UNIVERSIDADE DE SÃO PAULO
ESCOLA POLITÉCNICA

Beatriz Gaya Augutoli

TÉCNICAS DE PESQUISA OPERACIONAL APLICADAS AO PROBLEMA DE PROGRAMAÇÃO DAS ATIVIDADES DE LIMPEZA EM AMBIENTE HOSPITALAR

São Paulo
2025

Beatriz Gaya Augutoli – 12555798

TÉCNICAS DE PESQUISA OPERACIONAL APLICADAS AO PROBLEMA DE PROGRAMAÇÃO DAS ATIVIDADES DE LIMPEZA EM AMBIENTE HOSPITALAR

Trabalho de conclusão de curso para bacharelado em Engenharia de Produção apresentado ao Departamento de Engenharia de Produção (PRO) da Escola Politécnica da Universidade de São Paulo.

Orientadora: Profa. Dra. Débora Pretti Ronconi

VERSÃO ORIGINAL

São Paulo
2025

Catalogação-na-publicação

Augutoli, Beatriz
TÉCNICAS DE PESQUISA OPERACIONAL APLICADAS AO PROBLEMA DE PROGRAMAÇÃO DAS ATIVIDADES DE LIMPEZA EM AMBIENTE HOSPITALAR / B. Augutoli -- São Paulo, 2025.
78 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Produção.

1.Pesquisa operacional I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Produção II.t.

AGRADECIMENTOS

Agradeço, primeiramente, à minha orientadora, Prof.ª Dr.ª Débora Pretti Ronconi, pela dedicação, pela orientação segura e pelas contribuições fundamentais ao longo de todas as etapas deste trabalho. Sua experiência, rigor acadêmico e disponibilidade foram essenciais para o amadurecimento técnico desta pesquisa.

Ao Instituto CEMA, registro meu sincero agradecimento pela receptividade e pela colaboração institucional que possibilitou o acesso aos dados e o entendimento aprofundado da rotina operacional. À equipe de Hotelaria e aos profissionais do setor de Higienização, agradeço pela disponibilidade, pelo compartilhamento de informações e pela abertura ao diálogo, essenciais para que este trabalho refletisse de forma fiel a realidade prática da instituição.

Ao meu pai, Wilson Augutoli Junior, pelo apoio incondicional e por ter viabilizado o contato com o Instituto CEMA, contribuindo diretamente para que este projeto se tornasse possível. À minha mãe, Nívia Maria Gaya, e a toda a minha família, agradeço pelo suporte constante, não apenas durante a realização deste trabalho, mas ao longo de toda a minha trajetória universitária. A presença, o carinho e a força de vocês foram decisivos em todos os momentos dessa caminhada.

Ao meu namorado, Mateus de Pina, agradeço pelo companheirismo diário e pela valiosa colaboração na revisão deste trabalho, cuja dedicação contribuiu diretamente para sua qualidade final. Aos meus amigos e colegas, em especial Douglas Luchetti, Gabrielle Rodrigues e Isabela Ide, agradeço pela parceria, pelas conversas, pelo apoio e pela leveza que tornaram esta jornada muito mais significativa.

Por fim, deixo registrado meu agradecimento a todos que, direta ou indiretamente, contribuíram para este trabalho, seja por meio de suporte técnico, intelectual ou emocional. A cada um de vocês, meu sincero muito obrigada.

RESUMO

Augutoli, B. **Técnicas de pesquisa operacional aplicadas ao problema de programação das atividades de limpeza em ambiente hospitalar**. 2025. 78 p. Trabalho de Conclusão de Curso - Escola Politécnica, Universidade de São Paulo, São Paulo, 2025.

A higienização hospitalar é um elemento central para a segurança do paciente, a prevenção de infecções e a manutenção da continuidade assistencial, porém sua programação diária ainda é frequentemente conduzida de forma manual, sujeita a inconsistências, sobrecarga e baixa padronização. Este trabalho apresenta o desenvolvimento de um modelo de Programação Linear Inteira Mista e de uma heurística de decomposição voltados à organização das atividades de limpeza no Instituto CEMA, considerando a complexidade de diferentes tipos de tarefas, múltiplos regimes de turno e janelas rígidas. Embora o modelo exato represente adequadamente o problema, sua aplicação integral em escala real mostrou-se inviável, o que justificou a adoção de uma abordagem heurística estruturada em fases. A metodologia proposta foi capaz de gerar cronogramas factíveis em tempos compatíveis com o uso operacional, garantindo a execução de 100% das tarefas previstas no horizonte de 15 dias e promovendo uma redução próxima de 48% no desbalanceamento da carga entre profissionais, quando comparado ao processo manual.

Palavras-chave: Pesquisa operacional. Programação linear inteira mista. Escalonamento de pessoal. Higienização hospitalar.

ABSTRACT

Augutoli, B. **Operational research techniques applied to the scheduling problem of cleaning activities in hospital environments**. 2025. 78 p. Trabalho de Conclusão de Curso - Escola Politécnica, Universidade de São Paulo, São Paulo, 2025.

Hospital cleaning plays a fundamental role in patient safety, infection prevention, and the continuity of healthcare operations, yet its daily scheduling is often carried out manually, resulting in inconsistencies, overload, and limited standardization. This study presents the development of a Mixed-Integer Linear Programming model and a decomposition-based heuristic designed to organize cleaning activities at the CEMA Institute, taking into account the complexity arising from different types of tasks, multiple work-shift regimes, and strict execution windows. Although the exact model accurately represents the problem, its full-scale application proved computationally infeasible, which motivated the adoption of a multi-phase heuristic approach. The proposed methodology successfully generated feasible schedules within operationally acceptable runtimes, ensuring the execution of 100% of all tasks required over the 15-day planning horizon and achieving an approximate 48% reduction in workload imbalance among staff when compared with the manual process. These results suggest that optimization techniques can meaningfully support hospital cleaning management by improving operational organization, promoting fairer task distribution, and enhancing the reliability of daily routines.

Keywords: Operational research. Mixed-integer linear programming. Workforce scheduling. Hospital cleaning.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A introdução estabelece o contexto geral deste trabalho, apresentando a relevância da higienização hospitalar para a segurança do paciente, os desafios operacionais enfrentados pelo Instituto CEMA e a natureza complexa do problema de escalonamento enfrentado. Neste capítulo, discutem-se os elementos que motivaram o estudo, o escopo da pesquisa, os principais objetivos e a justificativa para a aplicação de técnicas de Pesquisa Operacional. O capítulo ainda contextualiza o ambiente hospitalar e evidencia a necessidade de soluções estruturadas e automatizadas para um problema hoje conduzido manualmente.

1.1 Contextualização: desafios da gestão em saúde e o papel crítico da higienização

A gestão de serviços de saúde representa um dos maiores desafios contemporâneos, pressionada por uma confluência de fatores demográficos, epidemiológicos e econômicos. O envelhecimento da população e o aumento da prevalência de doenças crônicas impulsionam uma demanda crescente e cada vez mais complexa por cuidados, enquanto os sistemas de saúde, sejam eles públicos ou privados, operam sob orçamentos progressivamente mais restritos. Neste cenário, a busca por eficiência operacional deixa de ser uma opção e torna-se um imperativo para a sobrevivência e sustentabilidade das instituições.

Dentro da complexa engrenagem hospitalar, onde múltiplos processos clínicos e de apoio ocorrem simultaneamente, a higienização de ambientes e superfícies emerge como um pilar fundamental e muitas vezes subestimado. Sua função transcende a mera hotelaria, posicionando-se como uma barreira de defesa essencial contra a proliferação de microrganismos e, consequentemente, a ocorrência de Infecções Relacionadas à Assistência à Saúde (IRAS).

As IRAS constituem um grave problema de saúde pública global. A Organização Mundial da Saúde (OMS) estima que, a qualquer momento, mais de 1,4 milhão de pessoas no mundo sofram de infecções adquiridas em hospitais. Em países desenvolvidos, estima-se que entre 5% e 10% dos pacientes admitidos em hospitais de cuidados agudos adquiram pelo menos uma IRAS, enquanto em países em desenvolvimento, como o Brasil, o risco é de 2 a 20 vezes maior, podendo afetar até 25% dos pacientes (WORLD HEALTH ORGANIZATION, 2011). Os dados nacionais mais recentes corroboram a gravidade do cenário. Segundo o boletim da Agência Nacional de Vigilância Sanitária (ANVISA, 2024), a densidade de incidência de infecções primárias de corrente sanguínea associadas a cateter venoso central em UTIs de adultos no Brasil foi de 2,91 por 1.000 cateteres-dia em 2023, um indicador que, apesar de demonstrar uma tendência de queda, ainda reflete um risco significativo à segurança do paciente. O impacto dessas infecções é devastador: aumento da morbimortalidade, prolongamento do tempo de internação, necessidade de uso de antimicrobianos mais potentes – o que acelera a resistência bacteriana – e, consequentemente, uma elevação substancial dos custos assistenciais.

A ligação entre a higienização ambiental e a incidência de IRAS é cientificamente bem estabelecida. O ambiente hospitalar atua como um reservatório contínuo de patógenos. Estudos demonstram a sobrevivência de microrganismos clinicamente relevantes em superfícies secas por meses, como *Enterococcus spp.* (incluindo VRE, resistente à vancomicina), *Staphylococcus aureus* (incluindo MRSA, resistente à meticilina) e *Clostridioides difficile* (KRAMER; SCHWEBKE; KAMPF, 2006). A transmissão ocorre quando profissionais de saúde ou pacientes tocam nessas superfícies contaminadas, transferindo os microrganismos para si ou para outros pacientes. Pesquisas evidenciam que a implementação de protocolos de limpeza e desinfecção mais rigorosos e monitorados resulta em uma redução significativa da contaminação ambiental e está associada a uma diminuição nas taxas de IRAS (DONSKEY, 2013). O próprio Manual de Hotelaria do Instituto CEMA, objeto deste estudo, reconhece este risco ao afirmar que "o ambiente em si é um importante reservatório de microrganismos, especialmente os “multirresistentes” (INSTITUTO CEMA, 2023).

Paralelamente ao impacto clínico, a higienização desempenha um papel crucial na percepção de qualidade pelo paciente. Para pacientes e seus familiares, que muitas vezes não possuem conhecimento técnico para avaliar a qualidade clínica do cuidado, o ambiente físico é um dos principais indicadores da qualidade geral do serviço. A limpeza do quarto, dos banheiros e das áreas comuns é um fator tangível e constantemente avaliado. Estudos que analisam dados de pesquisas de satisfação de pacientes, como o HCAHPS (Hospital Consumer Assessment of Healthcare Providers and Systems) nos Estados Unidos, consistentemente demonstram uma forte correlação positiva entre a percepção de limpeza do ambiente hospitalar e as notas de satisfação geral e a disposição do paciente em recomendar o hospital (GLICKMAN *et al.*, 2010; BOULDING *et al.*, 2011). Um ambiente percebido como sujo pode gerar desconfiança e ansiedade, minando a confiança do paciente na instituição, independentemente da excelência do cuidado clínico prestado.

Portanto, a gestão eficiente da equipe de higienização é uma atividade de missão crítica, com um duplo impacto estratégico: uma intervenção de segurança para a prevenção de danos clínicos (IRAS) e um componente essencial da experiência do paciente, influenciando diretamente a reputação e a competitividade da instituição hospitalar.

1.2 Objeto de estudo: o Instituto CEMA

O presente trabalho desenvolve-se a partir de um estudo de caso no Instituto CEMA, uma instituição privada de referência, localizada na cidade de São Paulo. Fundado em 1975, o hospital consolidou-se como um dos principais centros de excelência do país nas especialidades de Oftalmologia e Otorrinolaringologia, expandindo sua atuação para outras áreas, como Cirurgia de Cabeça e Pescoço e Fonoaudiologia. Sua reputação é construída sobre um corpo clínico qualificado e um alto volume de atendimentos, o que o torna um ambiente ideal para o estudo de problemas de gestão de operações em saúde.

A complexidade logística que motiva esta pesquisa é evidenciada pela escala de suas operações. A infraestrutura física do hospital, com 24.000 m² de área construída distribuída em sete andares, comporta uma capacidade instalada de 200 leitos de internação, 14 salas cirúrgicas, 81 consultórios e 21 salas de exames. O volume operacional reflete essa estrutura robusta: são realizadas, em média, 70 cirurgias por dia, com um fluxo diário de aproximadamente 4.000 pessoas, entre pacientes, acompanhantes e colaboradores. Este intenso fluxo de pessoas e procedimentos gera uma demanda contínua e complexa por serviços de apoio, dentre os quais a higienização é um dos mais críticos.

Atualmente, o serviço de higienização do hospital é garantido por uma equipe dedicada de 59 profissionais. A responsabilidade pela criação das escalas de trabalho e pela programação das tarefas de limpeza recai sobre a gestão de hotelaria hospitalar, que realiza este processo de forma manual. O acesso a dados primários e documentação oficial da instituição, como o Manual de Hotelaria e o Mapa de Criticidade de Limpeza Terminal, confere ao problema uma base concreta e alinhada à realidade operacional do hospital. É justamente neste contexto de alta demanda, grande complexidade de ambientes e um processo de agendamento manual que a oportunidade para a aplicação de métodos de otimização se torna evidente. O Instituto CEMA, portanto, não é apenas o cenário, mas o próprio laboratório para o desenvolvimento e a validação do modelo proposto, que busca transformar um processo empírico em um sistema otimizado e baseado em dados.

1.3 O custo da higienização ineficaz: impacto operacional e financeiro

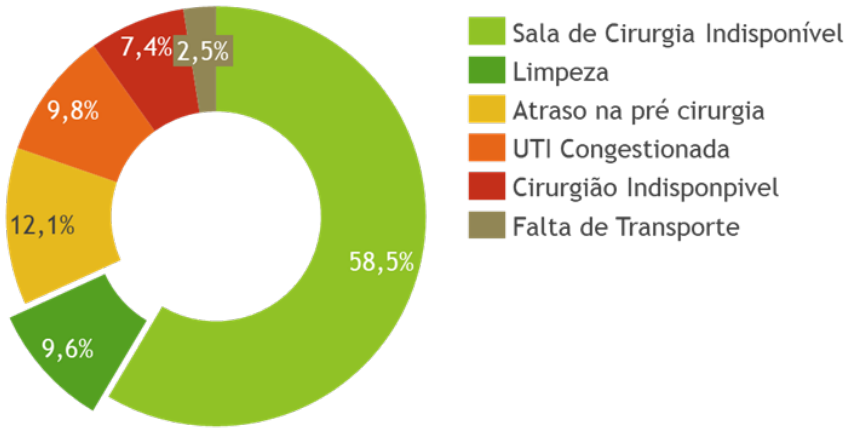
A criticidade da higienização não se limita ao controle de infecções; ela impacta diretamente a eficiência operacional e financeira do hospital. A falha no cumprimento de um cronograma de limpeza pode levar à indisponibilidade de ambientes essenciais, gerando um efeito cascata que paralisa atividades geradoras de receita. O estudo de Weinbroum, Ekstein e Ezri (2003) sobre a eficiência de centros cirúrgicos é particularmente elucidativo: os autores identificaram que a "indisponibilidade da sala de cirurgia" e a "limpeza" são responsáveis, juntas, por quase 68% dos atrasos no início dos procedimentos.

