


UM *FRAMEWORK* PARA APOIAR ESPECIALISTAS NO PROBLEMA DE ESCALONAMENTO DE EXAMES HOSPITALARES**A FRAMEWORK TO SUPPORT SPECIALISTS IN THE PROBLEM OF HOSPITAL EXAMINATION SCALING****UN *FRAMEWORK* PARA APOYAR A LOS ESPECIALISTAS EN EL PROBLEMA DE LA ESCALA DE LOS EXÁMENES HOSPITALARIOS** <https://doi.org/10.56238/rcsv16n5-001>

Data de submissão: 05/04/2026

Data de aprovação: 05/05/2026

Francisco Levi Souza Anastácio

Graduando em Ciência da Computação
Instituição: Universidade Federal do Ceará
Endereço: Ceará, Brasil
E-mail: levianast70@alu.ufc.br
Orcid: <https://orcid.org/0009-0002-4002-5168>
Lattes: <http://lattes.cnpq.br/3999205523051331>

Mayrton Dias de Queiroz

Doutor em Ciência da Computação
Instituição: Universidade Federal do Ceará
Endereço: Ceará, Brasil
E-mail: mayrton@ufc.br
Orcid: <https://orcid.org/0000-0003-0116-5854>
Lattes: <http://lattes.cnpq.br/2061236974018450>

RESUMO

A gestão eficiente do fluxo de pacientes em ambientes hospitalares, especialmente no escalonamento de exames diagnósticos, representa um desafio crítico diante da elevada demanda e da limitação de recursos. O Problema de Escalonamento de Exames Hospitalares, consiste em determinar a sequência de processamento de exames em múltiplas estações, considerando prioridades clínicas e tempos de chegada. O objetivo deste trabalho consiste em identificar uma alternativa capaz de escalonar exames hospitalares, por meio de algoritmos heurísticos. A metodologia incluiu uma revisão sistemática da literatura para fundamentação teórica, seguida da implementação de três algoritmos de escalonamento: Arrival Order Priority (AOP), baseado na ordem de chegada; Minimum Processing Time Priority (MPTP), baseado no menor tempo de processamento; e Roulette Selection Strategy (RSS), de natureza probabilística. Para analisar as soluções geradas, foi desenvolvido um framework capaz de reconstruir dinamicamente o fluxo de processamento a partir dos dados produzidos. Os experimentos foram conduzidos em 150 cenários distintos, variando-se o número de exames e de estações de processamento. Os resultados indicaram que o AOP apresentou o melhor desempenho médio em termos de tempo total de processamento em todos os cenários analisados, enquanto o RSS obteve os maiores tempos médios, embora tenha ampliado a exploração do espaço de soluções. Conclui-se que a política de escalonamento impacta diretamente o desempenho operacional e que o framework proposto contribui para a análise, validação e integração de estratégias de apoio à decisão em ambientes hospitalares.

Palavras-chave: Exames Diagnósticos. Fluxo de Pacientes. *Framework*. Heurísticas. Escalonamento Hospitalar.

ABSTRACT

Efficient management of patient flow in hospital settings, especially in the scheduling of diagnostic tests, represents a critical challenge given the high demand and limited resources. The Hospital Test Scheduling Problem consists of determining the processing sequence of tests across multiple stations, considering clinical priorities and arrival times. The objective of this work is to identify an alternative capable of scheduling hospital tests using heuristic algorithms. The methodology included a systematic literature review for theoretical grounding, followed by the implementation of three scheduling algorithms: Arrival Order Priority (AOP), based on arrival order; Minimum Processing Time Priority (MPTP), based on the shortest processing time; and Roulette Selection Strategy (RSS), of a probabilistic nature. To analyze the generated solutions, a framework capable of dynamically reconstructing the processing flow from the produced data was developed. The experiments were conducted in 150 distinct scenarios, varying the number of tests and processing stations. The results indicated that AOP presented the best average performance in terms of total processing time in all scenarios analyzed, while RSS obtained the longest average times, although it expanded the exploration of the solution space. It is concluded that the scheduling policy directly impacts operational performance and that the proposed framework contributes to the analysis, validation, and integration of decision support strategies in hospital environments.

Keywords: Diagnostic Tests. Patient Flow. Framework. Heuristics. Hospital Escalation.

RESUMEN

La gestión eficiente del flujo de pacientes en entornos hospitalarios, especialmente en la programación de pruebas diagnósticas, representa un desafío crítico dada la alta demanda y los recursos limitados. El problema de programación de pruebas hospitalarias consiste en determinar la secuencia de procesamiento de las pruebas en múltiples estaciones, considerando las prioridades clínicas y los tiempos de llegada. El objetivo de este trabajo es identificar una alternativa capaz de programar pruebas hospitalarias utilizando algoritmos heurísticos. La metodología incluyó una revisión sistemática de la literatura para fundamentar la teoría, seguida de la implementación de tres algoritmos de programación: Prioridad de Orden de Llegada (AOP), basada en el orden de llegada; Prioridad de Tiempo Mínimo de Procesamiento (MPTP), basada en el tiempo de procesamiento más corto; y Estrategia de Selección por Ruleta (RSS), de naturaleza probabilística. Para analizar las soluciones generadas, se desarrolló un marco capaz de reconstruir dinámicamente el flujo de procesamiento a partir de los datos producidos. Los experimentos se llevaron a cabo en 150 escenarios distintos, variando el número de pruebas y estaciones de procesamiento. Los resultados indicaron que AOP presentó el mejor rendimiento promedio en términos de tiempo total de procesamiento en todos los escenarios analizados, mientras que RSS obtuvo los tiempos promedio más largos, aunque amplió la exploración del espacio de soluciones. Se concluye que la política de programación impacta directamente en el rendimiento operativo y que el marco propuesto contribuye al análisis, validación e integración de estrategias de apoyo a la toma de decisiones en entornos hospitalarios.

Palabras clave: Pruebas Diagnósticas. Flujo de Pacientes. *Framework*. Heurística. Escalada Hospitalaria.

1 INTRODUÇÃO

O crescimento da demanda por serviços de saúde, aliado à limitação de recursos físicos, humanos e tecnológicos em ambientes hospitalares, tem intensificado os desafios relacionados à gestão eficiente do fluxo de pacientes. Em especial, a realização de exames diagnósticos representa uma etapa crítica do atendimento, pois atrasos nesse processo impactam diretamente o tempo de permanência do paciente no sistema e a qualidade do serviço prestado. Segundo Hall (2013), a má gestão do fluxo de pacientes está entre as principais causas dos atrasos nos serviços de saúde, resultando em gargalos operacionais e sobrecarga dos recursos disponíveis.

Hospitais operam em ambientes altamente dinâmicos, nos quais a chegada de pacientes ocorre de forma variável e imprevisível, enquanto os recursos disponíveis, como equipamentos de diagnóstico e equipes especializadas, são limitados (Pereira, 2024). Litvak (2010) destaca que essa variabilidade inerente aos sistemas hospitalares exige métodos de planejamento e escalonamento capazes de lidar com mudanças constantes, uma vez que abordagens rígidas tendem a falhar diante de flutuações na demanda. Nesse sentido, a literatura recente aponta que a otimização de recursos limitados em cenários de alta complexidade demanda a adoção de tecnologias e protocolos estruturados de apoio à tomada de decisão, os quais oferecem subsídios essenciais para que gestores e especialistas garantam respostas adaptativas e eficientes (Lindner & Mastella, 2025; Cardeal *et al.* 2026).

Nesse contexto, destaca-se o Problema de Escalonamento de Exames Hospitalares, que consiste em determinar uma sequência de processamento para exames considerando múltiplas estações de processamento, tempos de chegada, prioridades clínicas e compatibilidade entre exames e estações. Problemas de escalonamento com características semelhantes são amplamente reconhecidos na literatura como computacionalmente complexos, sendo, em sua forma geral, associados à classe dos problemas NP-difíceis (Garey e Johnson, 1979). Em razão dessa complexidade, estudos anteriores, como os de Goldberg (1989), utilizaram diferentes abordagens, incluindo métodos exatos e heurísticos para tratar problemas similares de escalonamento.

