vasopressors	95	112	207	6,16%
32	Record the date and time the patient was transferred from the ICU is followed by Record the prescribed the time for the medication Fluids	91	74	165	4,91%

Fonte: Elaboração própria.

Como a análise de aderência já traz *insights* relevantes para entender o processo de tratamento da sepse, uma análise importante a ser feita é a de desvios. Em pacientes que ocorreram determinado desvio, a mortalidade foi maior ou menor? Essa parte do texto possui o

intuito de responder tal questionamento. Sendo assim, foram selecionados os desvios mais notáveis, isto é, os que ocorreram acima de 5% dos casos e que não eram considerados como desvios permitidos por questão de ordem. Desses desvios, obtivemos os seguintes resultados:

- Desvio 1: *Measure the patient's vital signs is directly followed by Record the prescribed the time for the medication Antibiotic.*
  - Por que é um desvio: A primeira atividade ocorre na emergência e a segunda na UTI. Entre essas atividades, deveria ter a alta na emergência e, posteriormente, a entrada na UTI.
  - Mortalidade: 8,86%
- Desvio 2: *Record the date and time the patient was transferred from the ICU is directly followed by Record the date and time of the routine Blood Culture.*
  - Por que é um desvio: Após a saída da UTI, nenhuma atividade deve ser realizada. Para uma nova atividade ser realizada deve ser aberta uma nova internação.
  - Mortalidade: 5,79%
- Desvio 3: *Record the prescribed the time for the medication Antibiotic is directly followed by Record the date and time the patient was transferred to the ICU.*
  - Por que é um desvio: A primeira atividade desse desvio ocorre na UTI, isto é, ele não poderia realizá-la sem antes ter sido internado. É importante ressaltar, que as atividades com "via Pyxis" são feitas, exclusivamente, na emergência.
  - Mortalidade: 29,87%

Em relação aos dados desses casos não aderentes, pode-se perceber que o desvio 3 possui uma maior mortalidade. Isso pode ser explicado que, talvez, a atividade "*Record the prescribed the time for the medication Antibiotic*" é estritamente realizada na UTI.

4.4 Análise de aderência do tempo dos processos

Por fim, realizamos a análise de aderência do tempo dos processos. Essa análise foi realizada para permitir a explicação dos tempos padrões na metodologia e a comparação dos tempos dos processos em relação aos tempos padrões. Com base na ferramenta "Throughput time selection" do Celonis, foi possível identificar processos que excederam os tempos padrões estabelecidos, indicando possíveis gargalos e oportunidades de otimização para reduzir os atrasos e melhorar a eficiência do atendimento. Das 17 atividades analisadas, 5 recomendações

dependem de tempo e foram extraídas de Evans et al. (2021) e Kalimouttou et al. (2023). São elas:

- Para adultos com possível sepse sem choque, sugerimos um curso de investigação rápida por tempo limitado e, se a preocupação de infecção persistir, a administração de antimicrobianos dentro de 3 horas a partir do momento em que a sepse foi reconhecida pela primeira vez.
- Para adultos com possível choque séptico ou alta probabilidade de sepse, recomendamos a administração de antimicrobianos imediatamente, de preferência dentro de uma hora após o reconhecimento.
- Para adultos com sepse ou choque séptico que requerem internação na UTI, sugerimos a admissão dos pacientes na UTI em até 6 horas.
- Para adultos com suspeita de sepse, sugerimos medir o lactato sanguíneo na primeira hora.
- Para adultos com suspeita de sepse ou choque séptico, recomendamos a realização da hemocultura em até uma hora.

As duas primeiras recomendações sugerem que o tempo para administrar antimicrobianos/antibióticos deve ser dentro de 1 e aceitável 3 horas após o diagnóstico. Como citado anteriormente, no capítulo de metodologia, subseção “Geração do log”, foi anexado, manualmente, um registro de data e hora à tabela “*diagnoses_icd*” e foi considerado que o momento em que é feito o diagnóstico é quando o paciente entra na emergência.

Das 3358 internações e seguindo as recomendações de Evans et al. (2021) e Kalimouttou et al. (2023), foram observados os intervalos de tempo entre processos e a taxa de mortalidade daqueles que estão dentro do intervalo. Com 53,48% dos casos dentro da recomendação da administração de antimicrobianos (Figura 12):