A Figura 1 apresenta a distribuição dos principais motivos de atraso em cirurgias eletivas. Observa-se que a "indisponibilidade da sala" e o processo de "limpeza" aparecem como causas dominantes, superando outras fontes como atrasos de equipe, problemas administrativos ou indisponibilidade de materiais. A predominância desses dois fatores evidencia que falhas na higienização têm impacto imediato no fluxo assistencial, pois impedem o início da primeira cirurgia – evento crítico que define o ritmo de todo o restante da agenda cirúrgica.

Essa indisponibilidade representa um custo de oportunidade tangível. Uma sala de cirurgia ociosa não apenas representa um ativo de alto valor subutilizado, mas também resulta na perda direta de receita associada aos procedimentos que deixaram de ser realizados. Além disso, atrasos na primeira cirurgia do dia se propagam para todas as subsequentes, mantendo equipes

médicas e de enfermagem em espera, elevando custos de mão de obra, aumentando a probabilidade de horas extras e, em casos mais graves, levando ao cancelamento de procedimentos. Esses dados reforçam que um cronograma de higienização ineficaz não é apenas um problema operacional, mas um fator estruturante do desempenho econômico e assistencial do hospital.

Figura 1 – Motivos de atraso nas cirurgias



Autoria própria, com base em Weinbroum, Ekstein e Ezri (2003)

1.4 O problema de escalonamento: complexidade e restrições

O escalonamento e a programação da equipe de higienização constituem um problema de otimização combinatória de alta complexidade. Este desafio envolve a alocação de recursos humanos limitados (a equipe de 59 profissionais do Instituto CEMA) para executar uma vasta gama de tarefas com diferentes requisitos, prioridades e restrições.

O arcabouço de restrições é multifacetado, abrangendo desde normas regulatórias até a logística operacional.

1. Restrições Legais e Sindicais: A alocação da equipe deve aderir estritamente a um corpo denso de regulamentações que ditam os limites e as condições do trabalho. A base deste arcabouço é a legislação trabalhista brasileira, em especial a Consolidação das Leis do Trabalho (CLT), que impõe restrições rígidas e inegociáveis. O Artigo 71 da CLT, por exemplo, estabelece a obrigatoriedade de um intervalo para repouso e alimentação (intrajornada) de, no mínimo, uma hora para qualquer trabalho contínuo cuja duração exceda seis horas. Esta pausa não é computada na duração do trabalho, devendo ser explicitamente modelada como um período de indisponibilidade do profissional. Outro pilar legal é o Artigo 59-A, inserido pela Reforma Trabalhista (Lei nº 13.467/2017), que faculta, mediante acordo individual escrito, convenção coletiva ou acordo coletivo de trabalho, o estabelecimento do regime de tempo parcial conhecido como escala 12x36. Esta escala,

amplamente utilizada em hospitais e descrita nos dados do CEMA, determina um ciclo fixo de 12 horas de trabalho por 36 horas ininterruptas de descanso, um padrão sequencial que define a disponibilidade do profissional de forma cíclica e previsível. Somam-se a este quadro as normas de segurança específicas do setor. A Norma Regulamentadora 32 (NR-32) estabelece as diretrizes para a segurança e saúde dos trabalhadores em serviços de saúde. Ela exige capacitação adequada, fornecimento e uso correto de Equipamentos de Proteção Individual (EPIs), e a implementação de protocolos de segurança para minimizar a exposição a riscos biológicos e químicos (BRASIL, 2022). Embora não dite regras de escalonamento, a NR-32 impõe uma restrição indireta crucial: o tempo alocado para cada tarefa deve ser suficiente para que ela seja executada de forma segura, sem pressa ou improviso que comprometa o uso de EPIs ou a correta aplicação de saneantes. Finalmente, o modelo deve considerar os acordos coletivos firmados por sindicatos representativos da categoria, como o SIEMACO-SP (Sindicato dos Trabalhadores em Empresas de Prestação de Serviços de Asseio e Conservação e Limpeza Urbana de São Paulo). As Convenções Coletivas de Trabalho (CCTs) negociadas por este sindicato introduzem especificidades sobre pisos salariais, adicionais (como o de insalubridade, que em grau máximo pode chegar a 40% do salário mínimo), regras para o pagamento de horas extras e outras condições que moldam não apenas a viabilidade de uma escala, mas também seu custo, tornando-se parâmetros fundamentais para a função objetivo de um modelo de otimização.

2. Restrições Operacionais: O hospital opera com três tipos de turno distintos, e a demanda por limpeza varia drasticamente com base na criticidade da área (Crítica, Semicrítica, Não Crítica) e na periodicidade das tarefas (Limpeza Terminal semanal, quinzenal ou mensal).

3. Natureza Estocástica: O problema ainda possui um componente de incerteza, representado pela demanda não programada por "Limpezas Imediatas", que exige que o cronograma possua um grau de flexibilidade e robustez para absorver eventos inesperados.

O processo de agendamento, atualmente manual, consome um tempo considerável da gestão e está sujeito a ineficiências. A origem da dificuldade está na própria estrutura combinatória do escalonamento: trata-se de selecionar, para cada tarefa, um subconjunto de profissionais elegíveis, obedecendo janelas temporais, cargas horárias máximas, requisitos de equipe simultânea, periodicidade das atividades e múltiplas restrições lógicas e operacionais. A combinação dessas decisões binárias gera um espaço de busca exponencial, análogo ao de problemas clássicos de *rostering*, *scheduling* e alocação de recursos com janelas de tempo – todos formalmente classificados como *NP-hard* segundo Garey e Johnson (GAREY; JOHNSON, 1979). Assim, a impossibilidade de avaliar todas as possibilidades em tempo hábil decorre da natureza intrínseca do problema, e não apenas da quantidade de trabalhadores envolvidos. Em consequência, métodos puramente exatos tornam-se impraticáveis para instâncias reais, justificando a adoção de heurísticas especializadas como alternativa computacionalmente viável.

Diante desses desafios, a aplicação de técnicas de Pesquisa Operacional torna-se plenamente justificada. A formulação de um modelo matemático permite representar o problema de maneira estruturada, incorporando explicitamente todas as suas restrições operacionais, lógicas e legais. A partir dessa representação formal, torna-se possível aplicar métodos de otimização capazes de buscar soluções que não apenas satisfaçam os múltiplos requisitos do sistema, mas também equilibrem objetivos potencialmente conflitantes, como cobertura, priorização e equidade entre profissionais. Essa abordagem supera substancialmente o planejamento manual, oferecendo decisões mais consistentes, reprodutíveis e alinhadas às necessidades estratégicas do serviço de higienização.

2 REVISÃO DA LITERATURA E FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta o arcabouço teórico que sustenta o desenvolvimento do modelo e da heurística propostos. Inicia-se com uma revisão das abordagens clássicas e modernas de *scheduling* em serviços, seguido de uma análise aprofundada do agendamento aplicado ao contexto hospitalar e da classificação formal do problema. São explorados trabalhos de referência, técnicas de otimização multicritério, métodos de decomposição e heurísticas relevantes, estabelecendo as bases conceituais necessárias para a formulação do problema. O objetivo é fornecer os elementos teóricos que fundamentam as decisões de modelagem adotadas posteriormente.

2.1 Programação de atividades em serviços: uma visão geral

A programação de atividades, ou *scheduling*, é uma área da Programação e Controle da Produção (PCP) dedicada à alocação de recursos escassos ao longo do tempo para a execução de um conjunto de tarefas. Embora suas origens estejam fortemente ligadas à otimização de processos de manufatura, sua aplicação expandiu-se massivamente para o setor de serviços, que hoje suporta o crescimento da economia global. O agendamento no setor de serviços difere fundamentalmente do agendamento na manufatura devido às características intrínsecas dos serviços: intangibilidade, perecibilidade, heterogeneidade e a coprodução com o cliente (PINEDO; ZACHARIAS; ZHU, 2015). A incapacidade de estocar um serviço (um leito de hospital vazio ou um assento de avião desocupado representam receita perdida para sempre) e a alta variabilidade da demanda tornam o *scheduling* uma ferramenta de gestão ainda mais crítica.

Pinedo, Zacharias e Zhu (2015) categorizam os problemas de agendamento em serviços em cinco grandes áreas: (*i*) agendamento de projetos, (*ii*) agendamento da força de trabalho (*workforce scheduling*), (*iii*) *timetabling*, reservas e agendamentos, (*iv*) agendamento de transportes e (*v*) agendamento no entretenimento. O problema em estudo se enquadra primariamente na categoria de agendamento da força de trabalho, que lida com a alocação de pessoal para atender a uma demanda flutuante. Esta categoria se subdivide em dois problemas clássicos:

1. *Shift Scheduling*: Focado em determinar quantos funcionários devem trabalhar em diferentes turnos (manhã, tarde, noite) para cobrir uma demanda que varia ao longo do dia, minimizando custos. É comum em *call centers* e varejo.

2. *Crew Scheduling*: Focado em construir sequências de tarefas (roteiros ou tours) para equipes (tripulações), como em companhias aéreas ou de transporte terrestre, de forma a cobrir todas as tarefas necessárias com o menor custo.

O problema do Hospital CEMA possui elementos de ambas as categorias. Diferentemente do *shift scheduling* clássico, em que os turnos representam blocos de trabalho a serem decididos

e alocados de acordo com a demanda temporal do serviço, no CEMA os turnos não pertencem às tarefas, mas aos próprios profissionais: cada colaborador já chega ao modelo com sua jornada definida de forma rígida (ADM, 12×36 diurno ou 12×36 noturno). Assim, não se decide "quantos funcionários devem trabalhar em cada turno", pois isso já está pré-estabelecido. Por outro lado, a necessidade de atribuir a cada profissional um conjunto de tarefas específicas para garantir a cobertura completa da higienização aproxima o problema do *crew scheduling*, em que cada trabalhador funciona analogamente a uma "tripulação" individual, e as tarefas de limpeza correspondem às "etapas" que devem ser distribuídas ao longo do horizonte.

Adicionalmente, o problema se posiciona entre agendamento estático e dinâmico. O agendamento estático refere-se a planos de longo prazo que não se espera que mudem (e.g., grade de voos de uma companhia aérea), permitindo um alto investimento computacional para encontrar a solução ótima. O agendamento dinâmico, por outro lado, lida com ambientes onde o cronograma muda frequentemente, exigindo soluções rápidas e robustas. O problema do CEMA é predominantemente estático no horizonte de um mês (o cronograma é gerado para o período inteiro), mas deve ser robusto o suficiente para acomodar os eventos dinâmicos e estocásticos das "Limpezas Imediatas".

2.2 Aprofundamento: *scheduling* em serviços hospitalares

O ambiente hospitalar é um dos cenários mais complexos para o *scheduling* de serviços. A interdependência entre os diversos departamentos (centro cirúrgico, UTI, enfermarias, laboratórios, higienização) cria uma rede complexa de restrições e consequências. A falha no agendamento de um serviço de apoio, como a limpeza, pode gerar um efeito cascata que paralisa atividades clínicas de alto valor, como demonstrado por Weinbroum, Ekstein e Ezri (2003).

A literatura de *scheduling* em saúde é vasta, mas historicamente focada em problemas de agendamento de recursos clínicos, como o *nurse rostering* (escalonamento de enfermeiros) e o *surgery scheduling* (agendamento de cirurgias). O *nurse rostering* é o mais análogo ao problema em estudo, pois lida com a criação de escalas para uma força de trabalho com múltiplos turnos, restrições legais e contratuais, e a necessidade de garantir a cobertura de pessoal para o cuidado ao paciente.

No entanto, o agendamento de serviços de apoio, como a higienização, possui particularidades. Enquanto a demanda por enfermagem está ligada ao número e à acuidade dos pacientes, a demanda por higienização é composta por uma parte determinística e periódica (limpezas concorrentes e terminais) e uma parte estocástica (limpezas de alta e imediatas). Essa estrutura de demanda mista é um desafio central.

A presente dissertação se posiciona na intersecção dessas áreas, aplicando a robustez teórica dos modelos de *nurse rostering* para tratar as complexas restrições de pessoal, ao mesmo tempo em que incorpora as particularidades da demanda de higienização hospitalar, contribuindo

para uma área de aplicação da Pesquisa Operacional que é crítica, mas comparativamente menos explorada.

A programação de equipes de serviços, especialmente em ambientes críticos como hospitais, tem sido extensivamente estudada na literatura de Pesquisa Operacional. Para contextualizar e fundamentar a abordagem de modelagem adotada neste trabalho, é essencial revisar os estudos que tratam de problemas com estrutura similar. A vasta gama de particularidades encontradas em problemas de escalonamento de pessoal (*personnel rostering*) torna a comparação direta entre diferentes abordagens um desafio. Para superar essa barreira, a comunidade acadêmica tem buscado sistemas de classificação que permitam a padronização da descrição dos problemas.

Uma das mais bem-sucedidas ferramentas de classificação surgiu na área de programação da produção, com a notação de três campos $\alpha \mid \beta \mid \gamma$ proposta por Graham *et al.* (1979). Este sistema permitiu organizar a pesquisa de problemas de escalonamento, facilitando a identificação da complexidade e a recomendação de métodos de solução. Reconhecendo a necessidade de uma ferramenta similar para a área de serviços, Causmaecker e Berghe (2011) propuseram uma adaptação desta notação especificamente para o problema de escalonamento de enfermeiros. Essa notação se torna aplicável ao problema, pois, como abordado, o escalonamento da equipe de higienização hospitalar compartilha muitas das complexidades do *nurse rostering*.

Tipos de Limpezas hospitalares

A rotina de higienização hospitalar compreende diferentes modalidades de limpeza, cada uma definida segundo normas e diretrizes operacionais da instituição e associada a objetivos específicos no controle de infecções e na manutenção da ambiência. A limpeza concorrente é o procedimento realizado de forma geral e cotidiana, englobando a higienização de pisos, instalações sanitárias, mobiliário e superfícies horizontais. Por sua frequência diária – e, em muitos setores, executada sempre que necessário – essa modalidade constitui a base da manutenção contínua do ambiente, garantindo a remoção regular de sujidades e reduzindo o acúmulo de matéria orgânica, que favorece a proliferação de microrganismos. Em termos operacionais, é também o tipo de limpeza que mais demanda tempo dos profissionais, influenciando diretamente a distribuição de carga de trabalho tratada pelo modelo de otimização desenvolvido neste estudo.

A limpeza terminal, por sua vez, corresponde a um procedimento profundo e abrangente, que envolve a higienização horizontal e vertical do ambiente, incluindo pisos, paredes, portas, janelas, mobiliário, equipamentos, camas, colchões, luminárias, superfícies internas e externas, além de elementos estruturais como tetos e grades de ventilação. Sua periodicidade é definida conforme o nível de criticidade da área: diariamente e sempre que necessário em setores críticos; semanalmente em setores semicríticos; e em intervalos mais espaçados em áreas não críticas. A Figura 2 ilustra essa distribuição por criticidade, destacando como a intensidade e a frequência das limpezas terminais variam em função do risco ocupacional e assistencial associado a cada ambiente. Esse tipo de limpeza representa um dos principais pontos de concentração de esforço no planejamento, pois envolve tempos de execução elevados e, frequentemente, a necessidade

de equipes simultâneas – aspectos que tornam sua modelagem particularmente relevante no contexto deste trabalho.