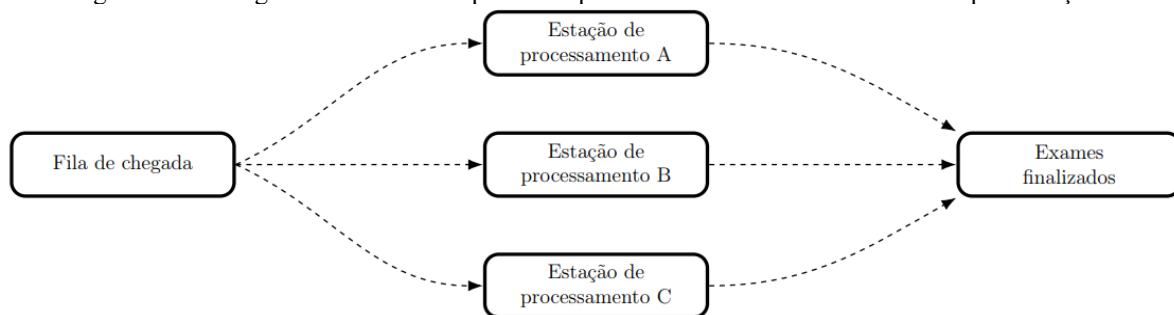
Diante desse cenário, o objetivo deste trabalho consiste em identificar uma alternativa capaz de escalonar exames hospitalares, por meio de algoritmos heurísticos e baseados em simulação de eventos discretos. Os algoritmos desenvolvidos permitem modelar e analisar o comportamento dinâmico do sistema de processamento de exames sob diferentes estratégias de ordenação e priorização, avaliando seus impactos sobre o tempo total de processamento e eficiência do fluxo de atendimento.

2 PROBLEMA DE ESCALONAMENTO DE EXAMES HOSPITALARES

O Problema de Escalonamento de Exames Hospitalares pode ser entendido como a organização do atendimento de exames diagnósticos em um ambiente com múltiplas estações de processamento,

no qual exames chegam ao longo do tempo, possuem diferentes níveis de prioridade clínica e apresentam restrições quanto às estações aptas a processá-los. Na Figura 1, apresenta-se uma visão do sistema hospitalar considerado neste trabalho, destacando as principais áreas envolvidas no processamento dos exames. O sistema é composto por uma fila de chegada, um conjunto de estações de processamento e uma área de conclusão, na qual os exames finalizados deixam o sistema. As estações representam equipamentos especializados responsáveis pela realização dos exames, enquanto a fila organiza os exames aguardando atendimento conforme a política de escalonamento adotada.

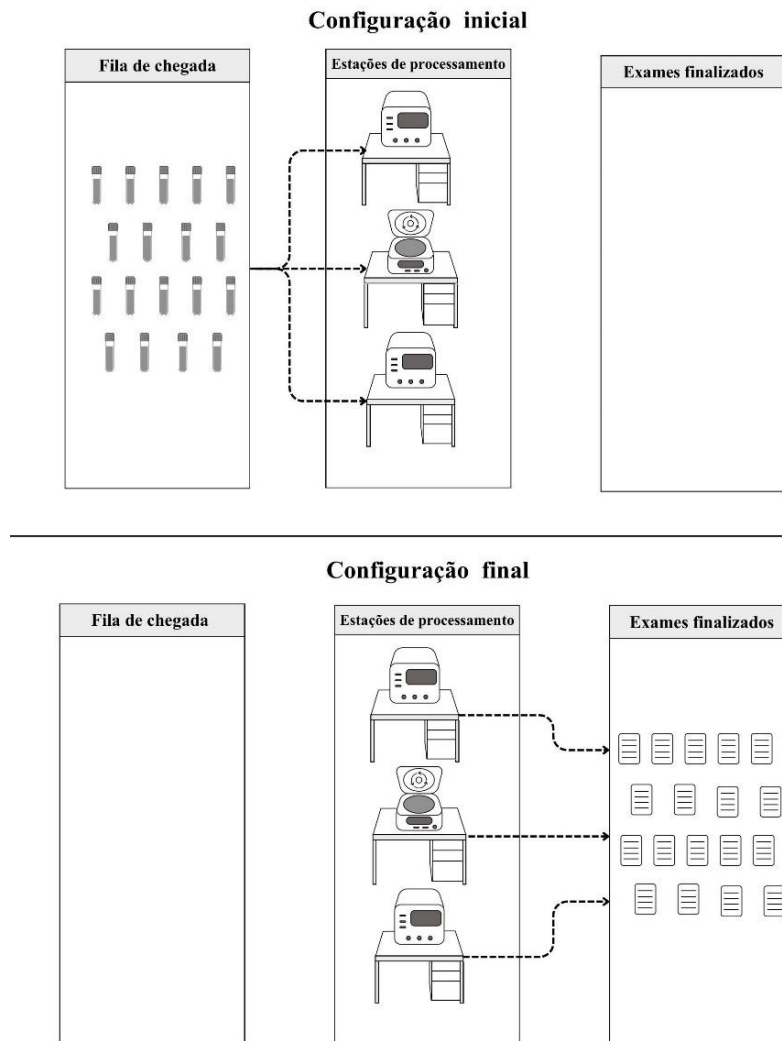
Figura 1 – Visão geral do sistema hospitalar de processamento de exames com múltiplas estações.



Fonte: Elaborada pelos autores (2026).

Os exames diagnósticos são processados em estações que operam de forma independente e não preemptiva, ou seja, uma vez iniciado o processamento de um exame, este deve ser concluído antes que a estação possa atender outro exame. Cada exame possui um instante de chegada e um tempo de processamento associado, que pode variar de acordo com o tipo de exame e a estação responsável por sua execução. A dinâmica do sistema pode ser analisada a partir de dois estados fundamentais: a configuração inicial e a configuração final. A configuração inicial corresponde ao estado no qual os exames chegam ao sistema e passam a compor a fila de espera, enquanto a configuração final representa o estado em que todos os exames foram devidamente processados e liberados. Assim como em problemas tradicionais de escalonamento, o objetivo consiste em conduzir o sistema da configuração inicial até a configuração final por meio de uma sequência de operações viáveis. Um exemplo simplificado dessas duas configurações é apresentado na Figura 2, na qual se observa um conjunto de exames aguardando atendimento e estações disponíveis, bem como o estado final após a conclusão de todos os processamentos.

Figura 2 – Representação das configurações de entrada e saída do sistema de escalonamento.



Fonte: Elaborada pelos autores (2026).

Cada exame é representado por um identificador único e ocupa uma posição lógica na fila de espera até ser alocado a uma estação compatível. O estado global do sistema é alterado quando ocorre um evento, como a chegada de um novo exame ou a finalização do processamento de uma estação, o que reforça o caráter dinâmico do problema. Os exames que chegam ao sistema hospitalar podem ainda ser classificados de acordo com suas prioridades clínicas, as quais influenciam diretamente a ordem de atendimento. Neste trabalho, consideram-se três categorias principais de exames:

Exames regulares: correspondem a procedimentos eletivos, sem caráter emergencial, que podem aguardar atendimento sem comprometer a segurança do paciente.

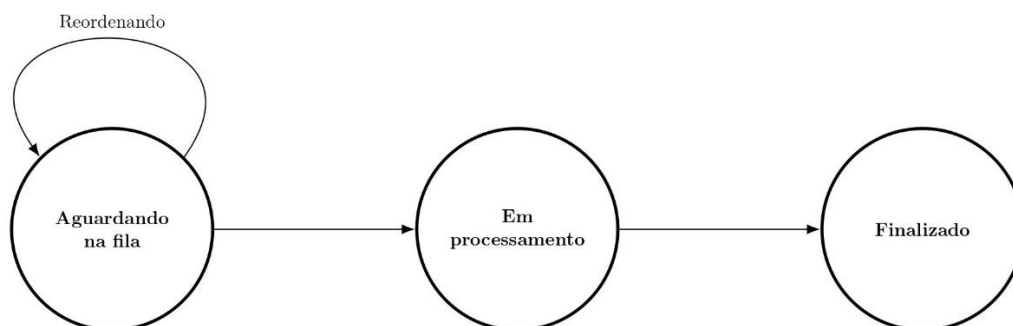
Exames prioritários: são aqueles que, embora não sejam emergenciais, demandam atendimento preferencial em relação aos exames regulares devido a condições clínicas específicas. Esses exames somente são processados após o atendimento de eventuais exames urgentes.

Exames urgentes: estão associados a situações clínicas críticas e demandam atendimento imediato. Esses exames possuem precedência sobre todos os demais tipos e devem ser alocados à próxima estação disponível sempre que possível.

A coexistência desses diferentes tipos de exames no sistema intensifica a competição pelos recursos disponíveis e evidencia a necessidade de políticas de escalonamento capazes de equilibrar prioridade clínica e eficiência operacional.

O funcionamento do sistema hospitalar pode ser descrito a partir das operações de processamento realizadas ao longo do tempo. Essas operações consistem na movimentação lógica dos exames entre os estados de espera, processamento e conclusão. Em termos gerais, um exame pode ser alocado da fila para uma estação compatível, liberado da estação após a conclusão do processamento ou reorganizado na fila em função da política de escalonamento adotada, especialmente quando novos exames chegam ao sistema. As possíveis transições entre esses estados estão representadas na Figura 3.

Figura 3 – Diagrama de estados do ciclo de processamento dos exames.