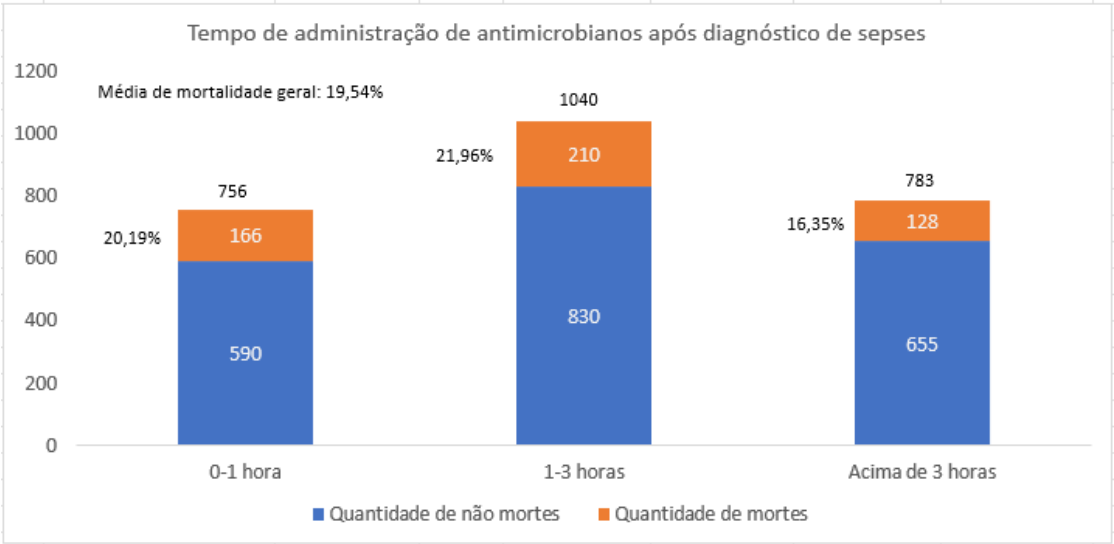

Figura 12 - Tempo de administração de antimicrobianos após diagnóstico de sepses.
Fonte: Baseado nos dados do MIMIC-IV.

Dos casos que performaram a administração de antimicrobianos dentro de 1 hora após o diagnóstico de sepsis houve taxa de mortalidade de 21,96%. Dos 756 casos dentro dessa recomendação houve 166 mortes.

Dos casos que performaram a administração de antimicrobianos dentro de 3 horas após o diagnóstico de sepsis houve taxa de mortalidade de 20,94%. Dos 1796 casos dentro do nível de aceitação houve 376 mortes.

Dos casos que performaram a administração de antimicrobianos depois de 3 horas após o diagnóstico de sepsis houve taxa de mortalidade de 16,35%. Dos 783 casos fora da recomendação houve 128 mortes.

Com 40,44% dos casos dentro da recomendação de transferência para UTI (Figura 13):

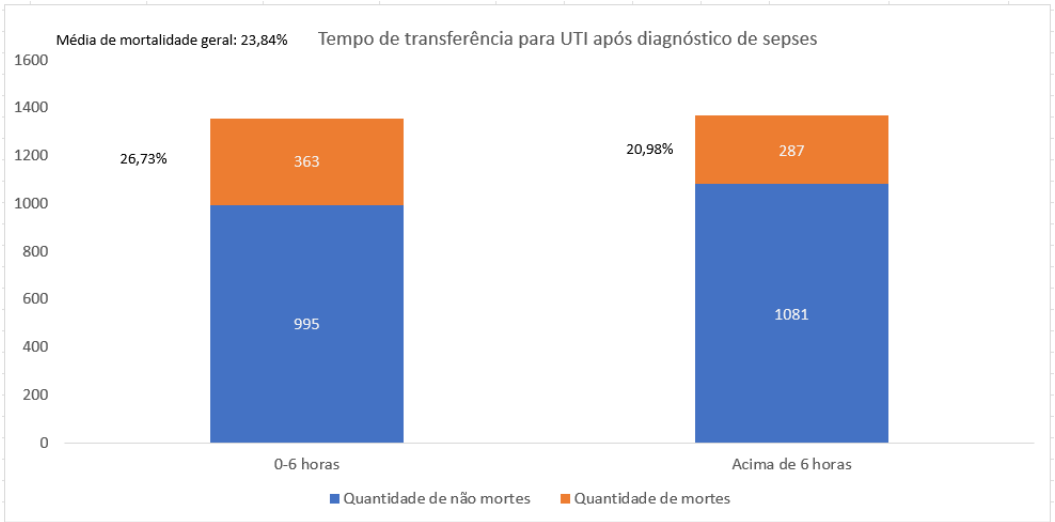

Figura 13 - Tempo de transferência para UTI após diagnóstico de sepses.
Fonte: Baseado nos dados do MIMIC-IV.

Dos casos que transferiram o paciente para UTI dentro de 6 horas após o diagnóstico de sepsis houve taxa de mortalidade de 26,73%. Dos 1358 casos dentro da recomendação houve 363 mortes.

Dos casos que transferiram o paciente para UTI fora de 6 horas após o diagnóstico de sepsis houve taxa de mortalidade de 20,98%. Dos 1368 casos fora da recomendação houve 287 mortes.