Figura 2 – Mapa de limpeza terminal por criticidade das áreas

MAPA DE LIMPEZA TERMINAL POR CRITICIDADE DAS ÁREAS					
ÁREAS CRITICAS	PERIODICIDADE	SEMI CRITICAS	PERIODICIDADE	NÃO CRITICAS	PERIODICIDADE
CENTRO CIRURGICO	SEMANAL	APARTAMENTOS / ENFERMARIAS	QUINZENAL	SALAS ADMINISTRATIVAS	MENSAL
PEQ. CIRÚRGIAS	SEMANAL	BANHEIROS	QUINZENAL	GUARITA	MENSAL
CME	SEMANAL	POSTOS DE ENFERMAGEM	QUINZENAL	ÁREAS COMUNS	MENSAL
PRONTO ATENDIMENTO	SEMANAL	CONSULTÓRIOS	QUINZENAL	CORREDORES	MENSAL
CAMARA FRIA	SEMANAL	ROUPARIA - ÁREA LIMPA	QUINZENAL	ALMOXARIFADO	MENSAL
EXPURGOS	SEMANAL	ELEVADORES	QUINZENAL	ELEVADORES	MENSAL
FARMÁCIA	SEMANAL	CORREDORES (CIRCULAÇÃO)	QUINZENAL	SALAS DE ESPERA	MENSAL
MED. TRABALHO	SEMANAL				
LEITOS DE ISOLAMENTO	ALTA PACIENTE				

Autoria própria, com base em material fornecido pela instituição

Por fim, a limpeza imediata – também denominada descontaminação – é realizada sempre que ocorre sujidade inesperada em áreas críticas ou semicríticas, geralmente associada a materiais orgânicos e químicos ou a situações que elevam o risco de disseminação microbiológica. Seu objetivo é remover prontamente o material contaminante e restaurar as condições seguras do ambiente, restringindo-se ao local da ocorrência e empregando técnicas adequadas ao tipo de sujidade. Essa modalidade possui natureza reativa e imprevisível, o que a torna incompatível com uma abordagem determinística de planejamento. Por essa razão, apesar de fazer parte da operação real, ela não é diretamente incorporada ao modelo matemático, mas reforça a importância de soluções que preservem margens de flexibilidade operacional para acomodar eventos não programados sem comprometer a rotina prevista de higienização.

2.3 Classificação do problema

Utilizando a notação de Causmaecker e Berghe (2011), o problema de higienização do Hospital CEMA foi classificado como ASB | V3 | LPRGM. A seguir, são detalhados cada um dos componentes desta classificação, estabelecendo uma ponte entre a teoria e a realidade operacional do hospital.

Campo α: ambiente do pessoal

O primeiro campo descreve as características e restrições inerentes à força de trabalho. Ele captura as regras que governam como os indivíduos podem ser alocados, refletindo desde regulamentações legais até políticas internas da organização.

- A (*Availability* – Disponibilidade): Este componente refere-se às restrições fundamentais que definem quando um profissional pode ou não trabalhar. No problema em questão, a disponibilidade não é flexível, mas sim um dado de entrada crucial. Ela é determinada por um conjunto de regras rígidas: *(i)* as leis trabalhistas brasileiras (CLT), que ditam os limites de jornada e os intervalos obrigatórios; *(ii)* os contratos de trabalho individuais, que especificam a carga horária mensal; e *(iii)* as escalas de trabalho fixas (Diurno A, Diurno 12x36, Noturno 12x36), que pré-determinam os dias de trabalho e folga para cada profissional. A modelagem deve, portanto, tratar a disponibilidade como um parâmetro inviolável.

- S (*Sequences* – Sequências): Este componente captura as regras sobre padrões de trabalho e descanso consecutivos. A característica mais proeminente no Instituto CEMA é a escala 12x36, que impõe um ciclo rígido de 12 horas de trabalho seguidas por 36 horas de descanso. Esta é uma restrição de sequência explícita que domina o planejamento para uma parcela significativa da equipe e deve ser refletida na matriz de disponibilidade dos profissionais. A literatura de *nurse rostering* frequentemente lida com restrições de sequência mais complexas (e.g., número máximo de dias de trabalho consecutivos, proibição de certas sequências de turnos), mas a natureza fixa das escalas no problema CEMA simplifica este aspecto, transformando-o em um padrão pré-definido.

- B (*Balance* – Balanceamento): Indica a necessidade de distribuir a carga de trabalho de forma equitativa entre os membros da equipe. Este é um dos objetivos explícitos do projeto, visando evitar a sobrecarga de alguns profissionais e a ociosidade de outros, o que impacta diretamente a satisfação, a fadiga e, consequentemente, a qualidade e segurança do serviço prestado. No modelo, isso se traduzirá em um objetivo ou restrição para minimizar a variância das horas de tarefas atribuídas entre os profissionais.

Campo β: características do trabalho

O segundo campo descreve a natureza da demanda de trabalho e a estrutura dos turnos.

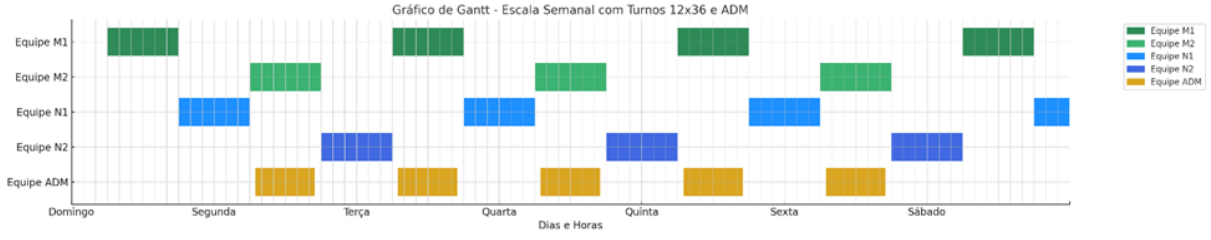
- V (*Fluctuating* – Flutuante): Este atributo é central para o problema e indica que a demanda por serviços não é constante ao longo do horizonte de planejamento. No caso do CEMA, a demanda por limpeza é altamente flutuante, ditada pela periodicidade das tarefas de Limpeza Terminal. A necessidade de realizar esta tarefa intensiva em áreas críticas semanalmente, em áreas semicríticas quinzenalmente e em áreas não críticas mensalmente cria picos de demanda em dias específicos do mês, como observado na Figura 2 . Esta variação temporal é uma fonte primária da complexidade combinatória do problema, pois a alocação de recursos deve ser capaz de absorver esses picos sem violar as restrições de capacidade.

Este número representa a quantidade de tipos de turnos distintos existentes. No Instituto CEMA, a equipe de higienização opera sob três regimes principais de jornada: o Turno Administrativo (7h–17h), o Turno Diurno 12x36 (6h–18h) e o Turno Noturno 12x36 (18h–06h). Essa heterogeneidade da força de trabalho – tanto na duração diária quanto no padrão de repetição das jornadas – exerce impacto direto sobre a capacidade operacional disponível a cada dia e precisa ser explicitamente incorporada ao modelo de programação. Em outras palavras, a quantidade efetiva de horas disponíveis em um determinado período não é constante, mas depende de quais tipos de turno estão ativos naquele dia e de sua interação com as janelas de execução das tarefas.

A Figura 3 ilustra essa estrutura por meio de um gráfico de Gantt que representa a escala semanal dos diferentes tipos de turno. Observa-se a alternância entre turnos longos (como os ciclos 12x36) e turnos convencionais (ADM), bem como a sobreposição parcial entre eles ao longo da semana. Esse padrão revela que a disponibilidade diária não é uniforme: alguns dias concentram mais profissionais simultaneamente (particularmente quando turnos ADM e diurnos coincidem), enquanto outros apresentam menor capacidade por depender majoritariamente dos turnos noturnos. Além disso, nota-se o efeito das folgas inerentes ao regime 12x36, gerando oscilações regulares que impactam o dimensionamento de tarefas distribuídas ao longo dos dias.

Do ponto de vista analítico, o gráfico evidencia dois desafios relevantes para o modelo de otimização. Primeiro, a capacidade horária total por dia é variável e dependente da combinação de turnos presentes, exigindo que o modelo respeite essa flutuação ao alocar tarefas longas e curtas. Segundo, a janela útil disponível para tarefas de longa duração é diferente entre turnos: o ADM, por exemplo, tem um bloco contínuo de 10 horas, enquanto os turnos 12x36, apesar de longos, estão sujeitos a limitações de início e fim que precisam ser compatibilizadas com janelas de higienização. Assim, o gráfico não apenas ilustra a escala real, mas explicita a natureza heterogênea da disponibilidade operacional – elemento central para o comportamento das soluções geradas.

Figura 3 – Gráfico de Gantt de escala de turnos semanais



Autoria própria, com base em material fornecido pela instituição

Campo γ: objetivos da otimização

O terceiro campo especifica os critérios que serão considerados para a resolução do problema, refletindo as prioridades da gestão hospitalar.

- L (*Coverage* – Cobertura): Indica que o objetivo central e mais crítico é garantir a cobertura

da demanda, ou seja, a realização de todas as tarefas de limpeza mandatórias. Em um ambiente hospitalar, a falha na cobertura não é apenas uma ineficiência operacional, mas um risco direto à segurança do paciente.

- P (*Personnel* – Pessoal): Refere-se a objetivos relacionados às preferências ou bem-estar dos funcionários. Neste caso, o principal objetivo desta categoria é o balanceamento da carga de trabalho (vinculado ao B do campo α), que visa a equidade e a sustentabilidade do desempenho da equipe.

- R (*Robustness* – Robustez): Refere-se à capacidade do cronograma gerado de manter sua qualidade mesmo diante de pequenas variações operacionais, tais como mudanças pontuais de demanda, redistribuições internas ou oscilações na carga de trabalho diária. No contexto do CEMA, a robustez não implica tratar explicitamente elementos estocásticos, mas sim estruturar decisões que reduzam a sensibilidade do plano a flutuações usuais do ambiente hospitalar. Uma formulação robusta tende a evitar alocações excessivamente concentradas em poucos profissionais, a depender de combinações frágeis de janelas e a produzir escalas que permaneçam viáveis sob ajustes mínimos. Assim, o conceito de robustez é incorporado não pela modelagem da incerteza, mas pela busca de soluções estáveis, equilibradas e menos suscetíveis a degradação quando submetidas à dinâmica real de operação.

- G (*General* – Geral): É uma categoria que engloba outros objetivos importantes não cobertos pelas demais letras, como a minimização da ociosidade da equipe e a maximização da eficiência operacional geral, garantindo que o tempo de trabalho remunerado seja utilizado da forma mais produtiva possível.

- M (*Multi-objective* – Multiobjetivo): A coexistência simultânea de diferentes metas – como garantir cobertura completa das tarefas (L), atender prioridades associadas às áreas críticas (P), preservar certo nível de robustez frente às variações operacionais (R) e promover equilíbrio na distribuição de carga entre os profissionais (G) – caracteriza o problema como inerentemente multiobjetivo. Esses objetivos, por natureza, apresentam tensões internas: maximizar a cobertura pode aumentar o desequilíbrio entre profissionais; priorizar áreas críticas pode reduzir a flexibilidade operacional; aumentar a robustez pode diminuir a eficiência em termos de utilização da capacidade. Assim, a estrutura do problema não admite um único estado que satisfaça plenamente todos os critérios ao mesmo tempo, uma vez que eles competem entre si. O resultado é um cenário em que compromissos são inevitáveis, e o problema deve ser entendido como uma busca por equilíbrio entre objetivos conflitantes.

2.4 Análise de trabalhos de referência

A classificação ASB | V3 | LPRGM permite filtrar a vasta literatura e focar nos trabalhos mais pertinentes para a construção do modelo matemático. No entanto, observa-se que praticamente todos os estudos encontrados com estrutura semelhante pertencem ao domínio do *nurse rostering*. Embora essa literatura seja madura e extensa, os trabalhos específicos voltados à alocação de profissionais de limpeza hospitalar são escassos, refletindo uma lacuna no campo de Pesquisa Operacional aplicada a serviços de apoio. Diante disso, este capítulo estabelece um paralelo entre os problemas clássicos analisados e o problema particular de higienização hospitalar abordado neste trabalho. Tal aproximação é possível porque ambos compartilham características estruturais centrais; assim, ainda que o contexto operacional seja distinto, as técnicas e modelagens discutidas na literatura fornecem um arcabouço conceitual robusto para fundamentar as decisões metodológicas empregadas neste estudo.

Azaiez e Sharif (2005) apresentam um modelo de *Goal Programming* para o escalonamento de enfermeiros, classificado como ASBN | V2 | PL. Sua principal contribuição é a capacidade de lidar com uma demanda flutuante (V) ao longo do tempo, similar à demanda de limpeza terminal com periodicidades semanais e quinzenais. O modelo visa garantir a cobertura mínima de pessoal em cada turno. Para o presente trabalho, essa lógica é adaptada: como as escalas dos profissionais são fixas, a decisão não é quantos profissionais alocar, mas se uma tarefa obrigatória será executada. Assim, a variável de decisão sobre a conclusão de uma tarefa (y_{ajk}) se torna a "meta" (*goal*) a ser atingida, e a restrição de cobertura garante que essa meta só seja cumprida se um profissional for efetivamente alocado.

Duenas, Tütüncü e Chilcott (2009) abordam o problema de escalonamento de enfermeiros tratando-o explicitamente como multiobjetivo (M), classificação AS | R3 | PLM. Os autores utilizam um Algoritmo Genético para encontrar soluções de compromisso (*trade-off*) entre objetivos conflitantes, como a cobertura da demanda e as preferências dos funcionários. Essa abordagem valida a estrutura da função objetivo, que também busca balancear múltiplos critérios.

O balanceamento da carga de trabalho (B) é uma característica central deste problema. Bilgin *et al.* (2012) diz que em um problema classificado como ASBI | RVNO | PLR, o balanceamento pode ser tratado como uma restrição rígida (*hard constraint*) ou flexível (*soft constraint*). Inspirados por essa dualidade, o balanceamento é modelado como uma restrição flexível, onde o desvio da carga de trabalho de cada profissional em relação à média é medido e penalizado na função objetivo. Esta abordagem evita a inviabilidade do modelo, que poderia ocorrer com uma restrição rígida, e busca a solução mais equitativa possível sem comprometer o objetivo primário de cobrir as tarefas críticas.

Por fim, a priorização de tarefas é informada pela lógica de ponderação diferencial, como visto em Dias *et al.* (2003). Neste estudo sobre escalonamento em um hospital brasileiro, os autores atribuem pesos de penalidade distintos a diferentes tipos de violações. Este conceito é

adaptado de forma positiva: ao invés de penalizar falhas, a conclusão de tarefas é recompensada com pesos (P_{ak}) que são ordens de magnitude maiores para áreas críticas. Isso força o modelo a alocar seus recursos escassos (horas de trabalho) prioritariamente às tarefas de maior impacto para a segurança do paciente.