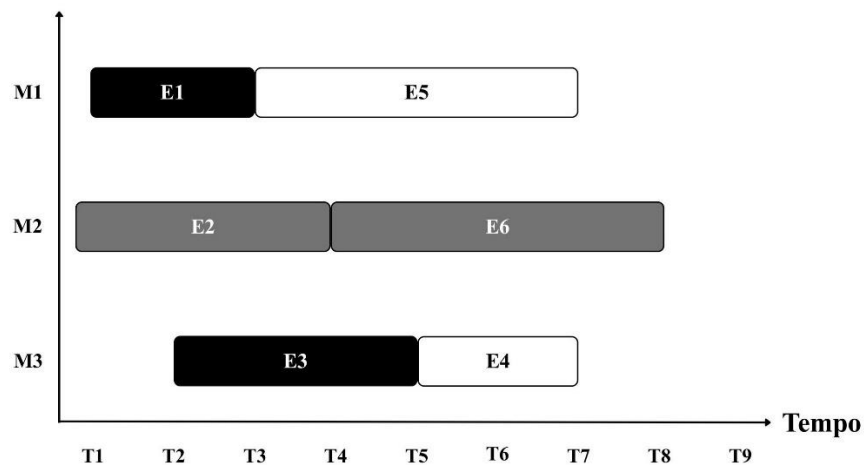
Fonte: Elaborada pelos autores (2026).

Cada exame E_i está associado a um tempo de processamento T_i que representa o tempo necessário para sua execução em uma estação compatível. Sendo N a quantidade total de exames, assim, $1 \leq i \leq N$. O desempenho do sistema pode ser avaliado a partir de métricas clássicas de escalonamento, como o tempo total de conclusão e o tempo médio de espera dos exames. Essas métricas permitem quantificar o impacto das diferentes políticas de escalonamento sobre a eficiência global do sistema e sobre o fluxo de atendimento hospitalar.

Para ilustrar os conceitos apresentados, a Figura 4 mostra um exemplo de instância simplificada do problema, composta por um conjunto reduzido de exames e estações por processamento. Nesse exemplo, observa-se que uma solução possível consiste em alocar inicialmente os exames urgentes, seguidos pelos exames prioritários e, por fim, pelos exames regulares.

Figura 4 – Exemplo de escalonamento de exames em ambiente com múltiplas estações.

id	Tempo de chegada	Duração	Prioridade
E1	T1	2	Urgente
E2	T1	3	Prioritário
E3	T2	3	Urgente
E4	T2	2	Regular
E5	T2	4	Regular
E6	T3	2	Prioritário

Estações de processamento**Níveis de prioridade:**

Regular Prioritário Urgente

Fonte: Elaborada pelos autores (2026).

É importante destacar que a verificação deve ser realizada ao iniciar o processo de decisão do próximo exame que será alocado na estação, visto que o tempo de chegada dos exames corresponde à chegada dos mesmos no ambiente hospitalar. Desta forma, se faz necessária a verificação de todos os exames a serem atendidos para que os exames urgentes e prioritários não fiquem aguardando na lista de entrada. No entanto, essa não é a única solução viável, uma vez que políticas alternativas de escalonamento podem gerar sequências distintas de processamento. Esse fato evidencia o desafio de investigar estratégias capazes de produzir soluções eficientes, reduzindo o tempo total de processamento e melhorando o fluxo do atendimento hospitalar.

3 REVISÃO DA LITERATURA

Nesta seção, apresenta-se a revisão de literatura referente a esse trabalho, bem como a descrição das principais fases desta etapa. A fim de estabelecer uma base sólida e confiável alinhada ao objetivo do trabalho, realizou-se um mapeamento sistemático de literatura, que se trata de um método de revisão bibliográfica organizado de forma estruturada e sistêmica. Esta metodologia, que segue procedimentos estabelecidos por Petersen *et al.* (2008) e Kitchenham e Charters (2007), foi aplicada com o propósito de constatar os principais trabalhos relacionados a esta pesquisa. Esta etapa da pesquisa foi estruturada em três fases: Planejamento, Condução e Resultados. Estas fases serão descritas nas seções seguintes.

3.1 FASE 1 - PLANEJAMENTO

A fase de planejamento compreende a elaboração do protocolo de pesquisa e inclui as seguintes ações: definição das questões de pesquisa; definição da *string* de busca; definição das bases de dados e definição dos critérios de inclusão e exclusão. Cada uma dessas ações é detalhada a seguir.

3.1.1 Definição das questões de pesquisa

Primeiramente, foram definidas as Questões de Pesquisa (QPs) que orientaram a fase de condução do mapeamento. Ao todo, para esta pesquisa, foram definidas duas QPs, que abordam os seguintes aspectos: o objetivo do trabalho relacionado e qual o algoritmo de otimização utilizado. As questões formuladas foram as seguintes:

PQ 1 - Qual o objetivo do trabalho?

- Escalonar enfermeiros/médicos
- Rotas ou posicionamento de veículos hospitalares
- Atendimento dos pacientes
- Alocação/logística dos produtos
- Outros

QP2 - Qual o algoritmo adotado no trabalho?

- Programação linear
- Algoritmo genético
- Algoritmo híbrido
- Outro

3.1.2 Definição da *string* de busca

Definidas as Questões de Pesquisa, a fase seguinte consistiu na elaboração de uma *string* de busca com os principais termos relacionados à temática, conforme destacado abaixo:

escalonamento AND heurística AND hospital AND (operações OR tarefas)

3.1.3 Definição da base de dados

Posteriormente, optou-se por utilizar a base de dados Google Scholar para a condução deste estudo, visto que essa base de dados apresentou uma quantidade significativa de trabalhos durante a definição da *string* de busca.

3.1.4 Definição dos critérios de inclusão e exclusão

Os critérios de inclusão e exclusão que todos os artigos deverão satisfazer. Os critérios definidos são:

Critérios de inclusão:

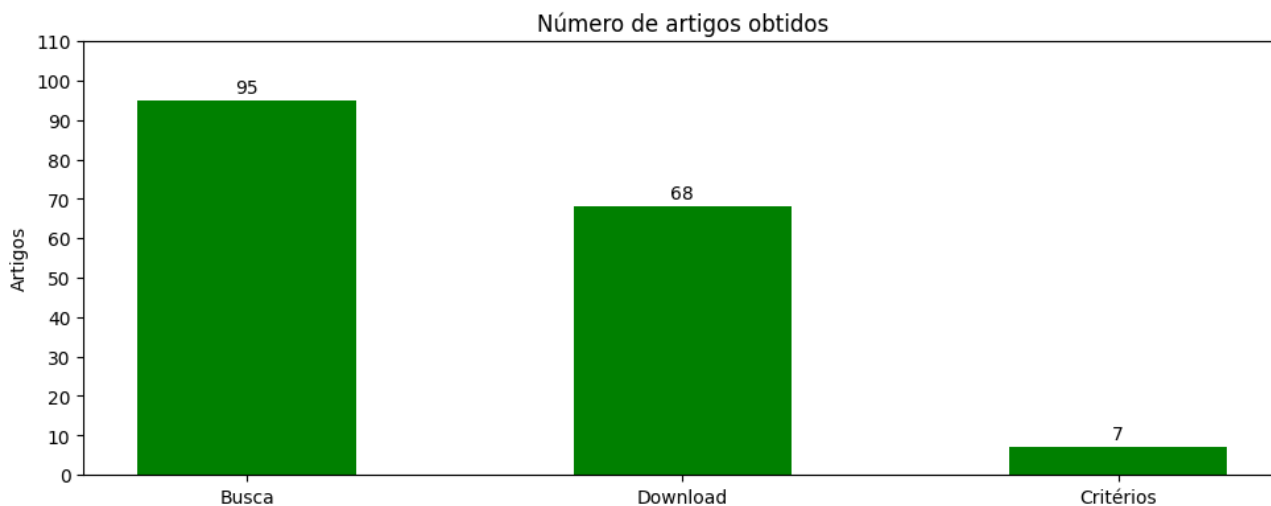
- Estudos completos publicados em revistas ou conferências sobre o Problema de Escalonamento em contextos hospitalares;
- Estudos teóricos ou experimentais com o objetivo de apresentar conceitos para o entendimento da área.

Critérios de exclusão:

- Estudos que não estejam relacionados ao Problema de Escalonamento em contextos hospitalares;
- Estudos que não respondem a nenhuma das questões de pesquisa;
- Artigos convidados, tutoriais, relatórios técnicos que não passam pelo critério de avaliação das conferências ou revistas;
- Estudos não disponíveis para *download* eletronicamente.

3.2 FASE 2 - CONDUÇÃO

O planejamento definido anteriormente foi colocado em prática na fase de condução. A *string* de busca foi aplicada no Google Scholar. A partir dos resultados obtidos, realizou-se o *download* dos estudos disponíveis. Logo em seguida, os trabalhos foram analisados considerando os critérios de inclusão e exclusão. A Figura 5 descreve a quantidade de estudos identificados na realização da busca, bem como aqueles que foram submetidos ao *download* e à filtragem pelos critérios. Após a filtragem pelos critérios de inclusão e exclusão, foram selecionados 7 artigos, os quais foram submetidos a uma leitura completa para responder às duas questões da pesquisa.