Com 8,64% dos casos dentro da recomendação de medição do lactato (Figura 14):

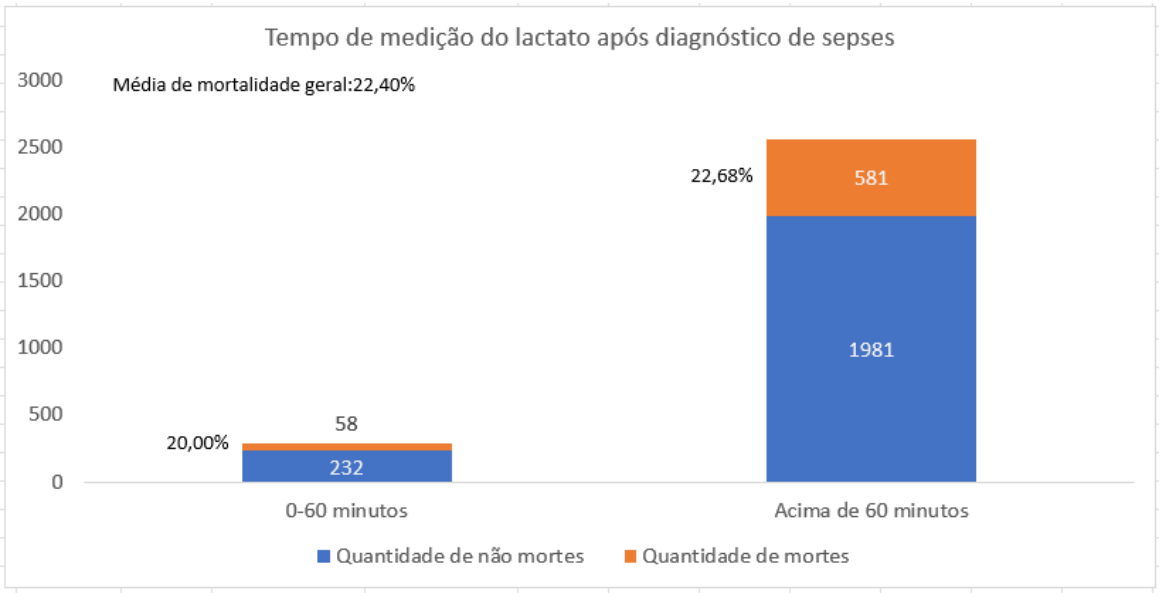

Figura 14 - Tempo de medição de lactato após diagnóstico de sepses.
Fonte: Baseado nos dados do MIMIC-IV.

Dos casos que mediram o lactato dentro de 1 hora após o diagnóstico de sepsis houve taxa de mortalidade de 20,00%. Dos 290 casos dentro da recomendação houve 58 mortes.

Dos casos que mediram o lactato após 1 hora do diagnóstico de sepsis houve taxa de mortalidade de 22,68%. Dos 2562 casos fora da recomendação houve 581 mortes.

Com isso, pode-se perceber que, nem sempre, atender à especificação de tempo, necessariamente, reduz a taxa de mortalidade.

5 Discussão / Considerações Finais

Um artigo com ideia semelhante a este, mas com proposta diferente é o Kalimouttou et al. (2023). Nele, a conformidade com as diretrizes da *Surviving Sepsis Campaign* (SSC) foi analisada a fim de encontrar as recomendações que minimizem a mortalidade em 28 dias. Das 79 recomendações incluídas na última versão do SSC, na época, 22 foram avaliáveis no MIMIC-IV. Contudo, as 22 recomendações do artigo, de maneira geral, são diferentes das presentes neste estudo. Dessas, 5 recomendações são mútuas em relação ao presente trabalho. Estas são as recomendações 1, 2, 3, 5 e 21 do artigo de Kalimouttou et al. (2023). A taxa geral de adesão a essas 22 recomendações foi de 77,2%. No caso deste estudo, foi de 34,69%. Contudo, os métodos foram diferentes. No caso do artigo, foi analisada a adesão a cada recomendação/atividade, assim como neste estudo. Entretanto, foi feita uma média entre todos os 22 *guidelines*, obtendo o valor de 77,2%. No caso deste trabalho, a conformidade de 34,69% é referente a seguir todas as recomendações relevantes ao mesmo tempo, pelo menos. A análise de adesão a cada recomendação/atividade foi analisada no capítulo de Resultados, mas, foi obtido o valor de 66,11% para a adesão média, considerando as seguintes atividades: antibiótico (emergência e UTI), lactato, vasopressor (UTI), fluidos (UTI), hemocultura de rotina e corticosteroides (UTI). A variabilidade da conformidade do artigo foi de 15,1% até 99,9%, enquanto o deste trabalho foi de 22,01% até 97,59%.

Das limitações presentes, duas são as mais relevantes: SOFA e choque séptico. De acordo com Singer et al. (2016), além do critério de qSOFA, o SOFA deve ser utilizado para verificar se um paciente possui sepse. Entretanto, o SOFA possui 7 variáveis usadas para seu cálculo: PaO2/FiO2, plaquetas, bilirrubina, MAP (*Mean Arterial Pressure*), *Glasgow Coma Scale score*, creatinina e urina. Nem todos os pacientes possuem essas medidas. Para verificar isso, duas tabelas foram analisadas: "*icu/chartevents*" e "*hosp/labevents*". Na primeira, nenhum paciente apresentou PaO2/FiO2 ou *Glasgow Coma Scale score*. A seguir, há uma tabela (Tabela 6) de quantas internações cada uma das outras medidas estão faltando e seu respectivo percentual de falta no total:

Tabela 6 - Medições faltantes para cada variável em icu/chartevents.