2.5 Abordagens de otimização multiobjetivo

Problemas de otimização que envolvem mais de um critério de desempenho são denominados multiobjetivos ou multicritérios. Nesses casos, a busca por uma única solução ótima global, como ocorre em problemas de objetivo único, torna-se inviável, pois frequentemente os objetivos apresentam naturezas conflitantes. Assim, a solução depende das preferências do tomador de decisão e da forma como esses objetivos são priorizados ou combinados (PINEDO, 2016).

De acordo com Coello (1999), as principais estratégias para tratar problemas multiobjetivo podem ser agrupadas em três categorias: *(i)* métodos de soma ponderada, *(ii)* métodos baseados na fronteira de Pareto e *(iii)* métodos lexicográficos. Cada uma dessas abordagens reflete uma maneira distinta de representar a importância relativa entre os objetivos e guiar o processo de busca da solução.

No método da soma ponderada, atribuem-se pesos a cada objetivo, transformando o problema multiobjetivo em um problema escalar de objetivo único. Essa técnica é simples de implementar, mas depende fortemente da escolha dos pesos, que pode ser subjetiva e alterar significativamente o resultado obtido. Além disso, a combinação linear pode falhar em identificar soluções localizadas em regiões não convexas da fronteira de Pareto (COELLO, 1999).

A fronteira de Pareto, por sua vez, representa o conjunto de soluções não dominadas, nas quais a melhoria de um objetivo implica a piora de outro. Essa abordagem é amplamente utilizada em problemas de programação da produção, pois fornece um panorama das alternativas possíveis, permitindo ao decisor escolher a solução mais adequada às suas preferências (FANG *et al.*, 2013; MANSOURI; AKTAS; BESIKCI, 2016). No entanto, a obtenção completa da fronteira de Pareto pode demandar alto custo computacional, sobretudo em problemas de grande escala.

A otimização lexicográfica, proposta formalmente por Isermann (1982), organiza os objetivos segundo uma hierarquia de prioridade. Nesse método, o primeiro objetivo é otimizado de forma absoluta; somente após atingir seu valor ótimo busca-se otimizar o segundo, mantendo o primeiro fixo ou dentro de uma margem aceitável. Esse processo é repetido até que todos os objetivos sejam tratados. A principal vantagem dessa abordagem é a clareza na definição das prioridades, sendo especialmente adequada em contextos em que um critério possui importância claramente superior aos demais. Por outro lado, sua rigidez impede compensações entre objetivos, o que pode levar à desconsideração de soluções com melhor equilíbrio global (PINEDO, 2016).

No contexto da programação da produção, a adoção de abordagens multiobjetivo é

particularmente relevante, uma vez que diferentes indicadores de desempenho frequentemente apresentam relações conflitantes. Assim, a escolha da abordagem adequada deve considerar tanto o grau de prioridade entre os objetivos quanto a necessidade de representatividade das soluções viáveis (COELLO, 1999; ISERMANN, 1982; PINEDO, 2016).

Em síntese, as abordagens multiobjetivo constituem um arcabouço essencial para a modelagem de sistemas complexos, pois permitem representar simultaneamente critérios que frequentemente se encontram em tensão, como cobertura de demanda, equidade na distribuição de carga de trabalho e estabilidade operacional. Neste trabalho, adotou-se a abordagem lexicográfica por sua adequação a contextos em que existe uma clara hierarquia entre metas. Em particular, a garantia da execução das tarefas críticas de higienização assume prioridade absoluta, enquanto o equilíbrio da carga de trabalho entre os profissionais e a preservação de margens de robustez operam como objetivos secundários. Essa estrutura hierárquica assegura coerência entre a formulação matemática e as prioridades operacionais do ambiente hospitalar analisado, refletindo fielmente as exigências práticas do sistema real.

2.6 Heurísticas de decomposição

A crescente complexidade dos problemas de escalonamento e alocação de recursos humanos tem motivado o desenvolvimento de métodos heurísticos e metaheurísticos capazes de fornecer soluções de alta qualidade em tempo computacional reduzido. Muitos desses problemas pertencem à classe dos problemas *NP-hard*, caracterizada pela inexistência de algoritmos conhecidos que os resolvam exatamente em tempo polinomial à medida que o tamanho da instância cresce. Em termos práticos, isso significa que o esforço computacional necessário para obter soluções ótimas cresce de forma exponencial, tornando inviável o uso exclusivo de métodos exatos em cenários reais de grande porte (PINEDO, 2016).

Dessa forma, a necessidade de respostas rápidas e operacionalmente viáveis justifica o emprego de estratégias aproximadas, como heurísticas de decomposição e abordagens em duas fases. Tais métodos estruturam o problema em subproblemas menores e mais tratáveis, permitindo que decisões de natureza estratégica e tática sejam tratadas de forma hierarquizada, ao mesmo tempo em que se preserva a qualidade global das soluções obtidas.

Segundo Pinedo (2016), as heurísticas de decomposição representam uma das classes mais relevantes de técnicas para o tratamento de problemas complexos de programação de tarefas. Essas abordagens subdividem o problema original de forma sistemática – seja por recurso, por tarefa ou por intervalo temporal – de modo a reduzir a dimensionalidade combinatória e facilitar a obtenção de soluções aproximadas. O autor distingue quatro categorias principais de decomposição: *(i)* decomposição por máquina, em que subproblemas são resolvidos sequencialmente para cada recurso, como no método *shifting bottleneck*; *(ii)* decomposição por trabalho (*job-based*), que resolve iterativamente subconjuntos de operações associadas a um mesmo trabalho; *(iii)* decomposição temporal (*rolling horizon*), que limita o horizonte de planejamento

a blocos sucessivos de tempo; e *(iv)* decomposição híbrida, que combina mais de um critério de partição, explorando complementaridades entre os níveis de decisão. A eficácia dessas estratégias depende fortemente da estrutura de controle adotada, isto é, do modo como os subproblemas são resolvidos e reotimizados iterativamente, de forma a garantir consistência global da solução.

Além disso, Pinedo (2016) enfatiza o papel das relaxações lineares como base para heurísticas construtivas. Em termos gerais, uma relaxação linear consiste em remover ou flexibilizar as restrições de integralidade de um modelo de Programação Linear Inteira Mista (PLIM), permitindo que variáveis originalmente binárias ou inteiras assumam valores contínuos dentro do intervalo permitido. Essa simplificação reduz drasticamente a complexidade do problema, viabilizando a obtenção de soluções ótimas para a versão relaxada em tempos computacionais muito inferiores. Segundo Pinedo, tais relaxações são fundamentais porque fornecem limites inferiores (*lower bounds*) para o problema original e permitem derivar soluções iniciais factíveis por meio de estratégias de arredondamento ou reconstrução heurística. Essa filosofia permeia grande parte das abordagens modernas, nas quais modelos exatos e heurísticos são combinados em estruturas hierárquicas de otimização, aproveitando simultaneamente a força preditiva das relaxações lineares e a flexibilidade operacional das heurísticas.

Heurística de duas fases para atribuição flexível de atividades e tarefas

O estudo de Elahipanah, Desaulniers e Lacasse-Guay (2013) constitui uma referência seminal na aplicação de heurísticas de decomposição em duas fases a problemas de alocação flexível de tarefas e atividades. Os autores abordam o *Flexible Activity and Task Assignment Problem* (FATAP), típico de setores de serviços, no qual é necessário compatibilizar a disponibilidade de empregados, suas habilidades e um conjunto de tarefas indivisíveis com janelas temporais específicas. O objetivo é determinar uma programação factível que minimize o custo de operação e maximize a cobertura de tarefas, considerando extensões de turno, movimentação de intervalos (*breaks*) e contratação de trabalhadores temporários.

O problema é formulado como um modelo de Programação Linear Inteira Mista de grande porte, cuja resolução direta é impraticável em instâncias reais. Para contornar essa limitação, os autores propõem uma heurística de duas fases, que constitui uma decomposição hierárquica entre decisões estruturais e táticas:

Fase 1 – Modelo aproximado (PLIM simplificado): Nesta etapa, define-se uma alocação inicial de tarefas e atividades utilizando um subconjunto reduzido de restrições. São considerados apenas os aspectos estruturais do problema, como a atribuição de tarefas de longa duração e a definição de turnos candidatos, enquanto restrições relacionadas à movimentação de intervalos, continuidade de atividades e custos de transição são relaxadas. Essa simplificação permite a rápida identificação de uma solução-base viável e a geração de um conjunto restrito de candidatos a extensões de turno e contratações temporárias.

Fase 2 – Modelo de Refinamento por Geração de Colunas: Com base nas decisões

estruturais obtidas na Fase 1, a segunda etapa realiza o refinamento das alocações por meio de um modelo mestre responsável pela seleção de combinações viáveis de atividades e períodos para cada funcionário ao longo do horizonte de planejamento. Essa etapa utiliza o método de geração de colunas (*column generation*), técnica amplamente empregada em problemas de grande escala em que o número de possíveis turnos ou padrões de trabalho é demasiado elevado para ser enumerado integralmente. Em vez de listar antecipadamente todos os turnos candidatos, o modelo mestre trabalha inicialmente com um conjunto reduzido de colunas e, iterativamente, subproblemas especializados avaliam a possibilidade de gerar novos turnos com custo reduzido. Quando encontrados, esses turnos são adicionados ao modelo mestre, ampliando gradualmente o espaço de decisão. Esse processo evita a explosão combinatória e garante que apenas padrões relevantes sejam considerados. Além disso, a fase incorpora um procedimento de horizonte rolante, no qual subintervalos do planejamento são resolvidos sequencialmente, assegurando coerência temporal e continuidade das decisões entre blocos consecutivos.

A formulação de referência utiliza os conjuntos de períodos P, atividades A com demandas n_{ap}, turnos regulares H e suas extensões P_h^R e P_h^{Oi}, turnos temporários L, e tarefas T com conjuntos de inícios admissíveis F_t. A estrutura da função objetivo e das restrições do modelo mestre pode ser expressa de forma simplificada por:

$$\min \sum_{h \in H} c_h x_h + \sum_{l \in L} c_l y_l \tag{2.1}$$

sujeito a:

$$\sum_{h \in H} a_{hp} x_h + \sum_{l \in L} a_{lp} y_l \geq n_{ap}, \quad \forall a \in A,\, p \in P, \quad x_h,\, y_l \in \{0{,}1\}, \quad \forall h \in H,\, l \in L. \tag{2.2}$$

Nesse modelo, x_h e y_l representam a ativação de turnos regulares e temporários, respectivamente, enquanto a_{hp} e a_{lp} indicam se o turno h ou l é capaz de atender a atividade a no período p. Os custos c_h e c_l refletem o esforço, duração ou características operacionais associadas a cada turno. As restrições de cobertura garantem que a soma das contribuições dos turnos selecionados atenda às demandas previstas. Dessa forma, cada coluna corresponde a um padrão completo de trabalho e o problema mestre seleciona aqueles que satisfazem as necessidades operacionais com o menor custo possível.

Heurísticas baseadas em cenários para turnos estocásticos com demanda por tarefas

O trabalho de Wu *et al.* (2023) amplia a discussão sobre heurísticas de decomposição ao considerar ambientes estocásticos, nos quais a demanda por tarefas e a duração efetiva das atividades são incertas. Os autores estudam o *Stochastic Shift Design Problem with Task-Based Demand*, no qual o objetivo é projetar turnos de trabalho robustos frente à variabilidade de

demanda, mantendo a probabilidade de cobertura de todas as tarefas acima de um nível mínimo predefinido.

A formulação do problema é estruturada como um modelo de otimização estocástica com restrição de probabilidade (*chance-constrained programming*), representado por:

$$\min \mathbb{E}[C(x,\xi)] \quad \text{sujeito a} \quad \Pr\{g(x,\xi) \leq 0\} \geq \alpha, \tag{2.3}$$

em que x representa as decisões de alocação de turnos, ξ denota os parâmetros aleatórios associados à demanda e à duração das tarefas, e α é o nível de confiabilidade desejado.

O modelo busca minimizar o custo esperado $C(x,\xi)$, garantindo que a restrição operacional $g(x,\xi)$ seja satisfeita em pelo menos uma fração α dos cenários.

Para resolver o problema, os autores propõem duas heurísticas complementares: *(i)* de único estágio, que aumenta conservadoramente as durações das tarefas (*buffers*) para absorver incertezas; e *(ii)* de dois estágios com evolução de cenários, que alterna entre geração de soluções candidatas e avaliação probabilística, refinando iterativamente a cobertura das tarefas e o custo.

Os experimentos indicam que o método de dois estágios alcança melhor equilíbrio entre robustez e eficiência computacional, reduzindo o custo médio e o tempo de execução em relação à abordagem determinística tradicional.

Decomposição e horizonte rolante em sistemas de produção e serviços

A obra de Pinedo (2016) fornece o embasamento teórico que sustenta a maioria das heurísticas de decomposição discutidas na literatura contemporânea. O autor demonstra que a decomposição é uma forma de hierarquização natural das decisões em problemas de escalonamento: subproblemas locais (máquinas, tarefas ou intervalos) são resolvidos sequencialmente, enquanto um mecanismo de coordenação global garante coerência entre as soluções parciais. Em particular, o conceito de horizonte rolante (*rolling horizon*) é enfatizado como técnica que permite combinar otimização e replanejamento contínuo, aplicável tanto a sistemas produtivos quanto a serviços intensivos em recursos humanos.

O autor ainda apresenta formulações canônicas de PLIM para problemas de máquina única e de múltiplas máquinas, destacando a utilidade das relaxações lineares como instrumentos heurísticos. Essas relaxações podem gerar políticas de priorização e algoritmos de inserção iterativa, constituindo a base teórica para heurísticas de duas fases semelhantes à proposta de Elahipanah, Desaulniers e Lacasse-Guay (2013).

Síntese e relação com o problema de higienização hospitalar

A revisão dos estudos evidencia que as heurísticas de decomposição são particularmente eficazes em problemas com múltiplos níveis de decisão, restrições heterogêneas e alto volume de variáveis binárias – características inerentes ao problema de programação das atividades de higienização hospitalar.

A estratégia de Elahipanah, Desaulniers e Lacasse-Guay (2013) inspira diretamente o presente trabalho, ao adotar uma decomposição em duas fases que separa decisões estruturais (alocação de tarefas longas e críticas) de decisões táticas (preenchimento de janelas com tarefas curtas). De forma análoga, a abordagem de Wu *et al.* (2023) reforça o potencial de integração de cenários estocásticos para aprimorar a robustez do planejamento.

Por fim, os princípios apresentados por Pinedo (2016) oferecem a base teórica para a adoção de uma heurística de decomposição no contexto hospitalar, assegurando coerência com a literatura clássica de *scheduling* e com os métodos de otimização hierárquica aplicados a sistemas de grande escala.

3 METODOLOGIA

No capítulo de Metodologia, apresenta-se o processo de coleta, organização e tratamento dos dados utilizados para parametrizar o modelo. São descritas as fontes de informação, as ferramentas computacionais adotadas, as etapas de estruturação das planilhas e os critérios utilizados para estimar tempos, esforços e periodicidades das tarefas. Além disso, o capítulo detalha o ambiente técnico de desenvolvimento, justificando a escolha do Python, do Gurobi e dos demais recursos auxiliares. O objetivo é oferecer transparência metodológica e assegurar que as bases utilizadas no modelo reflitam fielmente as condições reais da operação.