Figura 5 – Quantidade de artigos obtidos após a busca, *download* e aplicação dos critérios de inclusão e exclusão.

Fonte: Elaborada pelos autores (2026).

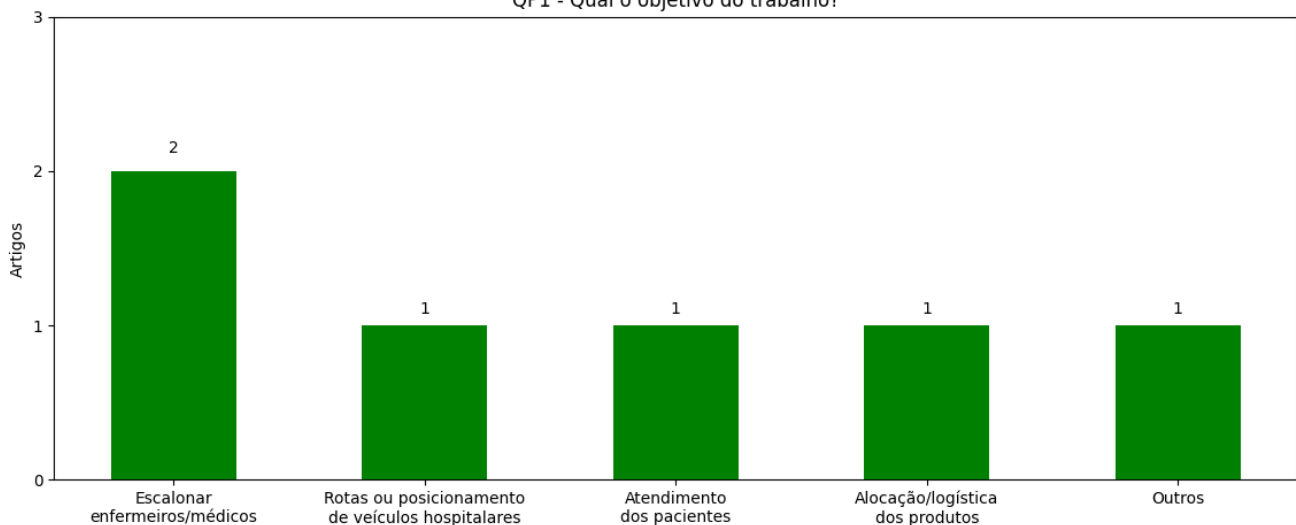
3.3 FASE 3 - RESULTADOS

A seguir, será apresentada a análise dos resultados obtidos a partir das respostas das questões de pesquisa QP1 e QP2. Na Figura 6, observa-se um gráfico de barras correspondente à QP1, referente ao objetivo do trabalho. Ao observar o gráfico, é possível notar que os trabalhos analisados apresentam diferentes propósitos. O trabalho de Meister (2020) teve como objetivo escalonar o horário dos enfermeiros e médicos no Hospital das Clínicas de Porto Alegre, enquanto o trabalho de Ackermann (2020) focou no atendimento dos pacientes, buscando otimizar os processos relacionados à demanda hospitalar. O trabalho de Flórez (2020) visou melhorar o gerenciamento de um sistema de emergência hospitalar, com aplicação de técnicas de inteligência artificial para suporte à tomada de decisão. Já o trabalho de Silva (2023) também concentrou-se no escalonamento dos horários dos enfermeiros e médicos. No caso do trabalho de Mondo (2024), o objetivo foi a alocação e logística de produtos, enquanto o trabalho de Reis (2024) tratou da alocação de voluntários para atuação em uma ONG. Por fim, o trabalho de Ferrari (2022) teve como abordagem o problema de rotas e posicionamento de veículos hospitalares, mostrando a aplicação de métodos de otimização em logística hospitalar.

Em relação à QP2, que buscou identificar o algoritmo adotado em cada trabalho, foi possível observar uma diversidade de abordagens, conforme mostra a Figura 7. Na alternativa Programação Linear, destacam-se os trabalhos de Mondo (2024) e Reis (2024), ambos fundamentados em modelos matemáticos clássicos de otimização. Já a alternativa Algoritmo Genético é referida nos estudos de Flórez (2020) e Ferrari (2022), que exploraram essa meta-heurística para lidar com problemas de otimização complexos. A alternativa Algoritmo Híbrido foi contemplada apenas no trabalho de Silva (2023), o qual adotou um algoritmo com duas fases compostas por duas heurísticas: na primeira fase, uma heurística gulosa para construção da solução inicial, e, na segunda fase, uma heurística de busca local para o refinamento da solução. Por fim, a categoria Outros reuniu trabalhos que exploraram

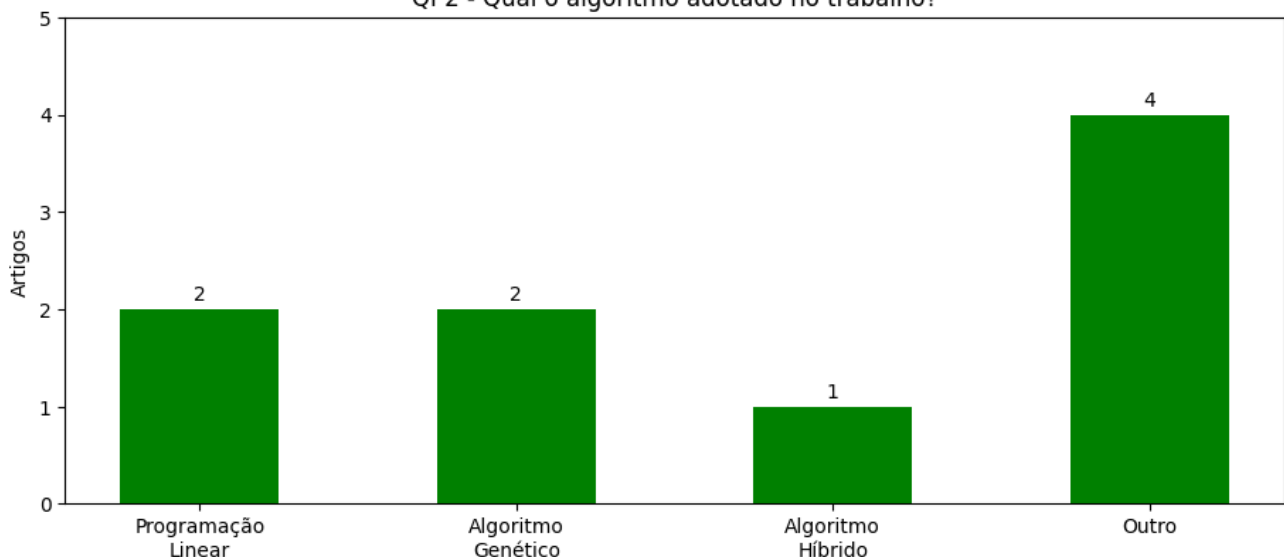
abordagens menos convencionais. O estudo de Meister (2020) utilizou a técnica *Variable Neighborhood Search* (VNS); o estudo de Ackermann (2020) aplicou modelos de previsão baseados em ARIMA e no mapa logístico caótico; ademais, os trabalhos de Silva (2023) e Ferrari (2022) também integraram técnicas adicionais às suas propostas. O primeiro fez o uso de técnicas de aprendizado por reforço e computação evolutiva, enquanto o segundo adotou o método exato ϵ -Constraint, além de três variantes de uma heurística baseada em aleatoriedade (HCBA).

Figura 6 – Resultado da Questão de Pesquisa 1.
QP1 - Qual o objetivo do trabalho?



Fonte: Elaborada pelos autores (2025).

Figura 7 – Resultado da Questão de Pesquisa 2.
QP2 - Qual o algoritmo adotado no trabalho?



Fonte: Elaborada pelos autores (2026).

Em resumo, observa-se que o Problema de Escalonamento em contextos hospitalares tem sido abordado sob diferentes perspectivas, envolvendo desde a alocação de profissionais e recursos até a

otimização de fluxos de atendimento e logística. Nota-se, ainda, a predominância do uso de heurísticas, meta-heurísticas e abordagens híbridas, o que reforça a complexidade inerente a esse tipo de problema e a limitação de métodos exatos em cenários realistas. Entretanto, apesar da diversidade de aplicações identificadas, percebe-se uma lacuna relacionada ao escalonamento específico de exames hospitalares quando analisado sob a ótica do fluxo operacional e da tomada de decisão ao longo do tempo. Dessa forma, os resultados desta revisão evidenciam a relevância e a oportunidade do presente trabalho, ao direcionar a investigação para uma solução computacional aplicada ao Problema de Escalonamento de Exames em ambientes hospitalares.