Referencial	missing_hadm_id_count	Percentual (%)
Bilirrubina	590	22,0396
Creatinina	8	0,2988
Plaquetas	13	0,4856
Pressão Arterial	1833	68,4722

Fonte: Elaboração própria.

Na segunda, nenhum apresentou PaO2/FiO2 ou Glasgow Coma Scale score ou MAP (Mean Arterial Pressure). O mesmo modelo de tabela (Tabela 7) anterior foi feito para essa:

Tabela 7 - Medições faltantes para cada variável em hosp/labevents.

Referencial	missing_hadm_id_count	Percentual (%)
Bilirrubina	1509	43,3971
Creatinina	6	0,1805
Plaquetas	5	0,1504

Fonte: Elaboração própria.

Em uma versão anterior do MIMIC-IV, a 1.0, havia uma tabela presente no módulo "*derived*" chamada "*sofa*", que já servia para obter o SOFA Score. Isso pode ser constatado no Conteúdo Suplementar de Kalimouttou et al. (2023). Por conseguinte, como essa tabela não existe mais, o cálculo do SOFA torna-se ainda mais complexo no MIMIC-IV.

Outra limitação importante, também por falta de dados, é a verificação de choque séptico. Ainda segundo Singer et al. (2016), além dos cálculos de qSOFA e SOFA, é importante verificar se os pacientes possuem choque séptico, de acordo com dois critérios:

- Necessidade de vasopressores manterem MAP ≥ 65 mm Hg
- Nível sérico de lactato > 2 mmol/L

De maneira geral, o processo de identificação dos pacientes segue o seguinte formato (Figura 15):

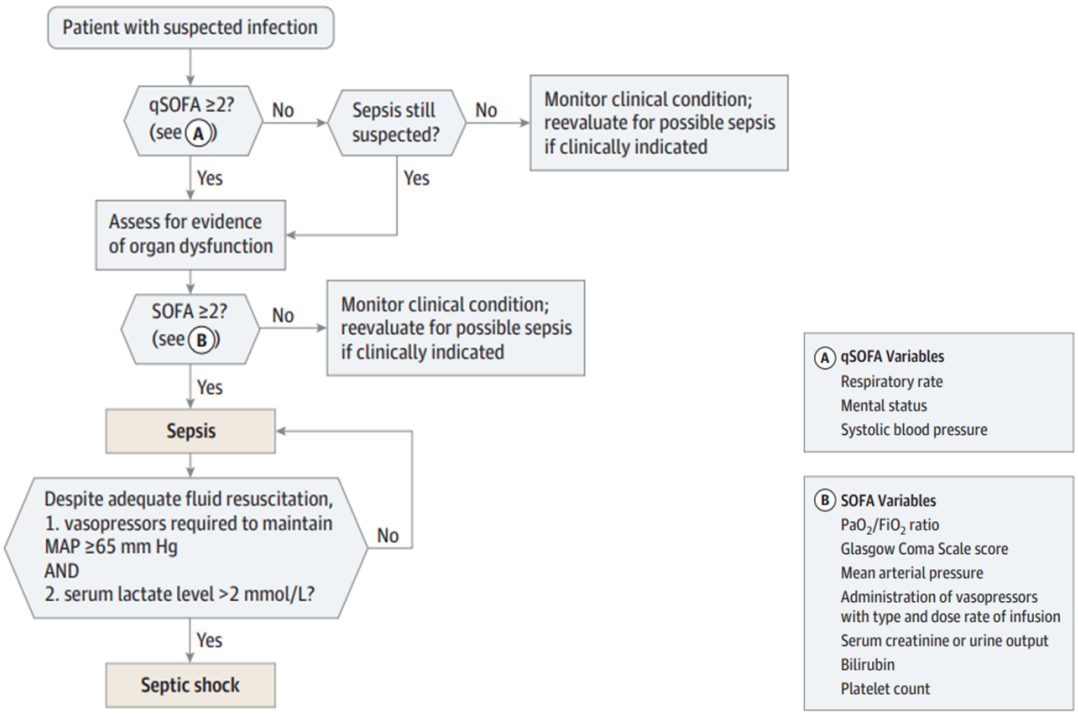

Figura 15 - Operacionalização de Critérios Clínicos. Identificadores de Pacientes com Sepse e Choque Séptico. Fonte: Singer et al. (2016).

Assim como discutido na limitação do SOFA, pelo fato de nem todos os pacientes possuírem as variáveis relevantes para os critérios, o filtro de choque séptico não foi possível de ser realizado.

Além disso, assim como citado em Kalimouttou et al. (2023), a base MIMIC-IV só reflete as práticas de um hospital específico. Dados observacionais estão sujeitos a vieses, principalmente se provenientes de um único centro. As informações contidas nestes dados sobre a conformidade com as diretrizes de sepse são apenas referentes às práticas locais. Ainda sobre as recomendações, é importante ressaltar que o tratamento da sepse deve ser seguir o maior número possível de recomendações, especialmente as com fortes evidências.