3.1 Coleta de dados

A etapa de coleta de dados teve como objetivo obter informações quantitativas e qualitativas necessárias para a parametrização do modelo de otimização de higienização hospitalar proposto. Para tanto, foram conduzidas entrevistas e observações diretas junto a diferentes perfis de colaboradores do setor de limpeza do Instituto CEMA, abrangendo tanto níveis operacionais quanto administrativos.

Foram realizadas conversas estruturadas com os supervisores de limpeza e com o responsável pela hotelaria, a fim de compreender as políticas internas de higienização, as normas institucionais de periodicidade das tarefas e os critérios de criticidade atribuídos a cada tipo de ambiente. Complementarmente, quatro profissionais de execução foram entrevistados individualmente, com o intuito de capturar percepções práticas sobre o esforço físico envolvido nas atividades, tempos médios de execução e eventuais fatores contextuais que impactam o desempenho.

A coleta de dados subjetivos – como percepção de esforço e complexidade da tarefa – apresentou desafios consideráveis, uma vez que tais dimensões variam entre indivíduos e dependem de fatores como experiência, turno de trabalho e familiaridade com o ambiente. Para mitigar essa variabilidade, foi solicitado que cada profissional atribuísse uma nota de 1 a 5 para o nível de esforço percebido em cada ambiente, onde 1 representa esforço mínimo e 5 esforço máximo. As notas individuais foram posteriormente consolidadas pela média aritmética, resultando em um indicador agregado de esforço por ambiente, conforme mostra a Tabela 1.

Tabela 1 – Notas de esforço por ambiente

Ambiente	**Op1**	**Op2**	**Op3**	**Op4**	**Esforço médio**
CENTRO CIRÚRGICO	5	5	5	4	4,75
PEQ. CIRÚRGIAS-CORREDORES	5	5	5	4	4,75
PEQ. CIRÚRGIAS-SALAS	4,5	5	5	4,5	4,75
CME	5	5	5	4,4	4,85
PRONTO ATENDIMENTO	3,5	3,5	4	4	3,75
CÂMARA FRIA	4	4	2,5	4,5	3,75
EXPURGOS	5	5	5	5	5,00
FARMÁCIA	3	3,5	3	3,5	3,25
MED. TRABALHO	2	1,5	2,5	2	2,00
LEITOS DE ISOLAMENTO	5	4	4,5	4	4,38
APARTAMENTOS / ENFERMARIAS	3,5	3,5	3	4	3,50
BANHEIROS	4	3	3	3,5	3,38
POSTOS DE ENFERMAGEM	3	3,5	3	3,5	3,25
CONSULTÓRIOS	3	2,5	2,5	3	2,75
ROUPARIA - ÁREA LIMPA	3	4	3	3	3,25
ELEVADORES	2	2	3	2	2,25
CORREDORES (CIRCULAÇÃO)	3,5	3,5	4,5	3	3,63
SALAS ADMINISTRATIVAS	3	2	2	2	2,25
GUARITA	2	1,5	1	2,5	1,75
ÁREAS COMUNS	3	2,5	3,5	3,5	3,13
ALMOXARIFADO	2	3	2	2,5	2,38
SALAS DE ESPERA	3,5	4	4	3,5	3,75

Fonte: Elaboração própria com dados fornecidos pelo CEMA.

Além das avaliações subjetivas, foram obtidos dados objetivos de tempo de execução das tarefas, tanto para a limpeza terminal (Tabela 2) quanto para a limpeza concorrente (Tabela 3), a partir de registros de observação direta e cronometração de atividades. Os valores coletados foram transformados em tempos médios de referência, utilizados na modelagem das restrições de capacidade e alocação de profissionais. Também foram registradas as quantidades mínimas de pessoas necessárias para a execução simultânea de cada tarefa, conforme observação de práticas operacionais e confirmação dos supervisores.

Tabela 2 – Tempos médios de limpeza terminal por ambiente (min)

Ambiente	T1	T2	T3	T4	Tempo médio
CENTRO CIRÚRGICO	40	45	45	45	43,75
PEQ. CIRÚRGIAS – CORREDORES	60	60	60	60	60,00
PEQ. CIRÚRGIAS – SALAS	45	40	40	45	42,50
CME	150	160	150	150	152,50
PRONTO ATENDIMENTO	120	135	120	120	123,75
CÂMARA FRIA	50	45	45	45	46,25
EXPURGOS	40	40	40	40	40,00
FARMÁCIA	120	120	120	120	120,00
MEDICINA DO TRABALHO	60	60	75	60	63,75
LEITOS DE ISOLAMENTO	60	60	60	60	60,00
APARTAMENTOS / ENFERMARIAS	75	75	75	80	76,25
BANHEIROS	60	60	55	60	58,75
POSTOS DE ENFERMAGEM	60	60	60	60	60,00
CONSULTÓRIOS	45	45	40	40	42,50
ROUPARIA – ÁREA LIMPA	60	60	60	60	60,00
ELEVADORES	40	40	40	40	40,00
CORREDORES (CIRCULAÇÃO)	120	135	120	125	125,00
SALAS ADMINISTRATIVAS	60	65	60	60	61,25
GUARITA	55	55	55	60	56,25
ÁREAS COMUNS	90	90	95	90	91,25
ALMOXARIFADO	90	90	90	90	90,00
SALAS DE ESPERA	90	100	90	90	92,50

Fonte: Elaboração própria com dados fornecidos pelo CEMA.

Tabela 3 – Tempos médios de limpeza concorrente por ambiente (min)

Ambiente	T1	T2	T3	T4	Tempo médio
CENTRO CIRÚRGICO	20	20	15	20	18,75
PEQ. CIRÚRGIAS – CORREDORES	20	15	15	20	17,50
PEQ. CIRÚRGIAS – SALAS	20	15	20	15	17,50
CME	30	30	35	30	31,25
PRONTO ATENDIMENTO	30	30	30	30	30,00
CÂMARA FRIA	15	15	10	15	13,75
EXPURGOS	25	30	25	25	26,25
FARMÁCIA	30	35	30	30	31,25
MED. TRABALHO	25	20	20	20	21,25
LEITOS DE ISOLAMENTO	30	30	30	30	30,00
APARTAMENTOS / ENFERMARIAS	25	25	25	20	23,75
BANHEIROS	30	25	30	30	28,75
POSTOS DE ENFERMAGEM	30	30	30	30	30,00
CONSULTÓRIOS	15	20	20	20	18,75
ROUPARIA - ÁREA LIMPA	25	30	25	25	26,25
ELEVADORES	15	10	15	15	13,75
CORREDORES (CIRCULAÇÃO)	25	30	30	30	28,75
SALAS ADMINISTRATIVAS	20	25	20	20	21,25
GUARITA	25	25	25	30	26,25
ÁREAS COMUNS	30	30	25	30	28,75
ALMOXARIFADO	45	40	40	40	41,25
SALAS DE ESPERA	25	30	30	30	28,75

Fonte: Elaboração própria com dados fornecidos pelo CEMA.

Adicionalmente, foram levantadas as normas internas de periodicidade das atividades de limpeza, associadas ao tipo e à criticidade do ambiente. Essa informação foi fundamental para

compor o conjunto de janelas temporais de execução dentro do modelo matemático, distinguindo, por exemplo, áreas críticas (com necessidade de higienização diária) de áreas de apoio (com periodicidade semanal), como mostrado na Figura 2.

A Tabela 4 apresenta a distribuição do número de ambientes por categoria funcional, enquanto a Tabela 5 indica o número médio de profissionais necessários para a execução adequada da limpeza em cada tipo de área, conforme levantamento empírico realizado junto à equipe de supervisão do Instituto CEMA.

Tabela 4 – Quantidade de ambientes por categoria

Ambiente	**Quantidade (unidades)**
CENTRO CIRÚRGICO	15
PEQ. CIRÚRGIAS – CORREDORES	1
PEQ. CIRÚRGIAS – SALAS	5
CME	3
PRONTO ATENDIMENTO	1
CÂMARA FRIA	1
EXPURGOS	6
FARMÁCIA	1
MEDICINA DO TRABALHO	1
LEITOS DE ISOLAMENTO	15
APARTAMENTOS / ENFERMARIAS	125
BANHEIROS	36
POSTOS DE ENFERMAGEM	2
CONSULTÓRIOS	65
ROUPARIA – ÁREA LIMPA	1
ELEVADORES	3
CORREDORES (CIRCULAÇÃO)	12
SALAS ADMINISTRATIVAS	6
GUARITA	1
ÁREAS COMUNS	1
ALMOXARIFADO	1
SALAS DE ESPERA	1

Fonte: Elaboração própria com dados fornecidos pelo CEMA.

Em síntese, a coleta de dados combinou elementos quantitativos (tempos, quantidades, periodicidade) e qualitativos (esforço, criticidade, relevância), permitindo a construção de um banco de dados coerente com a realidade operacional do setor e adequado para a parametrização do modelo de otimização.

Tabela 5 – Número de pessoas necessárias para a limpeza por ambiente

Ambiente	**Nº de pessoas**
CENTRO CIRÚRGICO	3
PEQ. CIRÚRGIAS – CORREDORES	3
PEQ. CIRÚRGIAS – SALAS	3
CME	3
PRONTO ATENDIMENTO	2
CÂMARA FRIA	1
EXPURGOS	1
FARMÁCIA	2
MEDICINA DO TRABALHO	1
LEITOS DE ISOLAMENTO	1
APARTAMENTOS / ENFERMARIAS	1
BANHEIROS	2
POSTOS DE ENFERMAGEM	1
CONSULTÓRIOS	3
ROUPARIA – ÁREA LIMPA	1
ELEVADORES	1
CORREDORES (CIRCULAÇÃO)	2
SALAS ADMINISTRATIVAS	3
GUARITA	1
ÁREAS COMUNS	1
ALMOXARIFADO	2
SALAS DE ESPERA	2

Fonte: Elaboração própria com dados fornecidos pelo CEMA.

3.2 Ferramentas utilizadas

O desenvolvimento deste trabalho envolveu a utilização de um conjunto integrado de ferramentas computacionais e de apoio à modelagem matemática, selecionadas com base em critérios de desempenho, acessibilidade e adequação ao tipo de problema abordado. As principais ferramentas empregadas foram o Python, o solucionador (*solver*) Gurobi Optimizer e o Microsoft Excel, além de bibliotecas auxiliares para manipulação de dados e visualização de resultados.

Linguagem Python

A linguagem Python foi escolhida como base para a implementação do modelo matemático e dos algoritmos de otimização devido à sua ampla aplicação em pesquisa operacional e ciência de dados. Trata-se de uma linguagem de alto nível, de código aberto, que oferece sintaxe simples e uma grande variedade de bibliotecas voltadas à análise numérica, como NumPy, Pandas e Matplotlib. O uso do Python proporcionou flexibilidade na estruturação do código, permitindo a integração direta com o *solver* de otimização e facilitando a leitura e escrita de dados. Além disso, sua popularidade na comunidade científica garante suporte e documentação extensos, o que contribuiu para maior eficiência no desenvolvimento.

Solver Gurobi Optimizer

O Gurobi Optimizer foi o *solver* utilizado para resolver o modelo de Programação Inteira Mista (PIM) proposto. Sua escolha se justifica pela elevada eficiência na resolução de problemas de otimização combinatória de grande porte, oferecendo algoritmos de última geração e técnicas

avançadas de *branch-and-bound*, *cutting planes* e paralelização automática. O Gurobi também possui integração nativa com a linguagem Python, por meio da biblioteca gurobipy, o que permitiu a formulação direta das variáveis, restrições e funções objetivo de forma programática. Essa característica viabilizou a criação de um ambiente experimental dinâmico, no qual diferentes cenários puderam ser testados com rapidez e controle total sobre os parâmetros de execução.

Microsoft Excel

O Microsoft Excel foi utilizado como ferramenta de apoio à coleta e organização dos dados necessários à parametrização do modelo, incluindo informações sobre ambientes, profissionais e tempos de execução. A escolha do Excel se deve à sua ampla disponibilidade e facilidade de uso, especialmente em contextos empresariais e institucionais. Todas as planilhas de entrada e saída do modelo – denominadas "máscaras" – foram desenvolvidas nesse ambiente, de modo a garantir compatibilidade com os processos já estabelecidos na instituição. Essa decisão visou não apenas facilitar a manipulação dos dados, mas também assegurar a futura adoção da ferramenta pelo time responsável, evitando a introdução de novas complexidades técnicas. O uso do Excel reflete o formato habitual de trabalho utilizado para elaboração de escalas e planos operacionais, permitindo que a solução desenvolvida se integre naturalmente à rotina existente, sem necessidade de treinamento especializado ou adoção de softwares adicionais.

Bibliotecas e Ferramentas Complementares

Além das ferramentas principais, foram empregadas bibliotecas auxiliares no ambiente Python, como OpenPyXL para leitura e escrita de planilhas Excel, Seaborn para a visualização dos resultados obtidos, e JSON para o armazenamento de checkpoints e parâmetros de execução. Essas ferramentas complementares possibilitaram maior automação no fluxo de trabalho, desde o carregamento dos dados até a análise dos resultados, garantindo reprodutibilidade e organização durante as diferentes etapas do projeto.

Justificativa da Integração das Ferramentas

A integração entre Python, Gurobi e Excel permitiu unir a robustez computacional da otimização matemática com a praticidade de manipulação e análise de dados em planilhas. Essa combinação tornou possível desenvolver uma solução completa, que abrange desde o tratamento dos dados de entrada até a geração automática dos resultados otimizados e relatórios, assegurando eficiência, transparência e facilidade de manutenção do código. Além disso, a adoção do Excel como interface principal de entrada e saída viabiliza a aplicação direta da ferramenta no contexto institucional, promovendo usabilidade, aderência às rotinas já praticadas e sustentabilidade na continuidade do uso após o término do projeto.

Hardware

Os experimentos foram executados localmente em um computador equipado com Windows 11 e processador Intel(R) Core(TM) Ultra 7 165U, com 12 núcleos físicos, 14 *threads* lógicas e frequência base de 1,7 GHz, além de 32 GB de memória RAM.

4 MODELO

Este capítulo descreve em profundidade o modelo matemático desenvolvido para representar o problema de programação das atividades de higienização. São apresentados seus conjuntos, parâmetros, variáveis, função objetivo e restrições, acompanhados de interpretações operacionais e justificativas conceituais. O capítulo também discute elementos estruturais do modelo, como janelas temporais, equipes simultâneas e balanceamento de esforço. Por fim, são apresentados experimentos de configuração, métricas de qualidade, limitações computacionais e análises comparativas entre diferentes formulações. O objetivo é oferecer ao leitor uma compreensão completa da lógica de otimização utilizada.