4 METODOLOGIA

Nesta seção são apresentados os procedimentos metodológicos adotados para o desenvolvimento e validação da abordagem desenvolvida neste trabalho para o Problema de Escalonamento de Exames Hospitalares. Considerando que o objetivo consiste em identificar uma alternativa capaz de escalonar exames hospitalares por meio de algoritmos heurísticos e simulação de eventos discretos, estruturou-se uma metodologia que contemplasse tanto a fundamentação teórica do problema quanto a validação experimental da abordagem investigada. A pesquisa foi organizada em cinco passos sequenciais e interdependentes: Revisão da Literatura; Definição e Geração dos Cenários de Teste; Definição dos Algoritmos de Escalonamento; Desenvolvimento do *Framework*; e Procedimento Experimental e Análise dos Resultados.

4.1 PASSO 1 – REVISÃO DA LITERATURA

O primeiro passo realizado no desenvolvimento da metodologia deste trabalho foi a Revisão da Literatura, com o objetivo de identificar os principais estudos que abordam o Problema de Escalonamento em contextos hospitalares. A descrição detalhada das três fases conduzidas durante essa revisão encontra-se apresentada na seção 3. Inicialmente, a aplicação da *string* de busca na base de dados selecionada resultou na identificação de 95 trabalhos. Após o *download* dos estudos disponíveis e a aplicação dos critérios de inclusão e exclusão, foram selecionados 7 artigos primários para análise, conforme ilustrado na Figura 5. A partir dos artigos selecionados, foi possível identificar os principais objetivos investigados, os algoritmos de otimização empregados e as características consideradas relevantes nos estudos analisados. De modo geral, observou-se a predominância de heurísticas, meta-heurísticas e abordagens híbridas aplicadas a diferentes problemas hospitalares, como alocação de profissionais, atendimento de pacientes e logística de recursos. Entretanto, não foram encontrados trabalhos que abordassem especificamente o escalonamento de exames hospitalares sob uma perspectiva de fluxo operacional dinâmico, o que reforça a motivação do presente estudo.

4.2 PASSO 2 – DEFINIÇÃO E GERAÇÃO DE CASOS DE TESTE

Para avaliar o desempenho dos algoritmos implementados, tornou-se necessário definir um conjunto de cenários que representassem diferentes configurações operacionais de um ambiente hospitalar. Os cenários foram gerados de forma sintética, variando-se dois parâmetros principais: o número de estações de processamento (3, 4 e 5) e a quantidade de exames (100, 200, 300, 400 e 500). Para cada combinação desses parâmetros, foram criadas 10 instâncias distintas, totalizando 150 cenários de teste. As instâncias diferem quanto à distribuição dos horários de chegada, aos tempos de processamento e aos níveis de prioridade clínica (urgente, prioritário e regular). Essa variação permitiu avaliar o comportamento dos algoritmos sob diferentes níveis de carga e disponibilidade de recursos.

4.3 PASSO 3 – DEFINIÇÃO DOS ALGORITMOS

Com base na predominância de heurísticas e metaheurísticas nos resultados obtidos na revisão da literatura, foram elaboradas três estratégias de escalonamento para implementação e avaliação no *framework* proposto. Todos os algoritmos respeitam a precedência por prioridade clínica (urgente > prioritário > regular) e operam no contexto de um simulador de eventos discretos.

O procedimento geral de escalonamento adotado no *framework* é apresentado no Algoritmo 1 na Figura 8, que, a partir dessa estrutura comum, descreve o fluxo geral do processo de escalonamento, sendo responsável por simular a evolução do sistema ao longo do tempo e coordenar a alocação dos exames às estações de processamento. Inicialmente, nas linhas 1–3, são realizadas operações de pré-processamento, nas quais os horários de chegada são convertidos para minutos e a lista de exames é ordenada em ordem crescente de chegada. Nas linhas 4–7, são inicializadas as estruturas auxiliares do algoritmo. A *FilaProntos* (linha 4) e a *FilaFinalizados* (linha 5) são inicializadas como vazias. Em seguida, define-se o *tempoInicial* como o menor horário de chegada (linha 6) e o *tempoAtual* é inicializado com esse valor (linha 7). A linha 8 define o laço principal do algoritmo, estruturado como um comando enquanto, cuja execução permanece ativa enquanto existir pelo menos uma das seguintes condições: exames ainda não processados, exames aguardando na fila de prontos ou estações ocupadas. Esse laço representa a evolução temporal do sistema. No interior desse laço, as linhas 9–11 definem um segundo laço enquanto, responsável por atualizar a *FilaProntos*. Enquanto existirem exames cuja chegada seja menor ou igual ao *tempoAtual*, esses exames são movidos para a *FilaProntos* (linha 10) e removidos da lista original (linha 11).

Figura 8 – Pseudocódigo do algoritmo.

Algoritmo 1: Escalonamento

Entrada: Lista *Exames*, Lista *Estacoes*, Política *P*
Saída: Tempo total de processamento

- 1 **início;**
- 2 converter horários de chegada para minutos;
- 3 ordenar *Exames* por horário de chegada crescente;
- 4 *FilaProntos* $\leftarrow \emptyset$;
- 5 *FilaFinalizados* $\leftarrow \emptyset$;
- 6 *tempoInicial* \leftarrow menor horário de chegada;
- 7 *tempoAtual* \leftarrow *tempoInicial*;
- 8 **enquanto** *Exames* $\langle \rangle \emptyset$ **ou** *FilaProntos* $\langle \rangle \emptyset$ **ou** \exists estação ocupada **faça**
- 9 **enquanto** *Exames* $\langle \rangle \emptyset$ **e** *Exames*[0].chegada \leq *tempoAtual* **faça**
- 10 mover *Exames*[0] para *FilaProntos*;
- 11 remover *Exames*[0];
- 12 **se** *P* = AOP **então**
- 13 ordenar *FilaProntos* por prioridade decrescente
- 14 e chegada crescente;
- 15 **senão se** *P* = MPTP **então**
- 16 ordenar *FilaProntos* por prioridade decrescente
- 17 e tempo processamento crescente;
- 18 **senão se** *P* = RSS **então**
- 19 ordenar *FilaProntos* por prioridade decrescente;
- 20 **para cada** *estacao* \in *Estacoes* **faça**
- 21 **se** *estacao.livre* **e** *FilaProntos* $\langle \rangle \emptyset$ **então**
- 22 *compativeis* \leftarrow exames em *FilaProntos* compatíveis
- 23 com *estacao* e mesma prioridade do primeiro;
- 24 **se** *compativeis* $\langle \rangle \emptyset$ **então**
- 25 *selecionado* \leftarrow roleta(*compativeis*);
- 26 *estacao.adicionar*(*selecionado*);
- 27 *estacao.livre* \leftarrow falso;
- 28 remover *selecionado* de *FilaProntos*;
- 29 **para cada** *estacao* \in *Estacoes* **faça**
- 30 **se** *estacao.livre* **e** *FilaProntos* $\langle \rangle \emptyset$ **então**
- 31 **para cada** *exame* \in *FilaProntos* **faça**
- 32 **se** *estacao* pode processar *exame.tipo* **então**
- 33 *estacao.adicionar*(*exame*);
- 34 *estacao.livre* \leftarrow falso;
- 35 remover *exame* de *FilaProntos*;
- 36 **para cada** *estacao* \in *Estacoes* **faça**
- 37 *finalizado* \leftarrow *estacao.processar*();
- 38 **se** *finalizado* $\langle \rangle$ nulo **então**
- 39 adicionar *finalizado* a *FilaFinalizados*;
- 40 *tempoAtual* \leftarrow *tempoAtual* + 1;
- 41 *tempoTotal* \leftarrow *tempoAtual* – *tempoInicial*;
- 42 **retorne** *tempoTotal*;
- 43 **fim;**

Fonte: Elaborada pelos autores (2026).

As linhas 12–18 implementam a aplicação da política de escalonamento. Na linha 12, verifica-se se $P=AOP$, realizando a ordenação por prioridade e chegada (linhas 13–14), que é melhor detalhada na subseção 4.3.1. Na linha 15, verifica-se se $P=MPTP$, cuja ordenação considera prioridade e menor tempo de processamento (linhas 16–17), cujo detalhamento encontra-se na subseção 4.3.2. Já nas linhas 18–28, trata-se do caso $P=RSS$, no qual a seleção dos exames envolve um processo probabilístico baseado no método da roleta, conforme descrito na subseção 4.3.3. Nas linhas 29–35, o algoritmo realiza a alocação dos exames às estações por meio de estruturas para cada, verificando as estações livres e buscando exames compatíveis na FilaProntos. Quando encontrado um exame compatível, este é alocado à estação, que passa a ser marcada como ocupada. Nas linhas 36–39, ocorre o processamento dos exames nas estações. Cada estação executa seu exame corrente e, caso haja finalização, o exame é adicionado à FilaFinalizados. Na linha 40, o tempo atual do sistema é incrementado, representando o avanço da simulação. Por fim, nas linhas 41–42, é calculado o tempo total de processamento, que é retornado como saída do algoritmo.