6 Conclusão

A sepse representa um grave problema de saúde, uma vez que nos Estados Unidos é uma das enfermidades mais onerosas para se tratar. Ainda mais, a incidência de sepse tem se ampliado, provavelmente devido ao aumento na idade média da população que apresenta mais comorbidades. Tem se notado, também, de forma crescente, que sujeitos que superam a sepse frequentemente lidam com fraquezas físicas, emocionais e cognitivas prolongadas, trazendo grandes impactos para o sistema de saúde e o âmbito social. No Brasil, o total de pacientes diagnosticados com sepse e choque séptico de acordo com o ILAS (Instituto Latino Americano de Sepse) de 2005 a 2022 atinge 134.532. Em contraste, somente no ano de 2021, foram registrados 14.366 casos, correspondendo a mais de 10% de todos os casos durante este período. Isso evidencia uma quantidade considerável de ocorrências ao longo dos anos, sugerindo que a sepse e o choque séptico permanecem como uma significativa questão de saúde no Brasil.

Nos últimos anos, ocorreu uma mobilização coletiva para a implementação de sistemas eletrônicos de documentação de saúde em instituições hospitalares. Nos Estados Unidos, em 2021, houve um aumento considerável na utilização de Sistemas Eletrônicos de Registros de Saúde (EHR) certificados. Nesta perspectiva, informações médicas obtidas por meio de levantamentos retrospectivos têm sido progressivamente empregadas em estudos epidemiológicos e em modelagem de previsão, em parte devido à efetividade das estratégias de modelagem em grandes volumes de dados. Contudo, o acesso aos dados médicos para aprimorar o tratamento de um paciente ainda é um desafio, muito por causa do seu acesso e disponibilização, sendo a privacidade do paciente uma das principais preocupações.

É nesse sentido que o presente trabalho busca usar uma base de dados médica, o MIMIC-IV, para aplicar técnicas de *process mining* e *conformance checking* em relação ao tratamento da sepse.

Na base do MIMIC-IV, foram utilizadas diversas técnicas com a linguagem de programação Python para tratar os dados a fim de gerar um *log* de eventos. Ademais, foi necessário, realizar filtros prévios em relação à coorte de pacientes, aos remédios e as tabelas que seriam relevantes a serem analisadas. Uma dificuldade encontrada foi o tamanho extremamente grande de algumas tabelas. Para resolver isso, foi preciso particionar alguns dados para facilitar o seu tratamento e manuseio.

Após gerado o *log* de eventos, a etapa de construir o BPMN padrão para o tratamento da sepse foi realizada. Para isso, foram consideradas 3 fontes, além do auxílio do especialista da área médica. Depois de chegado a um consenso de quais recomendações seriam relevantes a serem analisadas, o BPMN foi gerado por meio da ferramenta Celonis. Apesar disso, foi necessário adicionar algumas atividades para tornar o BPMN completo, sendo elas: lactato, hemocultura e corticosteroides.

Com o BPMN padrão feito, a etapa de resultados foi iniciada. Dessa parte, os principais achados estão relacionados a, primeiramente, uma menor mortalidade do grupo conforme ao BPMN padrão. O grupo conforme teve 9,10% de mortalidade, enquanto o não conforme 26,22%. Além disso, sobre a análise por processo, a atividade que teve maior destaque foi a de administração de antibióticos na UTI, com aderência de 97,59% e mortalidade desse grupo de 19,06%. Os não aderentes tiveram 49,38% de mortalidade. Em relação a análise de desvios, o mais impactante foi a atividade de administrar antibiótico na UTI no momento errado, em que teve uma mortalidade associada de 29,87%. Finalmente, a análise de aderência do tempo dos processos demonstrou que não necessariamente adequar-se a esses tempos resulta em uma menor mortalidade.

O presente estudo foi realizado como uma das iniciativas do Núcleo de Operações e Inteligência em Saúde (NOIS - www.nois.ind.puc-rio.br), um laboratório de Pesquisa, Desenvolvimento e Inovação (P, D & I) do Departamento de Engenharia Industrial da PUC-Rio, cujo objetivo é incorporar técnicas de Ciência de Dados e Otimização de recursos para endereçar problemas na área da Saúde.

Em relação a trabalhos futuros, assim como realizado em Cremerius et al. (2023), seria possível criar um extrator de log baseado no MIMIC-IV. Ao seguir a metodologia do presente trabalho, pode-se obter um log com os mesmos atributos. Entretanto, um trabalho futuro seria realizar uma ferramenta de fácil uso capaz de gerar um log, mas baseado nesta metodologia, assim tornando o trabalho de estudar outros diagnósticos mais acessível. Essa ferramenta levaria em consideração quais tabelas o usuário possui interesse, quais campos e quais diagnósticos específicos quer filtrar. Ao final, o log gerado possuiria as seguintes colunas: "*subject_id*", "*hadm_id*", "*stay_id*", "*stay_id_icu*", "*Activity*", "*Column*", "*Resource*", "*Init Timestamp*", "*End Timestamp*", "*Add. Info*"; diferente do que é feito no artigo citado anteriormente. Sendo necessário, somente, que o usuário baixe previamente a base e a descompacte.