4.1 Descrição geral do modelo

O modelo desenvolvido neste trabalho tem como objetivo estruturar e otimizar o processo de planejamento e alocação das tarefas de higienização hospitalar no Instituto CEMA. Trata-se de uma atividade complexa, que envolve uma grande quantidade de ambientes com diferentes níveis de criticidade, periodicidades de limpeza e tempos de execução distintos. A distribuição manual dessas tarefas, feita com base apenas na experiência dos supervisores, frequentemente resulta em sobrecarga de alguns profissionais, ociosidade de outros e, em certos casos, atraso na higienização de áreas críticas. Nesse contexto, a formulação proposta busca transformar esse processo empírico em uma decisão racional e sistemática, permitindo planejar toda a rotina de limpeza de maneira justa, eficiente e aderente às normas da instituição.

Para representar a realidade do setor de limpeza, o modelo precisa descrever matematicamente cada elemento envolvido – profissionais, dias de trabalho, ambientes, tipos de tarefa e janelas de periodicidade. Cada um desses componentes corresponde a um conjunto ou índice dentro do modelo, e o cruzamento entre eles dá origem às decisões de alocação. Assim, o modelo conhece quem pode trabalhar, quando, onde e em qual tipo de atividade.

Uma das estruturas mais importantes é o conjunto de ambientes. Um mesmo tipo de sala – por exemplo, uma sala de exames – pode existir em várias unidades físicas do hospital, espalhadas por diferentes andares ou setores. Para representar isso adequadamente, o modelo utiliza o conceito de ambientes expandidos, denotado por $\tilde{A}$. Esse conjunto não contém apenas as categorias genéricas de ambientes (como “consultório”, “enfermaria” ou “centro cirúrgico”), mas cada ocorrência individual, identificada de forma única. Assim, o ambiente “consultório 1” e o “consultório 2” aparecem como elementos distintos dentro de $\tilde{A}$, ainda que ambos pertençam ao mesmo tipo base A. Essa expansão é fundamental para permitir que o modelo planeje a limpeza de cada espaço físico separadamente, respeitando suas janelas de tempo e frequências específicas. Em outras palavras, A representa o tipo de área, enquanto $\tilde{A}$ representa as instâncias concretas dessas áreas no prédio – e é nesse nível de detalhe que o modelo toma suas decisões.

Além da estrutura dos ambientes, o modelo precisa representar os profissionais de limpeza e sua disponibilidade. Cada trabalhador possui uma carga horária diária definida, associada ao tipo de turno em que atua (manhã, tarde ou noite), e o modelo deve respeitar essas limitações rigorosamente. Por isso, há parâmetros que indicam a jornada máxima permitida por dia e outros que marcam em quais dias cada profissional está disponível. Esses dados garantem que o planejamento não atribua tarefas a quem não esteja de plantão ou extrapole o limite de horas.

Os tipos de tarefa de higienização são outro componente central. Cada tarefa – como limpeza concorrente, terminal, diária ou semanal – possui uma duração média e pode exigir diferentes quantidades de profissionais para ser executada. Além disso, tarefas associadas a ambientes críticos, como UTIs ou centros cirúrgicos, recebem pesos maiores de prioridade e esforço, representando tanto a sua importância operacional quanto a intensidade física envolvida (calculada pelo fator de esforço). Esses parâmetros permitem que o modelo diferencie atividades mais simples e rotineiras de outras que são mais exigentes e sensíveis.

Para que as tarefas sejam alocadas de forma realista, o modelo considera ainda o conceito de janelas de tempo. Cada ambiente e tipo de tarefa possuem uma periodicidade pré-definida – diária, semanal, quinzenal etc. – e o modelo traduz essas regras em intervalos de dias em que a limpeza pode ou deve ocorrer. Dentro de cada janela, há um número mínimo de vezes que a tarefa deve ser realizada, assegurando que as normas institucionais sejam cumpridas. Essa estrutura é o que garante que uma sala que exige limpeza a cada três dias, por exemplo, não seja negligenciada nem excessivamente agendada.

Por fim, as variáveis de decisão representam as escolhas que o modelo precisa fazer: se uma tarefa será executada em determinado dia e, se for o caso, quais profissionais serão responsáveis por ela. Com base nas combinações possíveis de ambientes, tarefas, dias e pessoas, o modelo gera um plano completo de trabalho, respeitando todas as restrições operacionais e buscando o melhor equilíbrio possível entre cobertura e equidade.

A função principal do modelo é encontrar, entre todas as combinações possíveis, aquela que maximiza a realização das tarefas, impedindo que tarefas críticas não sejam realizadas e, ao mesmo tempo, distribui de maneira equilibrada o esforço e o tempo de trabalho entre os profissionais. Assim, ele garante que as áreas críticas nunca fiquem sem higienização dentro do prazo, que a equipe seja utilizada de forma eficiente e que nenhum colaborador fique sobrecarregado em relação aos demais. O resultado é um plano operacional detalhado, que indica exatamente quais ambientes devem ser limpos, em que dias, por quem e por quanto tempo.

Em resumo, o modelo proposto é uma ferramenta de apoio à decisão que traduz a rotina de higienização hospitalar em uma estrutura lógica e mensurável. A introdução do conjunto de ambientes expandidos $\tilde{A}$ e dos demais parâmetros e índices é o que permite representar fielmente a complexidade do cenário real do Instituto CEMA, tornando possível automatizar o planejamento, padronizar as decisões e alcançar um nível de eficiência e confiabilidade que dificilmente seria obtido por meios manuais.

4.2 Formulação do problema

Conjuntos e índices:

- I: profissionais de higienização (índice i).
- J: dias no horizonte de planejamento (índice j).
- $\tilde{A}$: ambientes expandidos (índice a).
- K: tipos de tarefas (índice k).
- $\mathcal{W}_{ak}$: conjunto de janelas τ de (a,k); cada τ define um intervalo $[s_{ak\tau}, e_{ak\tau}] \subseteq J$ e um requisito $r_{ak\tau} \in \mathbb{Z}_{\geq 0}$.
- $\mathcal{S} \subseteq \tilde{A} \times J \times K$: triplas (a,j,k) habilitadas por alguma janela de (a,k) (isto é, com $j \in [s_{ak\tau},e_{ak\tau}]$ para alguma $\tau \in \mathcal{W}_{ak}$).

Parâmetros:

- $W_{ij} \in \{0,1\}$: disponibilidade do profissional i no dia j.
- H_i: jornada (horas/dia) do profissional i.
- D_{ak}: duração (horas) da tarefa k no ambiente expandido a (herdada da subárea base ou classe).
- E_{ak}: esforço (peso) da tarefa k no ambiente a.
- P_{ak}: prioridade (peso) da tarefa k no ambiente a.
- $C_{ajk} \in \mathbb{Z}_{\geq 1}$: tamanho de equipe requerido para (a,j,k).
- $M_{ijak} \in \{0,1\}$: máscara de elegibilidade (1 se é permitido alocar i em (a,j,k), considerando turno, proibições de janela/tarefa e disponibilidade; 0 caso contrário). No código, M_{ijak} é implementado removendo variáveis proibidas do domínio.
- $A_{ajk} \in \{0,1\}$: ativação por janela (1 se $(a,j,k) \in \mathcal{S}$; 0 caso contrário). No código, A_{ajk} é implícito via definição de $\mathcal{S}$.
- NCC: Número de Critérios Complementares; no caso, 2 (esforço e duração).

Variáveis de decisão:

- $z_{ijak} \in \{0,1\}$: 1 se o profissional i executa a tarefa k no ambiente a no dia j; 0 caso contrário.

- $y_{ajk} \in \{0,1\}$: 1 se a tarefa k no ambiente a é realizada no dia j; 0 caso contrário. (Só definida para $(a,j,k) \in \mathcal{S}$.)
- δ_i: desvio absoluto (horas) da carga de trabalho de i em relação à média.
- ϵ_i: desvio absoluto (adimensional) do esforço de i em relação à média.
- $\bar{H}$, $\bar{E}$: médias de horas e de esforço por profissional.

Definições auxiliares:

$$H_{\text{total}} = \sum_{i \in I} \sum_{j \in J} H_i \, W_{ij}, \quad W_{ij}, r_{ak\tau} \in \{0,1\} \tag{4.1}$$

$$E_{\text{total}} = \sum_{a \in \tilde{A}} \sum_{k \in K} \sum_{\tau \in \mathcal{W}_{ak}} E_{ak} \, r_{ak\tau} \tag{4.2}$$

4.3 O modelo

Função Objetivo:

$$\max \; Z \; = \; \sum_{(a,j,k) \in \mathcal{S}} P_{ak} \cdot y_{ajk} \; - \; \frac{1}{NCC} \left(\frac{\sum_{i \in I} \delta_i}{H_{\text{total}}} + \frac{\sum_{i \in I} \epsilon_i}{E_{\text{total}}} \right) \tag{4.3}$$

Sujeita às restrições:

Ativação por janela:

$$y_{ajk} \; \leq \; A_{ajk}, \qquad \forall \, (a,j,k) \in \tilde{A} \times J \times K. \tag{4.4}$$

Cobertura com equipe simultânea:

$$\sum_{i \in I} z_{ijak} \; = \; C_{ajk} \; \cdot y_{ajk}, \qquad \forall \, (a,j,k) \in \mathcal{S}. \tag{4.5}$$

Capacidade diária por profissional:

$$\sum_{\substack{(a,k): \\ (a,j,k) \in \mathcal{S}}} z_{ijak} \, D_{ak} \; \leq \; H_i \cdot W_{ij}, \qquad \forall \, i \in I, \forall \, j \in J \, . \tag{4.6}$$

Mínimos por janela (requisitos):

$$\sum_{j=s_{ak\tau}}^{e_{ak\tau}} y_{ajk} \; \geq \; r_{ak\tau}, \qquad \forall \, a \in \tilde{A}, \forall \, k \in K, \forall \, \tau \in \mathcal{W}_{ak}. \tag{4.7}$$

Máscara de elegibilidade (turno/proibição/disp.):

$$z_{ijak} \;\leq\; M_{ijak}, \qquad \forall\, i \in I, \forall\, (a,j,k) \in \mathcal{S}. \tag{4.8}$$

Ligações das médias e desvios absolutos (balanceamento):

$$\sum_{(a,j,k)\in\mathcal{S}} z_{ijak} \cdot\, D_{ak} - \bar{H} \;\leq\; \delta_i, \qquad \forall\, i \in I, \tag{4.9}$$

$$-\Big(\sum_{(a,j,k)\in\mathcal{S}} z_{ijak} \cdot\, D_{ak} - \bar{H}\Big) \;\leq\; \delta_i, \qquad \forall\, i \in I, \tag{4.10}$$

$$\sum_{(a,j,k)\in\mathcal{S}} z_{ijak} \;\cdot\, E_{ak} - \bar{E} \;\leq\; \epsilon_i, \qquad \forall\, i \in I, \tag{4.11}$$

$$-\Big(\sum_{(a,j,k)\in\mathcal{S}} z_{ijak} \;\cdot\, E_{ak} - \bar{E}\Big) \;\leq\; \epsilon_i, \qquad \forall\, i \in I. \tag{4.12}$$

Domínio das variáveis:

$$z_{ijak} \cdot\, y_{ajk} \in \{0,1\}, \qquad \delta_i \geq 0,\, \epsilon_i \geq 0. \tag{4.13}$$

4.4 Análise da função objetivo

A função objetivo (FO) proposta é de natureza multicritério, consolidando os objetivos estratégicos do hospital em uma única expressão a ser maximizada.

$$\max\; Z \;=\; \sum_{(a,j,k)\in\mathcal{S}} P_{ak} \cdot y_{ajk} \;-\; \frac{1}{NCC}\left(\frac{\sum_{i\in I}\delta_i}{H_{\text{total}}} + \frac{\sum_{i\in I}\epsilon_i}{E_{\text{total}}}\right) \tag{4.3}$$

Esta função pode ser decomposta em dois componentes principais, que representam um problema em dois níveis hierárquicos de decisão:

1. **Nível superior (maximização da cobertura ponderada):** O primeiro termo é o objetivo primário. Ao maximizá-lo, o modelo busca atender ao maior número possível de tarefas. A ponderação pelo parâmetro P_{ak}, que assume valores muito maiores para áreas semicríticas (e.g., $P_{ak} = 3$) do que para áreas não críticas ($P_{ak} = 1$), garante que o modelo priorize intrinsecamente a alocação de recursos para as tarefas de maior impacto na segurança do paciente. Esta abordagem, como discutido na revisão de literatura, reflete as práticas de modelagem que diferenciam a importância de diferentes metas. Isso se aplica somente para tarefas semicriticas e não críticas, mais adiante será posta uma restrição onde as tarefas críticas são obrigatórias, e não tem a opção de não serem realizadas. A decisão foi modelar a obrigatoriedade das tarefas críticas como uma restrição rígida (*hard constraint*), pois reflete com precisão a realidade operacional de um hospital, onde a higienização de áreas como UTIs e Centros Cirúrgicos é inegociável.

2. **Nível intermediário (minimização das penalidades por desbalanceamento):** De acordo com a estrutura de otimização multiobjetivo, este nível é tratado como um objetivo secundário dentro de uma abordagem lexicográfica (ISERMANN, 1982). Após o modelo alcançar um elevado nível de cobertura das necessidades operacionais – objetivo primário –, a minimização das penalidades associadas ao desbalanceamento entre profissionais atua como critério de refinamento, responsável por distinguir entre soluções de mesma qualidade em relação ao primeiro objetivo (COELLO, 1999; PINEDO, 2016). As variáveis δ_i e ϵ_i medem o desvio da carga de trabalho do profissional i em relação à média, considerando tanto a dimensão de duração (horas) quanto de dificuldade (esforço). Ao minimizar a soma desses desvios, o modelo busca uma distribuição mais equitativa das atividades, promovendo a justiça interna, prevenindo sobrecargas (*burnout*) e contribuindo para o bem-estar coletivo da equipe. Assim, o segundo nível de otimização complementa o primeiro, assegurando que a solução final não apenas seja eficiente do ponto de vista operacional, mas também equilibrada e sustentável do ponto de vista humano.

Um dos principais desafios em funções objetivo multicritério é a dominância de escala, situação em que um termo com valores numericamente muito maiores – como a pontuação de cobertura – pode ofuscar completamente os demais termos, como as penalidades por desbalanceamento, tornando-os praticamente irrelevantes no processo de otimização. Para evitar esse problema e garantir que todos os objetivos sejam considerados de forma proporcional, é fundamental aplicar uma técnica de normalização dos componentes da função objetivo.

A estratégia adotada consiste em escalar os termos de penalidade de modo que seus valores fiquem contidos em um intervalo previsível, tipicamente entre 0 e 1. Essa restrição garante que os objetivos secundários – responsáveis pelo balanceamento das cargas de trabalho – mantenham influência controlada sobre o modelo, sem competir numericamente com o objetivo principal, que é priorizado de acordo com a abordagem lexicográfica (ISERMANN, 1982; COELLO, 1999; PINEDO, 2016). Assim, evita-se que pequenas variações no termo dominante (cobertura) sejam mascaradas por diferenças de magnitude nas penalidades, preservando a hierarquia entre os critérios de otimização.