4.3.1 Algoritmo *Arrival Order Priority* (AOP)

O primeiro algoritmo é determinístico e realiza a ordenação dos exames com base na prioridade clínica. Entre exames pertencentes ao mesmo nível de prioridade, a decisão é tomada considerando o instante de chegada ao sistema, seguindo a política *First In, First Out* (FIFO). Essa abordagem representa uma política simples e amplamente adotada na prática hospitalar.

4.3.2 Algoritmo *Minimum Processing Time Priority* (MPTP)

O segundo algoritmo também é determinístico e mantém a precedência por prioridade clínica. Entretanto, para exames com mesma prioridade, a seleção é realizada com base no menor tempo de processamento estimado. Essa estratégia busca reduzir o tempo médio de espera ao priorizar a execução de exames mais curtos.

4.3.3 Algoritmo *Roulette Selection Strategy* (RSS)

O terceiro algoritmo introduz um componente probabilístico na decisão. Após filtrar os exames pela maior prioridade disponível, a escolha entre exames de mesma prioridade é realizada por meio do método da roleta. A probabilidade de seleção é definida de forma inversamente proporcional ao tempo de processamento, permitindo maior diversidade nas soluções geradas.

4.4 PASSO 4 – DEFINIÇÃO DO *FRAMEWORK*

Durante o desenvolvimento dos algoritmos de escalonamento, identificou-se a necessidade de uma ferramenta que possibilitasse visualizar e analisar o comportamento das soluções geradas ao longo do tempo. Em problemas de escalonamento de exames hospitalares, a sequência de decisões influencia diretamente o fluxo operacional, tornando relevante a observação detalhada das operações realizadas. Diante disso, foi desenvolvido um *framework* com a finalidade de consumir e visualizar soluções previamente geradas pelos algoritmos de escalonamento. Diferentemente de um ambiente de execução algorítmica, o *framework* não realiza a otimização propriamente dita, mas atua como uma camada de análise e integração, permitindo a interpretação estruturada das soluções produzidas.

4.4.1 Estrutura de entrada de dados

O funcionamento do *framework* baseia-se na leitura e interpretação de arquivos estruturados que descrevem tanto o cenário do problema quanto a solução gerada pelos algoritmos de escalonamento. Para isso, o sistema permite o *upload* de dois tipos de arquivos distintos. O primeiro corresponde ao arquivo de instância, em formato de texto, que contém as informações iniciais do cenário considerado. Nesse arquivo são definidos parâmetros como o número de estações de processamento, a lista de exames, seus respectivos tempos de processamento, níveis de prioridade clínica e demais configurações necessárias à caracterização do problema. Esse formato segue o mesmo padrão utilizado como entrada pelos algoritmos de escalonamento, garantindo consistência entre a fase de geração da solução e a etapa de visualização no *framework*. O segundo tipo de arquivo corresponde ao arquivo de solução, em formato JSON, que descreve o histórico completo do escalonamento realizado. Nesse arquivo são registradas todas as operações executadas ao longo do tempo, incluindo a atribuição de exames às estações e os respectivos instantes de início e término de processamento. A partir dessas informações estruturadas, o *framework* é capaz de reconstruir dinamicamente a execução da solução, permitindo a visualização do fluxo operacional desde o primeiro até o último evento registrado.

4.4.2 Integração com outros sistemas

A adoção do formato JSON para representação das soluções possibilita a integração com outros sistemas computacionais. Desde que o arquivo esteja estruturado conforme o padrão definido pelo *framework*, soluções geradas por algoritmos externos ou por sistemas legados podem ser importadas e visualizadas. Essa característica torna o *framework* independente da implementação dos algoritmos de otimização, permitindo sua utilização como ferramenta de análise, validação e apoio à tomada de decisão em diferentes contextos.

4.5 PASSO 5 – PROCEDIMENTO EXPERIMENTAL E ANÁLISE DOS RESULTADOS

Após a implementação dos algoritmos de escalonamento e o desenvolvimento do *framework*, foi conduzido um conjunto de experimentos computacionais com o objetivo de avaliar o comportamento das estratégias sob diferentes configurações de cenário. Os experimentos foram realizados considerando 150 cenários distintos, definidos a partir de combinações entre número de máquinas e número de exames. Para os algoritmos determinísticos, foram geradas 10 instâncias distintas para cada configuração de cenário, sendo os resultados consolidados por meio da média dos valores obtidos nessas instâncias. No caso do algoritmo probabilístico *Roulette Selection Strategy*, cada uma das instâncias foi executada 30 vezes, utilizando sementes distintas previamente definidas com base nas casas decimais do número π , que será melhor aprofundado na subseção 5.1. Inicialmente, foi calculada a média das 30 execuções para cada instância e, posteriormente, a média geral para cada configuração de cenário. Durante cada execução, foram registradas duas métricas principais:

- Tempo total de processamento, correspondente ao instante de conclusão do último exame do cenário;
- Tempo de execução computacional do algoritmo, medido em segundos.

Optou-se por concentrar a análise nessas métricas por representarem, respectivamente, o desempenho operacional do sistema e o custo computacional associado à geração da solução. Os resultados obtidos foram posteriormente organizados em tabelas comparativas, permitindo analisar o comportamento médio das estratégias em diferentes escalas de problema, bem como observar o impacto do aumento do número de exames e da capacidade do sistema no tempo total de processamento. A partir dessa consolidação, tornou-se possível avaliar o desempenho relativo dos algoritmos implementados e discutir suas características sob diferentes condições operacionais.

5 RESULTADOS

Nesta seção, serão descritos o ambiente de execução dos experimentos, os resultados obtidos com os algoritmos de escalonamento avaliados e um exemplo de visualização de uma solução utilizando o *framework* desenvolvido.

5.1 CONFIGURAÇÃO DO AMBIENTE

Os experimentos realizados neste trabalho foram executados em um computador equipado com processador AMD Ryzen 5 3500U com *Radeon Vega Mobile Gfx* (2.10 GHz), 8 GB de memória RAM (2400 MT/s), 2 GB de memória gráfica (AMD Radeon™ Vega 8 *Graphics*) e sistema operacional Windows 11 *Home Single Language*, versão 23H2 (compilação 26200.7840). Os algoritmos de escalonamento foram implementados na linguagem de programação Python, versão 3.11.5. Para a

geração de números pseudoaleatórios, foi utilizada a biblioteca *random* da linguagem. Para a definição das sementes empregadas no algoritmo probabilístico *Roulette Selection Strategy*, foram utilizadas as casas decimais do número $\pi = 3.14159265358979323846\dots$, considerando sêxtuplas consecutivas para a geração das sementes. Assim, a cada seis casas decimais, uma nova semente foi definida, por exemplo: $s_1 = 141592$, $s_2 = 653589$, $s_3 = 793238$, ..., até s_{30} , totalizando 30 sementes distintas utilizadas nos experimentos. Além disso, o *framework* desenvolvido para visualização das soluções foi implementado utilizando tecnologias web, incluindo HTML, CSS e JavaScript, responsáveis pela construção da interface gráfica e pela renderização dinâmica do fluxo de escalonamento a partir dos arquivos de solução em formato JSON.

5.2 EXECUÇÃO DOS ALGORITMOS

Para avaliar o comportamento das estratégias de escalonamento propostas neste trabalho, os algoritmos receberam como entrada o conjunto de cenários gerados conforme descrito na seção 4.2. Cada cenário é identificado pelo padrão Mx_Ey , em que x representa o número de máquinas (estações de exame) e y a quantidade de exames a serem processados. Por exemplo, o cenário $M3_E100$ corresponde a uma configuração composta por 3 estações de atendimento e 100 exames. Para cada combinação de máquinas e exames, foram geradas 10 instâncias distintas, variando-se a distribuição dos tempos de processamento e das prioridades clínicas, de modo a representar diferentes condições operacionais do sistema hospitalar. Os algoritmos avaliados neste experimento foram:

- *Arrival Order Priority* (AOP);
- *Minimum Processing Time Priority* (MPTP);
- *Roulette Selection Strategy* (RSS).

Para os algoritmos determinísticos (AOP e MPTP), os resultados apresentados na Tabela 1 correspondem à média obtida a partir das 10 instâncias geradas para cada configuração de cenário.

No caso do algoritmo probabilístico RSS, os resultados equivalem às médias consolidadas após 30 execuções para cada instância, utilizando sementes distintas geradas a partir das casas decimais do número π , conforme descrito na seção 5.1. A Tabela 1 apresenta a síntese dos resultados obtidos. Cada algoritmo é representado por duas colunas:

- **Valor (min):** média do tempo total de processamento dos exames, em minutos;
- **T (s):** média do tempo computacional necessário para execução do algoritmo, em segundos.

Tabela 1 – Resultados obtidos com a realização dos experimentos com os algoritmos.