Além disso, como mencionado na parte final de limitações, o uso desta metodologia ou semelhante em outras bases e em outros países traria outras perspectivas para trabalhos futuros. Assim como realizado em Cremerius et al. (2023), foi demonstrado o uso do MIMIC-IV no processo de tratamento de insuficiência cardíaca. Esse processo, assim como o da sepse, também possui recomendações a serem seguidas. O artigo citado poderia ter realizado uma análise de conformidade acerca dos *guidelines* dessa doença. Portanto, o uso de diferentes critérios e ferramentas leva a outros trabalhos possíveis que auxiliariam nas suas respectivas áreas e, principalmente, no tratamento de doenças.

Referências Bibliográficas

BIBLIOTECA VIRTUAL EM SAÚDE. 13/9 - DIA Mundial da Sepse. Disponível em: https://bvsms.saude.gov.br/13-9-dia-mundial-da-sepse-3/#:~:text=Sepse%2C%20conhecida%20como%20infec%C3%A7%C3%A3o%20generalizada,formas%20graves%20de%20Covid%2D19. Acesso em: 14 maio 2023.

CREMERIUS, J. et al.. Event log generation in MIMIC-IV research paper. Doi:10.1007/978-3-031-27815-0_22. *Scopus*. 2023. Disponível em: www.scopus.com. Acesso em: 10 jan. 2023.

EUA. The Office of the National Coordinator for Health Information Technology (ONC). National Trends in Hospital and Physician Adoption of Electronic Health Records. *Health It*. 2022. Disponível em: https://www.healthit.gov/data/quickstats/national-trends-hospital-and-physician-adoption-electronic-health-records#:~:text=As%20of%202021%2C%20nearly%204,physicians%20had%20adopted%20an%20EHR. Acesso em: 5 maio 2023.

EVANS, L. et al.. Surviving sepsis campaign: International guidelines for management of sepsis and septic shock. 2021. *Intensive Care Medicine*, 47(11), 1181-1247. Doi: 10.1007/s00134-021-06506-y.

FUCHS, Antonio. Sepse: a maior causa de morte nas UTIs. (2021). *Fiocruz*. Disponível em: https://portal.fiocruz.br/noticia/sepse-maior-causa-de-morte-nas-utis#:~:text=Nos%20Estados%20Unidos%2C%20a%20mortalidade,em%2020%25%20de%20mortalidade%20hospitalar. Acesso em: 04 jun. 2023.

GOLDBERGER, AL. et al.. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation* 101(23):e215-e220, 13 jun. 2000. Disponível em: http://circ.ahajournals.org/content/101/23/e215.full. Acesso em: 2 jan. 2023.

JANS, M.; SOFFER, P.; JOUCK, T. Building a valuable event log for process mining: An experimental exploration of a guided process. Enterprise Information Systems, 13(5), 601-630, 2019. Doi:10.1080/17517575.2019.1587788.

JOHNSON, A. et al.. MIMIC-IV (version 2.2). PhysioNet, 2023. https://doi.org/10.13026/6mm1-ek67.

JOHNSON, A.E.W. et al.. MIMIC-IV, a freely accessible electronic health record dataset. Sci Data 10, 1 (2023). https://doi.org/10.1038/s41597-022-01899-x

KALIMOUTTOU, A. et al.. Machine-learning-derived sepsis bundle of care. *Intensive Care Med*, jan. 2023, 49(1):26-36. Doi: 10.1007/s00134-022-06928-2. Epub 29 nov. 2022. PMID: 36446854.

KUSUMA, G. et al.. Process mining of disease trajectories in mimic-iii: A case study. In: *Process Mining Workshops*, p. 305-316, Springer, 2020.

LIANG, L.; MOORE, B.; SONI, A. National Inpatient Hospital Costs: The Most Expensive Conditions by Payer, 2017. 2020 Jul 14. In: *Healthcare Cost and Utilization Project (HCUP) Statistical Briefs [Internet]*. Rockville (MD): Agency for Healthcare Research and Quality (US); 2006 Feb. Statistical Brief #261. Disponível em: https://www.ncbi.nlm.nih.gov/books/NBK561141/. Acesso em: 5 maio 2023.

OBJECT MANAGEMENT. GROUP. *Graphical Notations for Business Processes*. Disponível em: omg.org/bpmn/index.htm. Acesso em: 25 maio 2023.

QUINTANO, Ricardo. *A multi-criteria process mining optimization tool and its application in a sepsis clinical pathway*.

RISTAD, E. S.; N.YIANILOS, P. *Learning string-edit distance*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(5), 522-532, 1998. Doi:10.1109/34.682181.

SINGER, M. et al.. The Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3). *JAMA*, 2016, 315(8):801-810. Doi:10.1001/jama.2016.0287.

VAN DER AALST, W. M. P. Process mining: A 360 degree overview. Doi:10.1007/978-3-031-08848-3_1. *Scopus*. 2022. Disponível em: www.scopus.com. Acesso em: 10 dez. 2022.

VOGELGESANG, T. et al.. Celonis PQL: A Query Language for Process Mining. In: POLYVYANYY, A. (eds.). *Process Querying Methods*. Springer, 2022. Doi:https://doi.org/10.1007/978-3-030-92875-9_13.