Operacionalmente, a normalização é obtida dividindo-se a soma dos desvios observados pelo respectivo valor máximo teórico possível. Sejam H_{total} o total de horas de trabalho disponíveis no horizonte de planejamento e E_{total} o esforço total máximo teórico, as penalidades associadas ao desbalanceamento temporal e de esforço são expressas como proporções relativas a esses totais. Posteriormente, aplica-se um fator de NCC de modo que a soma normalizada permaneça no intervalo $[0,1]$. Dessa forma, o modelo mantém a coerência dimensional e garante que todas as parcelas da função objetivo operem em uma mesma escala, compatível com o princípio de dominância hierárquica da otimização multiobjetivo.

4.5 Análise das restrições

As restrições do modelo garantem que a solução gerada seja operacionalmente viável, legalmente compatível e logicamente consistente.

Ativação por janela

$$y_{ajk} \;\leq\; A_{ajk} \qquad \forall\,(a,j,k) \in \tilde{A} \times J \times K. \tag{4.4}$$

A restrição assegura que a tarefa k no ambiente a só pode ser marcada como realizada ($y_{ajk} = 1$) no dia j se esse dia estiver habilitado por alguma janela válida ($A_{ajk} = 1$). Em outras palavras, A_{ajk} atua como "máscara temporal", desativando variáveis quando fora da janela permitida.

Cobertura com equipe simultânea

$$\sum_{i \in I} z_{ijak} \;=\; C_{ajk} \cdot y_{ajk} \qquad \forall\,(a,j,k) \in \mathcal{S}. \tag{4.5}$$

Garante que, quando a tarefa (a,j,k) é executada (y_{ajk}=1), exatamente C_{ajk} profissionais são alocados simultaneamente a ela naquele dia. Se a tarefa não é realizada ($y_{ajk} = 0$), nenhum profissional pode ser alocado. Formulações análogas para *job teaming/synchronization* estão presentes em modelos da literatura; por exemplo, em uma formulação mestre com rotas, a exigência de número de equipes quando uma tarefa é iniciada no tempo t é imposta por restrições lineares equivalentes (DOHN; KOLIND; CLAUSEN, 2009).

Capacidade diária por profissional

$$\sum_{\substack{(a,k):\\(a,j,k)\in\mathcal{S}}} z_{ijak} \cdot D_{ak} \;\leq\; H_i \cdot W_{ij} \qquad \forall\, i \in I, \forall\, j \in J. \tag{4.6}$$

Esta é uma restrição fundamental de capacidade de recursos, garantindo que a soma das durações das tarefas atribuídas a um profissional em um dia não exceda sua jornada de trabalho. O parâmetro W_{ij} anula a restrição em dias de folga. Restrições de tempo total de trabalho por recurso/servidor com janelas e deslocamentos são padrão em problemas de alocação de pessoal com janelas de tempo (LIM; RODRIGUES; SONG, 2004).

Mínimos por janela (requisitos)

$$\sum_{j=s_{ak\tau}}^{e_{ak\tau}} y_{ajk} \;\geq\; r_{ak\tau}, \qquad \forall\, a \in \tilde{A}, \forall\, k \in K, \forall\, \tau \in \mathcal{W}_{ak}. \tag{4.7}$$

Impõe que, dentro de cada intervalo de janela τ, a tarefa seja realizada ao menos $r_{ak\tau}$ vezes. Assim, frequências mínimas (diárias, semanais etc.) são respeitadas conforme a política de higienização (GOMES; RAMOS, 2019).

Máscara de elegibilidade (turno/proibição/disponibilidade)

$$z_{ijak} \;\leq\; M_{ijak}, \qquad \forall\, i \in I, \forall\, (a,j,k) \in \mathcal{S}. \tag{4.8}$$

Impossibilita alocações proibidas (por turno, regra de tarefa ou indisponibilidade). Nos casos em que $M_{ijak} = 0$, é vetada a alocação de i em (a,j,k) naquele dia.

Vínculos de média e desvios absolutos

A equidade na distribuição do trabalho é um conceito multifacetado. Um profissional pode cumprir uma carga horária similar à de um colega, mas ser consistentemente alocado a tarefas de maior complexidade e estresse físico/mental. Para abordar essa questão, o modelo incorpora duas dimensões de balanceamento:

a) Por Duração (Horas): Calcula o volume total de trabalho, medido em tempo, seja distribuído de forma justa.

$$\sum_{j,a,k} z_{ijak} \cdot D_{ak} - \bar{H} \leq \delta_i, \qquad \forall i \tag{4.9}$$

$$-\left(\sum_{j,a,k} z_{ijak} \cdot D_{ak} - \bar{H}\right) \leq \delta_i, \qquad \forall i \tag{4.10}$$

Estas duas restrições linearizam o cálculo do desvio absoluto da carga de trabalho em horas (δ_i) do profissional i em relação à média de horas por profissional $(\bar{H})$.

b) Por Esforço (Dificuldade): Reconhece que tarefas em áreas críticas são mais exigentes. Um parâmetro de esforço, E_{ak}, é definido com base na criticidade (e.g., $E_{ak} = 10$ para áreas críticas, 5 para semicríticas, 1 para não críticas).

$$\sum_{j,a,k} z_{ijak} \cdot E_{ak} - \bar{E} \leq \epsilon_i, \qquad \forall i \tag{4.11}$$

$$-\left(\sum_{j,a,k} z_{ijak} \cdot E_{ak} - \bar{E}\right) \leq \epsilon_i, \qquad \forall i \tag{4.12}$$

De forma análoga, essas restrições calculam o desvio absoluto do esforço total (ϵ_i) do profissional i em relação ao esforço médio por profissional $(\bar{E})$.

A inclusão de ambas as métricas na função objetivo, conforme a abordagem de restrição flexível (BILGIN *et al.*, 2012), permite ao modelo encontrar uma solução que não é apenas equilibrada em volume, mas também em esforço, promovendo uma distribuição de trabalho mais equitativa.

As restrições $\sum z \cdot D - \bar{H} \leq \delta_i$ e $-(\sum z \cdot D - \bar{H}) \leq \delta_i$ (e as análogas para esforço com ϵ_i) representam a linearização padrão da função valor absoluto. Elas são usadas para garantir que

δ_i (ou ϵ_i) seja maior ou igual ao desvio absoluto da carga de trabalho (em horas ou esforço) do profissional i em relação à média ($\bar{H}$ ou $\bar{E}$). Minimizar a soma (ou o máximo) dessas variáveis de desvio na função objetivo promove o balanceamento da carga de trabalho entre os profissionais. Esta técnica é fundamental em Programação por Metas (*Goal Programming*), conforme discutido em Azaiez e Sharif (2005).

Domínio das variáveis

$$z_{ijak}, y_{ajk} \in \{0,1\}, \qquad \delta_i \geq 0, \epsilon_i \geq 0. \tag{4.13}$$

A declaração define os tipos das variáveis utilizadas no modelo: z_{ijak} (alocação profissional-tarefa) e y_{ajk} (conclusão da tarefa) são binárias, representando decisões discretas; enquanto δ_i e ϵ_i são contínuas e não-negativas, representando as magnitudes dos desvios de carga horária e esforço. Esta é uma definição padrão para modelos de Programação Inteira Mista (PIM), onde coexistem variáveis discretas e contínuas (MOSEK APS, 2025).

4.6 Configuração dos testes

Para inspecionar o comportamento da função-objetivo em condições controladas, utilizou-se um *template* sintético com horizonte de 15 dias e 6 profissionais, preservando a lógica de janelas e prioridades do caso real. O conjunto de dados de entrada estão contemplados na Tabela 6.

Tabela 6 – Dados de entrada para o exemplo

Descrição	**Setup**
Profissionais (J)	6, divididos entre turnos (MANHÃ, NOITE E 5X2)
Dias no horizonte (D)	15
Atividades com prioridade (A)	36 (Críticas = 6; Semicríticas = 16; Não Críticas = 8)
Peso de esforço	Cada conjunto de tempo reportado
Janelas Limpeza Terminal	J=9, 1,5-9 dias (por subárea)
Janelas de Limpeza Concorrente	1 ambiente considerado

A Tabela 7 resume os principais quantitativos estruturais que caracterizam a instância real utilizada na execução do modelo. Esses valores evidenciam a magnitude combinatória do problema, tanto em termos de tarefas a serem alocadas quanto de possibilidades de início e combinações avaliadas ao longo do horizonte. A capacidade total disponível, o número elevado de tarefas e, sobretudo, o volume de janelas admissíveis para início das atividades ilustram o crescimento explosivo do espaço de busca, reforçando a natureza combinatorial do problema e sua complexidade intrínseca. Em conjunto, esses indicadores demonstram por que a resolução direta do modelo exato se torna inviável em cenários reais e justificam a adoção de estratégias de decomposição e heurísticas especializadas apresentadas ao longo deste trabalho.

Tabela 7 – Dados consolidados utilizados na execução do modelo

Descrição	**Expressão**	**Valor**
Capacidade total disponível (h)	$H_{total} = \sum_{j \in J} \sum_{d \in D} H_{j,d}$	508
Total de tarefas previstas no horizonte	$\lvert K \rvert = \sum_{k \in K} 1$	57
Total de alocações avaliadas pelo modelo	$\lvert J \rvert \cdot \lvert K \rvert$	2.293

Os termos secundários agregam desvios absolutos por profissional i, a média e o desvio total de horas do profissional i em relação à média de todos os profissionais, e o desvio de esforço em relação à média

4.6.1 Métricas de saída por modelo

Com os mesmos dados de entrada, compararam-se duas formulações, a que considera a penalidade por desbalanceamento de horas trabalhadas e de esforço (Modelo 1), e a que somente busca atender à demanda de tarefas (Modelo 2). Os principais indicadores estão apresentados na Tabela 8.

Tabela 8 – KPIs comparativos por modelo (mesmo input, formulações distintas)

Modelo	**FO 1º Nível**	**FO 2º Nível**	H_{total}	E_{total}	**Desv. E**	**# tarefas**
Modelo 1	655,00	0,32	508	270,75	14,35	57
Modelo 2	655,00	na	508	270,75	14,50	57

4.6.2 Complexidade estrutural

Executando o modelo nas configurações descritas na seção 3.2, a resolução via *branch-and-bound* convergiu em cerca de cinco minutos, com múltiplas soluções viáveis identificadas pelas heurísticas internas e refinadas até a otimalidade. O estágio de *presolve* reduziu aproximadamente 28% das restrições e 22% das variáveis, evidenciando redundâncias controladas e boa estrutura matricial do modelo.

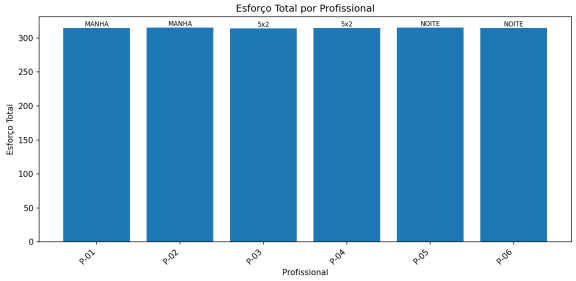
4.6.3 Análise de qualidade

A solução ótima atingiu cobertura integral das ocorrências planejadas, sem violações de capacidade horária. Como não há termos de balanceamento, eventuais assimetrias de carga entre profissionais são esperadas, mas o resultado assegura a máxima utilização das janelas viáveis e prioriza corretamente ambientes críticos, o que era o propósito desta função-objetivo de referência. A análise visual apresentada a seguir complementa essa leitura ao ilustrar como esforço, carga e cronograma se distribuem entre profissionais.

A Figura 4 apresenta a distribuição de esforço e de carga horária por profissional no Modelo 1. A partir dela, observa-se que os valores se mantêm relativamente próximos entre si, indicando um padrão de distribuição mais uniforme. A Figura 4a mostra que a soma ponderada

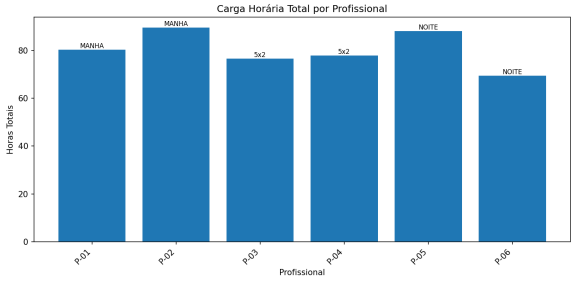
das atividades atribuídas não apresenta grandes discrepâncias; já a Figura 4b evidencia que a quantidade total de horas alocadas por profissional segue comportamento semelhante. Esses resultados sugerem que, mesmo sem um termo explícito de balanceamento, o Modelo 1 produz uma solução estruturalmente homogênea, em que nenhum colaborador se destaca como significativamente sobrecarregado ou subutilizado.

Figura 4 – Distribuição de esforço e carga por profissional – Modelo 1



(a) Esforço por profissional – Modelo 1

Autoria própria.



(b) Carga por profissional – Modelo 1

Autoria própria.

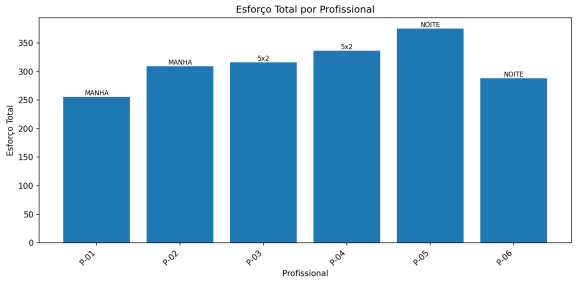
A Figura 5 complementa a análise ao apresentar o cronograma de alocação das tarefas por meio de um gráfico de Gantt. Nota-se que as tarefas foram organizadas de forma coerente com as janelas permitidas, sem sobreposições indevidas ou períodos extensos de ociosidade. O padrão visual revela uma ocupação consistente ao longo do horizonte, refletindo a boa utilização da capacidade horária disponível. Isso indica que o Modelo 1 distribui as tarefas de maneira eficiente no tempo, respeitando restrições e mantendo continuidade operacional.

Figura 5 – Alocação de tarefas com balanceamento de esforço – Modelo 1

Autoria própria.

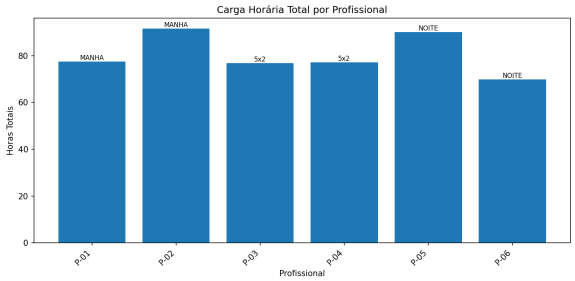
A Figura 6 apresenta a distribuição de esforço e carga no Modelo 2. Diferentemente do observado no Modelo 1, há maior dispersão entre mínimos e máximos, sugerindo desequilíbrios mais acentuados. Alguns profissionais concentram volumes significativamente maiores de esforço e carga, enquanto outros assumem quantidades substancialmente menores. Esse comportamento é coerente com o foco do Modelo 2, cuja função objetivo enfatiza o termo principal em detrimento da equidade entre colaboradores.

Figura 6 – Distribuição de esforço e carga por profissional – Modelo 2



(a) Esforço por profissional — Modelo 2

Autoria própria.