Instâncias	AOP		MPTP		RSS	
	Valor (min)	T(s)	Valor (min)	T(s)	Valor (min)	T(s)
M3_E100	1571.90	0.0042	1643.30	0.0052	1737.11	0.0123
M3_E200	2955.20	0.0166	3104.40	0.0072	3201.75	0.0441
M3_E300	4598.90	0.0433	4775.20	0.0117	5199.96	0.1097
M3_E400	5643.60	0.0196	6080.20	0.0205	6389.65	0.1787
M3_E500	7906.80	0.0271	8409.20	0.0260	8592.84	0.2740
M4_E100	1248.20	0.0037	1296.90	0.0045	1407.42	0.0134
M4_E200	2309.40	0.0074	2425.30	0.0080	2685.88	0.0651
M4_E300	3520.00	0.0132	3916.90	0.0159	4284.23	0.1305
M4_E400	4645.90	0.0179	5041.70	0.0203	5578.67	0.1716
M4_E500	5504.20	0.0231	6030.90	0.0251	6410.81	0.2605
M5_E100	1105.10	0.0042	1195.60	0.0046	1295.80	0.0149
M5_E200	1975.40	0.0081	2216.70	0.0099	2333.20	0.0375
M5_E300	3118.20	0.0227	3400.60	0.0207	3766.66	0.1412
M5_E400	4362.00	0.0379	4771.70	0.0364	5369.07	0.2441
M5_E500	4566.80	0.0319	5067.90	0.0474	5741.38	0.2913

Fonte: www.google.com

A partir dos valores apresentados na Tabela 1, observa-se um padrão consistente entre as estratégias avaliadas. Em todas as configurações analisadas, o algoritmo *Arrival Order Priority* (AOP) apresentou os menores valores médios de tempo total de processamento. O *Minimum Processing Time Priority* (MPTP) apresentou desempenho intermediário, enquanto o *Roulette Selection Strategy* (RSS) obteve, de forma recorrente, os maiores tempos totais. Verifica-se ainda que, à medida que o número de exames aumenta (E100 → E500), o tempo total de processamento cresce de maneira proporcional em todas as estratégias, demonstrando coerência estrutural do modelo frente ao aumento da carga do sistema. Contudo, a diferença entre as abordagens torna-se mais expressiva nos cenários de maior porte. Em configurações como *M5_E500*, a diferença absoluta entre AOP e MPTP ultrapassa centenas de minutos, representando variações percentuais superiores a 10%, o que evidencia que a escolha da política de escalonamento exerce impacto crescente sob alta demanda.

A estratégia MPTP foi elaborada com base no princípio do menor tempo de processamento primeiro, buscando reduzir o tempo médio de espera ao priorizar exames de menor duração. Em termos teóricos, heurísticas desse tipo tendem a produzir ganhos relacionados à redução de filas médias. Entretanto, no contexto deste trabalho, a existência de prioridades clínicas impõe uma restrição adicional ao processo decisório. Exames com maior prioridade podem ser antecipados na fila mesmo quando apresentam maior tempo de processamento, fazendo com que a ordenação deixe de ser exclusivamente temporal e passe a obedecer prioritariamente à hierarquia clínica. Essa sobreposição de critérios reduz parte da vantagem teórica associada ao escalonamento por menor tempo de processamento, o que explica o fato de o MPTP não ter superado o AOP nos cenários analisados. Em

ambientes hospitalares reais, onde a prioridade clínica é determinante, estratégias baseadas apenas na duração das tarefas tendem a sofrer limitações operacionais.

O RSS, por sua vez, apresentou os maiores valores médios de tempo total de processamento em todas as instâncias avaliadas. A diferença percentual em relação ao AOP variou aproximadamente entre 8% nos cenários menores, podendo ultrapassar 20% nas configurações com maior número de exames. Esse comportamento é compatível com a natureza probabilística da estratégia. Ao realizar a seleção dos exames com base em probabilidades associadas às prioridades clínicas, o RSS permite explorar diferentes sequências de escalonamento ao longo das execuções. Embora essa característica amplie o espaço de busca por soluções alternativas, ela também introduz maior dispersão na ordem de processamento. Pequenas decisões probabilísticas acumuladas ao longo da execução podem resultar em sequências menos regulares, afetando o balanceamento entre estações e, conseqüentemente, elevando o tempo total de processamento, especialmente em cenários de maior escala. Apesar do desempenho inferior sob o critério de tempo total de processamento, a inclusão do RSS no estudo possui relevância metodológica. Estratégias probabilísticas são amplamente utilizadas em problemas complexos para explorar diferentes arranjos de solução e evitar padrões fixos de alocação. No entanto, no contexto específico deste experimento, cujo foco principal é a minimização do tempo total de processamento, a variabilidade introduzida não resultou em ganhos de eficiência global.

Do ponto de vista computacional, os três algoritmos apresentaram tempos de execução da mesma ordem de grandeza. Embora o RSS tenha registrado valores ligeiramente superiores em alguns cenários, não se observa diferença estrutural significativa de complexidade entre as abordagens. Mesmo nas configurações de maior porte, os tempos permaneceram inferiores a um segundo por execução, evidenciando a viabilidade prática das estratégias como ferramentas de apoio à decisão em ambientes hospitalares.

5.3 EXEMPLO DE EXECUÇÃO DE UMA SOLUÇÃO NO *FRAMEWORK*

Após a obtenção das soluções por meio das estratégias de escalonamento avaliadas, torna-se fundamental disponibilizar um mecanismo que permita visualizar e verificar a sequência de processamento gerada pelos algoritmos. Nesse contexto, o *framework* desenvolvido neste trabalho tem como objetivo possibilitar a análise visual da alocação dos exames às estações de atendimento, permitindo a validação da viabilidade operacional da solução encontrada. Nesta subseção, é apresentado um exemplo ilustrativo de execução de uma instância no *framework*, demonstrando o fluxo desde o carregamento dos dados até a finalização da simulação da solução.

Inicialmente, o usuário deve carregar o arquivo de instância contendo as informações do cenário a ser analisado, tais como número de máquinas, quantidade de exames, tempos de

processamento e prioridades clínicas. A Figura 9 apresenta a interface do *framework* no momento em que a instância é carregada. Nessa etapa, o sistema realiza a leitura do arquivo e exibe na tela os principais dados do cenário selecionado.

Figura 9 – Interface do *framework* após o carregamento da instância do problema.



Fonte: Elaborada pelos autores (2026).

Após o carregamento da instância, o próximo passo consiste na inserção da solução gerada por um dos algoritmos de escalonamento. A Figura 10 ilustra o momento em que o arquivo de solução é importado para o *framework*. Nesse estágio, a sequência de execução dos exames é apresentada ao usuário, normalmente organizada em forma de lista ou estrutura ordenada, permitindo a visualização prévia da ordem definida pelo algoritmo.

Figura 10 – Importação da solução gerada por um algoritmo de escalonamento no *framework*.



Fonte: Elaborada pelos autores (2026).

Com a solução carregada, o usuário pode iniciar a simulação do processamento. A Figura 11 apresenta a interface durante a execução da solução, na qual é possível observar a alocação dinâmica dos exames às respectivas estações de atendimento ao longo do tempo. À medida que a simulação avança, o estado das máquinas é atualizado, evidenciando quais exames estão em processamento e quais já foram concluídos. Esse recurso permite ao especialista verificar o comportamento da estratégia de escalonamento e identificar possíveis gargalos ou períodos de ociosidade.

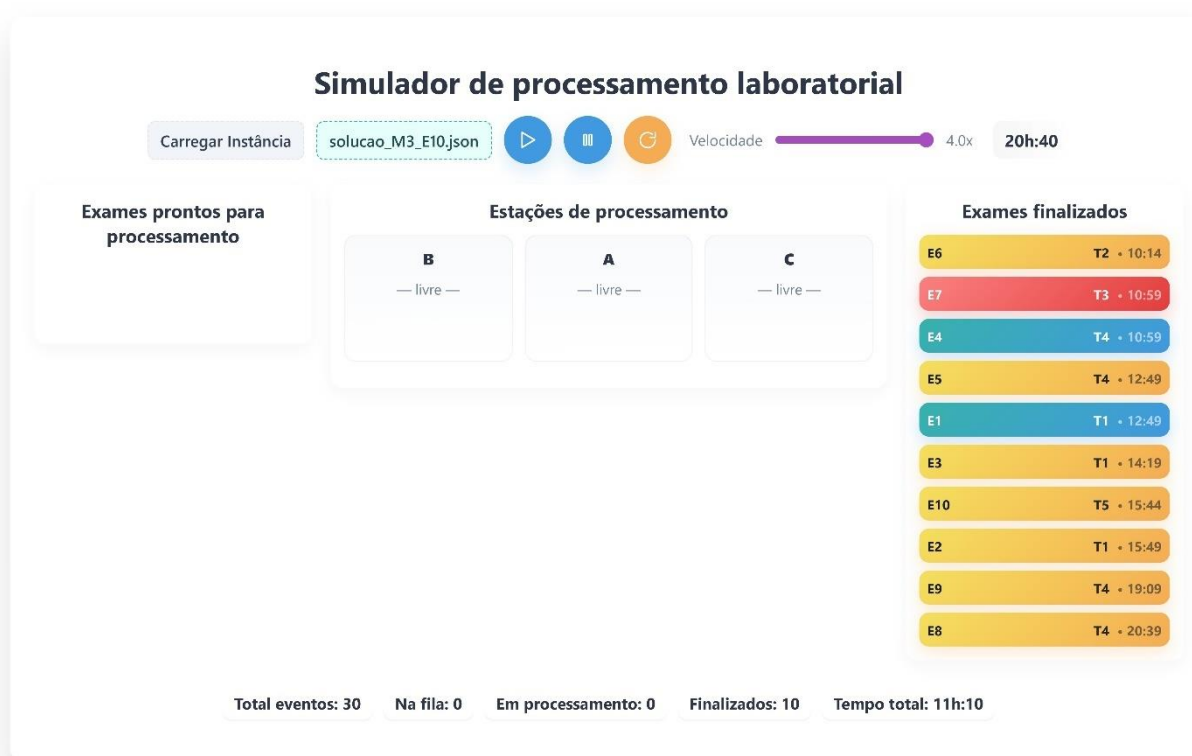
Figura 11 – Execução da simulação com alocação dinâmica dos exames às estações.