APÊNDICE A

Conceitos importantes

A fim de melhorar a experiência do leitor, este apêndice possui o objetivo de explicitar, previamente, conceitos importantes para o melhor entendimento do processo. Tais definições são:

- UTI: é a sigla para Unidade de Terapia Intensiva, que é um setor especializado de um hospital dedicado ao atendimento de pacientes que estão em situações críticas ou de alta complexidade. Os pacientes em UTIs requerem monitoramento e cuidados intensivos 24 horas por dia por uma equipe multidisciplinar, que pode incluir médicos intensivistas, enfermeiros, fisioterapeutas e outros profissionais de saúde. A sigla em inglês é "ICU" como "*Intensive Care Unit*".
- Pyxis: é um sistema de dispensação de medicamentos automatizado desenvolvido pela empresa de tecnologia médica CareFusion, que agora faz parte da BD (Becton, Dickinson and Company). O sistema Pyxis é projetado para aumentar a segurança do paciente e a eficiência no gerenciamento de medicamentos em hospitais e outras instalações de saúde. Ele funciona como uma espécie de máquina de venda automática para medicamentos, com controles de acesso para garantir que apenas profissionais de saúde autorizados possam dispensar os medicamentos.
- Github: uma plataforma de hospedagem de código para controle de versão e colaboração, permitindo que pessoas trabalhem juntas em projetos de qualquer lugar, por meio, principalmente, de alterações e adição de novos recursos. É possível visualizar o que outras pessoas fizeram e, caso algo dê errado, é capaz de voltar ao estado que o projeto era antes. O link oficial do github é: https://github.com/.
- No caso deste projeto, para obter os códigos e tabelas usadas para a análise, acesse o link: https://github.com/gabrielgoncalvess/TCC-MIMIC-IV.
- Celonis: é líder global em gerenciamento de execução. A empresa é conhecida por seu software de mineração de processos de negócios, que utiliza a inteligência artificial e outras tecnologias para ajudar as organizações a visualizar e melhorar seus processos comerciais. O software Celonis é capaz de capturar dados de uma ampla variedade de sistemas operacionais e de gerenciamento de negócios (como ERPs, CRMs, SRMs, entre outros) para fornecer uma visualização detalhada dos processos

de negócios. Ele identifica gargalos e ineficiências, permitindo que as organizações melhorem seus processos.

- Python: uma linguagem de programação interpretada, orientada a objetos e de alto nível com semântica dinâmica. Suas estruturas de dados embutidas e a tipagem e vinculação dinâmicas tornam o Python ideal para desenvolvimento rápido de aplicações e como linguagem de script. Sua sintaxe simples enfatiza a legibilidade, reduzindo os custos de manutenção do programa. Python suporta módulos e pacotes, promovendo a modularidade e reutilização do código. O interpretador Python e sua extensa biblioteca padrão estão disponíveis gratuitamente para todas as principais plataformas. Para mais informações, acesse o link oficial: https://www.python.org/.
- Função: em Python, uma função é um bloco de código organizado e reutilizável que é usado para realizar uma única ação relacionada. As funções proporcionam melhor modularidade para o aplicativo e um alto grau de reutilização de código. Por exemplo, uma função que eleva um número ao quadrado. Como parâmetro terá o número e retornará esse número elevado ao quadrado. Em código seria: “numero ** 2”. Mais informações estão disponíveis na documentação oficial do Python: https://docs.python.org/3/.
- String: uma sequência de caracteres - pode ser uma palavra, uma frase, ou até mesmo um parágrafo inteiro. Um exemplo de string é: “Olá, mundo!”. A sequência de caracteres se inicia com aspas simples ou duplas.
- OS: uma biblioteca padrão do Python, que proporciona uma maneira fácil de usar funcionalidades que são dependentes do sistema operacional. Exemplos de aplicações são: navegar pelo sistema de arquivos e executar comandos do sistema operacional.
- Dask: uma biblioteca de código aberto para a linguagem de programação Python que permite computação paralela e distribuída. Ela é comumente usada para trabalhar com grandes conjuntos de dados que não cabem na memória de um único computador. Ou seja, é uma ferramenta poderosa quando é necessário realizar cálculos em grandes conjuntos de dados e com o intuito de aproveitar a capacidade de processamento paralelo e distribuído. Para mais informações: https://www.dask.org/powered-by.
- Pandas: uma ferramenta/biblioteca de análise e manipulação de dados de código aberto rápida, poderosa, flexível e fácil de usar, construída sobre a linguagem de

programação Python. Esse recurso baseia-se, principalmente, nas estruturas de dados *DataFrame* e Series, que, de maneira resumida, são, respectivamente, como uma tabela Excel e uma única coluna. Para obter informações mais detalhadas sobre essa biblioteca há o link: https://pandas.pydata.org/. Essa é a principal ferramenta utilizada nesse trabalho e, dentro dela, existem alguns métodos que foram importantes para o tratamento adequado dos dados, que são:

- Merge: de maneira resumida, permite combinar duas tabelas diferentes com base em uma coluna comum. Por exemplo, duas tabelas de informações - uma lista de amigos e seus endereços de e-mail, e outra lista de amigos e seus números de telefone. Com o método merge, é possível combinar duas tabelas diferentes com base em uma coluna comum. Nesse caso, a coluna comum seria o nome do amigo, porque está presente em ambas as tabelas. O método permite criar uma única tabela que tem nomes de amigos, e-mails e números de telefone, tudo junto. Para mais informações e exemplos: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.merge.html.
- Apply: aplica uma função ao longo de um eixo do *DataFrame*. Cada item na coluna ou item é modificado por essa função. Seguindo o mesmo exemplo do tópico de função, com o apply é possível fazer com que cada número em uma coluna seja convertido para o seu valor elevado ao quadrado. Para mais informações e exemplos acerca desse método, há o link a seguir: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.apply.html.

APÊNDICE B

Processo de estratificação de gravidade e identificação da população de interesse.

Para obter os diagnósticos de pacientes, foi utilizada, principalmente, a tabela "*hosp/diagnoses_icd".* Inicialmente, foi feita a importação dessa tabela e a sua dimensão, "*hosp/d_icd_diagnoses"*, que fornece uma descrição mais precisa sobre o diagnóstico. Um "*merge"* foi realizado entre essas duas tabelas, utilizando a coluna "*icd_code*", a fim de obter uma tabela de diagnósticos mais detalhada. Em seguida, o propósito era identificar, nessa base, o diagnóstico de sepse. Para isso, criamos uma nova coluna chamada "*long_title2*", que é baseada na "*long_title*", ou seja, a descrição do diagnóstico. A "*long_title2*" é igual a "*long_title*" mas todo o seu texto está em minúsculo, por meio do método *"str.lower()".* Isso foi feito para facilitar a identificação da sepse. Por fim, o texto "seps" foi procurado dentro dessa nova coluna, por meio de um filtro e com o método "*str.contains('seps')*". Dessa forma, caso esse texto estivesse dentro do texto do diagnóstico completo, este seria considerado. No entanto, foi percebido que, para um paciente em uma mesma internação, este poderia ter mais de um diagnóstico. Para resolver essa questão, consideramos somente o último diagnóstico, por ser, teoricamente, o mais preciso. Para descartar os diagnósticos anteriores ao último, para cada paciente e internação, foi utilizado o método "*groupby"* do Pandas, com as colunas "*subject_id*" e "*hadm_id*" para agrupar. Logo em seguida, o método "*tail",* que considera apenas o último registro de cada grupo, foi utilizado. O número de pacientes com diagnóstico de sepse é de 12616 e existem 15558 internações. Essa tabela foi exportada, por meio do método "*to_excel"*, com o nome de "pacientes_diag_sepse.xlsx".

Além de pacientes com diagnóstico de sepse, foi necessário filtrar, dentre esses, quais possuem um qSOFA igual 2. Como primeiro passo, foi realizada a importação das tabelas relevantes para essa etapa utilizando a biblioteca Pandas.

Para esse filtro, foi necessário utilizar o método "*merge"* do pandas entre as tabelas "*patients"* e "*admissions"*, usando a coluna "*subject_id*". Com isso, conseguimos obter uma tabela com informações mais detalhadas sobre os pacientes e suas admissões. Com o intuito de obter somente os pacientes com diagnóstico de sepse na tabela anterior, foi realizado um "*merge"* entre esta e a "pacientes_diag_sepse.xlsx", usando as colunas "*subject_id"* e "*hadm_id"*.

Consideramos como importantes, a princípio, as colunas: "*subject_id"*, "*hadm_id"*, "*gender"*, "*anchor_age"*, "*dod"*, "*admittime"*, "*dischtime"*, "*deathtime"*, "*edregtime"*, "*edouttime"*. O método "*merge"* foi utilizado novamente entre a tabela resultante e "*edstays"*, usando as colunas "*subject_id*" e "*hadm_id*", e também com a tabela "*vitalsign",* com as colunas "*subject_id*" e "*stay_id*", a fim de obter dados mais detalhados acerca das estadias na emergência e sinais vitais dos pacientes. Após fundir esses *DataFrames*, a duração, em dias, do paciente no hospital foi calculada utilizando-se o método "*to_datetime"* do pandas e subtraindo as colunas "*dischtime*" e "*admittime*", o que originou a coluna "*LOS hosp*".

Para o cálculo do qSOFA, foram criadas duas colunas, "*pont_resprate*" e "*pont_sbp*", a fim de calcular o *score*. Por meio do uso do método "*apply",* foi criada uma função lambda, para, em cada linha, verificar os critérios e, em caso positivo, retornar o número 1, em caso contrário, retornar 0. Em seguida, foi utilizado o método "*groupby"*, agrupando pelo "*subject_id*" e "*hadm_id*", para agrupar a tabela de acordo com os pacientes e suas respectivas internações. Com as colunas "*pont_resprate*" e "*pont_sbp*", seus valores máximos foram calculados dentro do "*groupby*" utilizando o método "*max*". Desse modo, foi criada uma coluna "*pont_total*", somando os valores das colunas "*pont_resprate*" e "*pont_sbp*", a fim de obter o qSOFA. Por fim, pacientes com qSOFA menor que 2 foram descartados.