Fonte: Elaborada pelos autores (2026).

Por fim, a Figura 12 apresenta a interface ao término da execução da solução. Nesse momento, todos os exames foram processados e é possível visualizar o estado final do sistema, incluindo o tempo total de processamento obtido para o cenário analisado. Essa etapa permite confirmar a viabilidade da solução e validar se todas as restrições do problema foram respeitadas.

A utilização do *framework* como ferramenta de apoio visual contribui para a compreensão do comportamento das estratégias avaliadas, permitindo não apenas a análise quantitativa dos resultados, mas também a verificação qualitativa do fluxo operacional gerado por cada algoritmo. Dessa forma, o ambiente desenvolvido complementa a análise experimental apresentada nas seções anteriores, oferecendo uma visão integrada entre modelagem, execução e validação das soluções propostas.

Figura 12 – Interface do *framework* ao término da simulação da solução.

Fonte: Elaborada pelos autores (2026).

6 CONCLUSÃO

Com a elaboração deste trabalho, foi possível encontrar uma alternativa capaz de auxiliar especialistas na construção de soluções para o Problema de Escalonamento de Exames Hospitalares. Por meio da avaliação comparativa entre diferentes estratégias heurísticas, *Arrival Order Priority* (AOP), *Minimum Processing Time Priority* (MPTP) e *Roulette Selection Strategy* (RSS), identificou-se que a escolha da política de ordenação exerce impacto direto sobre o tempo total de processamento e a eficiência do fluxo de processamento, sendo o algoritmo AOP a alternativa que apresentou o melhor desempenho médio nos cenários analisados.

Durante a revisão da literatura, foi possível identificar os principais trabalhos relacionados ao escalonamento em contextos hospitalares, o que orientou a construção do *framework* de forma que ele fosse genérico e extensível. A arquitetura adotada permite a integração com outras técnicas de otimização, sejam elas heurísticas, meta-heurísticas ou métodos exatos, além de possibilitar a comunicação com sistemas legados eventualmente já em operação em ambientes hospitalares reais.

Adicionalmente, o desenvolvimento do *framework* possibilitou que os especialistas não apenas observem a solução final de forma intuitiva, mas também verifiquem o impacto das decisões de escalonamento ao longo de toda a execução do sistema. Através da reconstrução dinâmica do fluxo de processamento, o *framework* permite identificar gargalos, períodos de ociosidade das estações e a

evolução temporal do atendimento, contribuindo para uma compreensão mais aprofundada do comportamento das soluções geradas pelos algoritmos AOP, MPTP e RSS.

O *framework* também possibilita a integração com outros sistemas computacionais, uma vez que adota arquivos padronizados nos formatos texto (para as instâncias) e JSON (para as soluções). Essa padronização permite que dados provenientes de sistemas hospitalares legados sejam incorporados como novas instâncias a serem otimizadas, bem como que as soluções geradas neste trabalho possam ser estendidas ou reutilizadas em abordagens futuras que abordem variações do Problema de Escalonamento de Exames Hospitalares, como a incorporação de restrições adicionais.

Quanto às propostas de trabalhos futuros, pode-se citar:

- a integração do *framework* com outras técnicas de otimização, como algoritmos genéticos, *simulated annealing* ou programação inteira mista, visando explorar se tais abordagens podem produzir soluções de melhor qualidade que as heurísticas avaliadas neste trabalho;
- a realização de um estudo para verificar a experiência de usuário com especialistas da área de saúde, avaliando a usabilidade do *framework*, a clareza das visualizações e a utilidade prática das informações fornecidas para o apoio à tomada de decisão.

Em suma, este trabalho atingiu seu objetivo ao identificar alternativas viáveis para o escalonamento de exames hospitalares por meio dos algoritmos AOP, MPTP e RSS, e ao disponibilizar um *framework* que integra geração de soluções, visualização do fluxo e interoperabilidade com outros sistemas. Acredita-se que a abordagem desenvolvida possa contribuir para o apoio à tomada de decisão de especialistas, auxiliando na gestão do fluxo de amostras a serem processadas e na qualificação da assistência prestada em ambientes hospitalares.

REFERÊNCIAS

- ACKERMANN, Andres Eberhard Friedl. *Modelo de previsão de demanda do serviço de urgência em um hospital de pronto atendimento*. Dissertação (Mestrado em Engenharia de Produção e Sistemas) Universidade do Vale do Rio dos Sinos, São Leopoldo. 2020.
- CARDEAL, Lara Morena; PATERNO, Maria Beatriz Piccelli; CANTANE, Daniela Renata; DE OLIVEIRA, Rogério Antônio. Análise do comportamento estocástico de um modelo matemático aplicado à infecção hospitalar. *Revista Sistemática*, [S. l.], v. 16, n. 3, p. e9774, 2026. DOI: 10.56238/rcsv16n3-008.
- FERRARI, Thayse. *Abordagens multiobjetivo para o problema de localização de bases de veículos de resgate*. Tese (Doutorado em Engenharia de Transportes). Universidade Federal do Rio de Janeiro, Rio de Janeiro. 2022.
- FLÓREZ, Camilo Andrés Cáceres. *Proposta metodológica de otimização baseada em IA, ciência de dados e indústria 4.0. estudo de casos: serviços de emergência em e-hospital e sistema de gerenciamento de energias híbridas*. Tese (Doutorado em Engenharia Mecânica). Universidade Estadual de Campinas, Campinas. 2020.
- GAREY, M. R.; Johnson, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco: W. H. Freeman and Company. 1979.
- GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Boston: Addison-Wesley. 1989.
- HALL, R. W. (2013). *Patient Flow: Reducing Delay in Healthcare Delivery*. 2. ed. Southern California: Springer. *International Series in Operations Research & Management Science*, v. 206. 2013.
- KITCHENHAM, Barbara *et al.* *Guidelines for performing systematic literature reviews in software engineering*. 2007.
- LITVAK, E. *Managing Patient Flow in Hospitals: Strategies and Solutions*. 2. ed. Oakbrook Terrace: Joint Commission Resources. (2010).
- LOPES L., V., & MASTELLA, M. Processo de decisão: protocolo de tomada de decisão frente à Covid -19 no Brasil. *Revista Gestão Do Conhecimento E Tecnologia Da Informação*, 8(1), 61-74. 2025. <https://doi.org/10.31501/rgcti.v8i1.15310>.
- MEISTER, T. C. *Um algoritmo heurístico aplicado ao problema de escalonamento de médicos do Hospital de Clínicas de Porto Alegre*. Dissertação (Mestrado em Ciência da Computação) Universidade Federal do Rio Grande do Sul, Porto Alegre. 2020.
- MONDO, C. W. *Otimização da alocação de produtos em uma estrutura de armazenagem flow-rack utilizando programação linear inteira mista*. Dissertação (Mestrado em Engenharia Elétrica). Universidade Tecnológica Federal do Paraná, Ponta Grossa. 2024.
- PEREIRA, K. A. Rumo a uma maior eficiência na saúde: O impacto da digitalização de registros médicos. *Revista Sistemática*, 14(3), 709-714. 2024. <https://doi.org/10.56238/rcsv14n3-020>

PETERSEN, K.; FELDT, R.; MUJTABA, S.; MATTSSON, M. Systematic mapping studies in software engineering. *In 12th international conference on evaluation and assessment in software engineering (EASE)*. BCS Learning & Development. 2008.

REIS, E. R. D. S. *Alocação otimizada de voluntários em programa de restabelecimento de laços familiares por uma ONG Internacional*. Monografia (Bacharelado em Engenharia Ambiental). Universidade Estadual Paulista, São José dos Campos. 2024.

SILVA, H. L. R. D. *Abordagens heurísticas para planejamento e escalonamento de equipes médicas em cirurgias eletivas*. Monografia (Bacharelado em Engenharia de Produção). Universidade Federal de Campina Grande, Sumé. 2023.