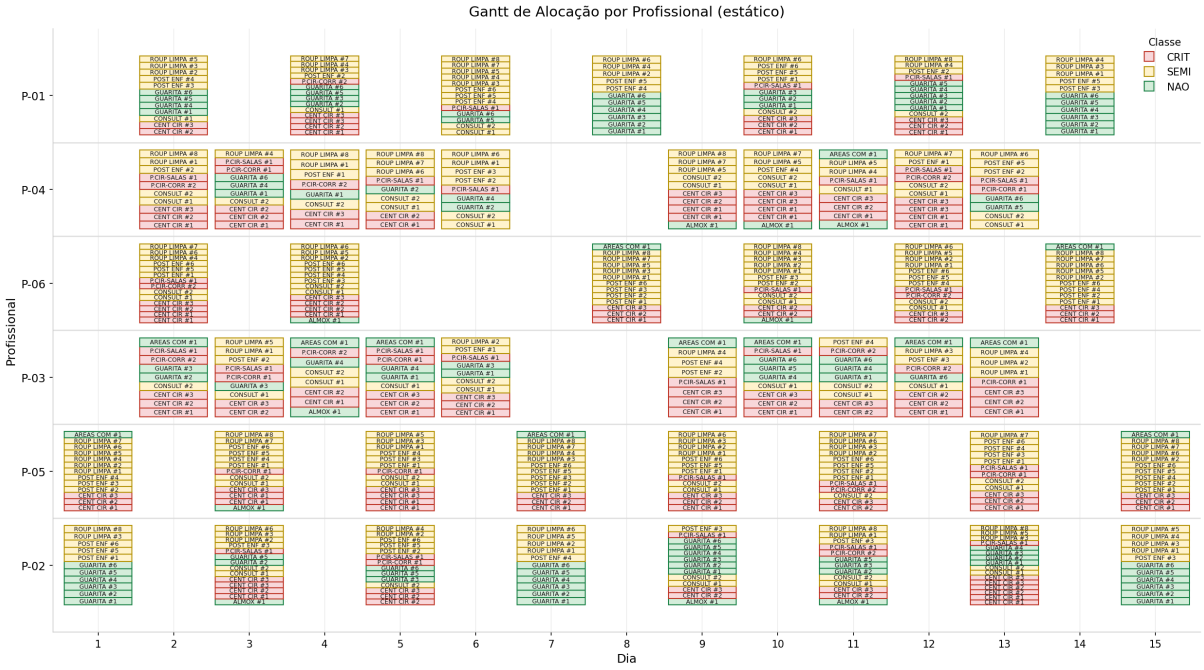
(b) Carga por profissional — Modelo 2

Autoria própria.

A Figura 7 reforça essa interpretação ao exibir o cronograma completo do Modelo 2. Observa-se uma concentração mais evidente de tarefas em poucos profissionais, criando jornadas mais carregadas para alguns e mais leves para outros. Embora a coerência temporal das janelas seja preservada, o padrão resultante apresenta menor uniformidade quando comparado

ao Modelo 1. Esse comportamento, embora garanta 100% de cobertura das tarefas, indica maior assimetria operacional e potencial risco de sobrecarga em aplicações reais.

Figura 7 – Alocação de tarefas sem balanceamento de esforço – Modelo 2



Autoria própria.

4.6.4 Qualidade da solução vs. balanceamento

A função objetivo do Modelo 2 apresenta valor levemente superior (655,00 contra 654,69), porém à custa de um desequilíbrio significativamente maior, ainda que ambos os modelos executem exatamente o mesmo número de tarefas. Esse ganho marginal não decorre de maior produtividade, mas de diferenças na estrutura de pesos e dos termos secundários da função objetivo. No Modelo 2, o somatório de desvios em horas é $\sum \delta = 45{,}00$ (aproximadamente 9% de desequilíbrio frente às horas totais trabalhadas), enquanto o desequilíbrio em esforço é $\sum \epsilon = 177{,}00$ (cerca de 85% frente ao esforço total dos profissionais). Em contraste, no Modelo 1, esses valores são 34,125 (aproximadamente 7% desequilíbrio frente as horas totais trabalhadas) e 1,75 (menos de 1%), respectivamente. Como H_{total}, E_{total} e o número de tarefas são idênticos em ambos os modelos, conclui-se que a diferença de desempenho decorre exclusivamente da estrutura da função objetivo. Portanto, em contextos em que a equidade operacional é fator relevante, o Modelo 1 se mostra mais adequado; por outro lado, se o foco for maximizar apenas o termo principal, o Modelo 2 oferece leve ganho em função objetivo, embora com maior assimetria distributiva.

4.6.5 Escalabilidade

O comportamento do modelo frente ao aumento gradual da quantidade de ambientes evidencia de forma clara o seu limite prático de escalabilidade. Quando o número de ambientes cresce mantendo-se fixo o número de funcionários, o problema se aproxima rapidamente da capacidade máxima de execução do time, reduzindo o espaço de folga e tornando a região factível extremamente densa. Do ponto de vista de otimização inteira, isso significa que o *branch-and-bound* precisa explorar um volume muito maior de nós para distinguir pequenas variações de alocação, pois a maioria das combinações viáveis opera próxima ao limite de horas, janelas e esforço das equipes.

Essa sensibilidade aparece já nos cenários controlados. Com 6 funcionários e 35 ambientes, o modelo obtém solução ótima em poucos minutos, com *gap* nulo ou desprezível. No entanto, um acréscimo moderado para 41 ambientes (mantendo os mesmos 6 funcionários) provoca um salto abrupto no tempo de processamento e na dificuldade de certificar a otimalidade: ambos os modelos (sem e com penalidade de balanceamento) passam a consumir quase todo o limite de tempo imposto, encerrando com *gap* positivo, embora pequeno. Situação análoga é observada no cenário com 12 funcionários e 58 ambientes: mesmo com maior capacidade de recursos humanos, o *solver* atinge o limite de tempo com *gap* residual, indicando que, nessa faixa, o problema se torna efetivamente "duro" do ponto de vista computacional.

O caso real reforça esse diagnóstico. Ao executar o modelo completo para a instância do hospital – composta por 59 funcionários, horizonte mínimo de 15 dias e um número substancialmente maior de ambientes, tarefas e janelas – o *solver* foi executado continuamente por 24 horas. Ainda assim, não houve convergência: o algoritmo encerrou com *gap* elevado e apenas produziu um *upper bound* de 9645,22411 para a função objetivo. Esse valor representa o melhor limite superior encontrado pelo método de *branch-and-bound*, porém sem solução primal associada que pudesse ser considerada ótima ou mesmo próxima da otimalidade. Em termos práticos, significa que, em escala realista, o modelo exato isolado se torna inviável, exigindo tempos de processamento muito superiores às janelas operacionais disponíveis e incompatíveis com uso rotineiro.

Os resultados sintéticos são apresentados na Tabela 9, utilizando o mesmo formato adotado anteriormente:

Tabela 9 – Resultados dos Modelos por Cenário

# Funcionários	**# Ambientes**	**Sem penalidade**		**Com penalidade**	
		Tempo (s)	GAP (%)	Tempo (s)	GAP (%)
6	35	307	–	7200	0,25
6	41	7200	0,29	7200	0,48
12	58	7200	0,18	7200	0,33
12	67	7200	0,32	7200	0,52

Observa-se que pequenas variações na quantidade de ambientes, quando o sistema já opera próximo ao limite da capacidade das equipes, produzem aumentos desproporcionais no tempo de resolução. Nos cenários mais carregados (6-41 e 12-58), o *solver* atinge o tempo limite pré-definido, encerrando com *gap* distinto de zero, ainda que baixo – ou seja, obtém-se uma boa solução, porém sem certificação de otimalidade exata. A inclusão da penalidade por desbalanceamento de esforço e horas agrava consideravelmente a complexidade do modelo. Esse termo adicional na função objetivo introduz dependências adicionais entre variáveis binárias e aumenta a dificuldade de navegação da árvore de decisão, tornando a convergência mais lenta e a região factível mais fragmentada.

Em síntese, o aumento sutil da quantidade de ambientes, quando o sistema já se encontra próximo à saturação, causa um crescimento exponencial no esforço computacional. Além disso, a penalização adicional pelo desbalanceamento de esforço – embora essencial para promover equidade entre profissionais – eleva significativamente a dificuldade de resolução. Assim, para o caso real do hospital, mesmo com 24 horas de processamento contínuo, o modelo exato alcançou apenas um *upper bound* de 9645,22411, tornando evidente que sua utilização isolada é inviável dentro dos prazos operacionais necessários. Dessa forma, a ferramenta só se torna aplicável quando combinada com heurísticas, decomposição de instâncias ou ajustes de escopo que limitem o crescimento da complexidade computacional.

5 HEURÍSTICA

Neste capítulo, é apresentada a heurística de decomposição proposta, desenvolvida para superar a inviabilidade computacional observada na resolução direta do modelo exato em instâncias reais. A estrutura da heurística é dividida em fases, abordando respectivamente a alocação de tarefas longas, a inserção de tarefas curtas e o balanceamento iterativo entre profissionais. O capítulo detalha o racional de cada etapa, discute suas vantagens e descreve o pseudocódigo completo da abordagem. Com isso, busca esclarecer como a heurística reproduz, de forma eficiente, decisões complexas que seriam impraticáveis via otimização pura.

5.1 Aplicação da heurística de decomposição em três fases

A formulação completa representa integralmente o problema de programação das atividades de higienização hospitalar. Entretanto, a resolução direta desse modelo em instâncias reais – que envolvem dezenas de profissionais e centenas de tarefas de curta duração – apresenta elevada complexidade combinatória e tempo computacional proibitivo.

Para mitigar esse problema, foi desenvolvido um método heurístico de decomposição em três fases, inspirado na abordagem de Elahipanah, Desaulniers e Lacasse-Guay (2013), fundamentado nos princípios de Pinedo (2016), e expandido neste trabalho por meio da inclusão de uma terceira fase de balanceamento de carga.

A heurística proposta permite resolver progressivamente subproblemas de diferentes granularidades, mantendo aderência às restrições operacionais, de jornada e de periodicidade. A seguir, apresenta-se a estrutura conceitual e o algoritmo proposto.

5.1.1 Estrutura geral da heurística

A heurística é composta por três etapas hierárquicas e sequenciais:

Fase 1: Alocação estrutural de tarefas longas

Nesta fase, o modelo é restrito às tarefas de maior duração (superiores a 30 minutos). O objetivo é definir a estrutura-base do cronograma, garantindo a cobertura das áreas críticas e ocupando as janelas de tempo principais de cada profissional. Essa etapa é resolvida via modelo de PLIM completo, conforme as equações apresentadas no modelo, porém limitado ao subconjunto:

$$K_{\text{long}} = \{k \in K \mid D_{ak} > 0{,}5\}, \tag{5.1}$$

em que D_{ak} representa a duração (em horas) da tarefa k na área a.

O resultado desta fase é um conjunto de alocações z^*_{ijak} que cobre integralmente as tarefas críticas e gera, para cada profissional i, o tempo total ocupado e as janelas residuais disponíveis (H_i^{livre}).

Fase 2: Inserção de tarefas curtas

Mantendo fixas as decisões estruturais obtidas na Fase 1, a segunda etapa insere as tarefas curtas ($D_{ak} \leq 0{,}5$ h) nas janelas ociosas remanescentes. Para isso, considera-se o subconjunto:

$$K_{\text{short}} = \{k \in K \mid D_{ak} \leq 0{,}5\}, \tag{5.2}$$

que representa todas as tarefas de curta duração elegíveis para preenchimento das lacunas deixadas pelas alocações estruturais da fase anterior.

O subproblema é formulado como:

$$\max \sum_{a,j,k \in K_{\text{short}}} P_{ak} \cdot y_{ajk} \tag{5.3}$$

$$\text{s.a.} \sum_{a,k \in K_{\text{short}}} D_{ak} \cdot z_{ijak} \leq H_i^{\text{livre}}, \quad z_{ijak} \in \{0{,}1\}, \tag{5.4}$$

onde P_{ak} representa a prioridade de cada tarefa. Alternativamente, pode-se empregar uma heurística *greedy*, em que as tarefas curtas são ordenadas por prioridade e inseridas sucessivamente nas janelas disponíveis do profissional com maior capacidade remanescente.

Fase 3: Balanceamento iterativo de carga

Após a alocação completa das tarefas, aplica-se uma etapa de redistribuição iterativa com o objetivo de reduzir a diferença entre os profissionais mais e menos carregados. O processo visa equilibrar o esforço total sem violar as restrições de capacidade diária ou de janela de execução.

A heurística identifica os profissionais $i_{\max}$ e $i_{\min}$ com maior e menor carga total alocada (H_i^{aloc}) e calcula:

$$\Delta = H_{i_{\max}}^{\text{aloc}} - H_{i_{\min}}^{\text{aloc}}. \tag{5.5}$$

Em seguida, busca-se, no conjunto de tarefas atribuídas a $i_{\max}$, aquela cuja duração D_{ak} seja mais próxima de Δ. Caso a transferência dessa tarefa para $i_{\min}$ não viole sua capacidade ($H_{i_{\min}}^{\text{aloc}} + D_{ak} \leq H_i$), a tarefa é realocada. O processo repete-se até que:

$$\max_i \, H_i^{\text{aloc}} - \min_i \, H_i^{\text{aloc}} < \min_{a,k} D_{ak}. \tag{5.6}$$

Essa condição assegura que o desbalanceamento residual é inferior à menor tarefa disponível, momento em que o sistema é considerado balanceado.

5.1.2 Algoritmo proposto

O pseudocódigo a seguir resume a implementação da heurística desenvolvida neste trabalho para resolver as fases baseadas em PLIM.

Tabela 10 – Pseudocódigo da Heurística de Decomposição em Três Fases

	Pseudocódigo da Heurística de Três Fases
1	Definir $K_{\text{long}} \leftarrow \{k \in K \mid D_{ak} > 0.5\ h\}$
2	Definir $K_{\text{short}} \leftarrow \{k \in K \mid D_{ak} \leq 0.5\ h\}$
3	Inicializar $S \leftarrow \emptyset$
— Fase 1: Alocação de tarefas longas —	
4	Resolver modelo MIP restrito a K_{long}
5	Armazenar solução z^* e calcular $H_i^{livre} \leftarrow H_i - \sum_{a,j,k \in K_{\text{long}}} D_{ak} z^*_{ijak}$
— Fase 2: Inserção de tarefas curtas —	
6	Ordenar K_{short} por prioridade decrescente (P_{ak})
7	**Para cada** tarefa $k \in K_{\text{short}}$:
8	Selecionar profissional i com maior H_i^{livre} que possa executar k
9	**Se** $(D_{ak} \leq H_i^{livre})$ então:
10	Atribuir $z_{ijak} = 1$
11	Atualizar $H_i^{livre} \leftarrow H_i^{livre} - D_{ak}$
12	**Fim se**
13	**Fim para**
— Fase 3: Balanceamento iterativo de carga —	
14	Calcular H_i^{aloc} para todos os profissionais $i \in I$
15	**Enquanto** $(\max(H_i^{aloc}) - \min(H_i^{aloc})) \geq \min(D_{ak})$ faça:
16	Identificar $i_{\max} \leftarrow \arg\max(H_i^{aloc})$
17	Identificar $i_{\min} \leftarrow \arg\min(H_i^{aloc})$
18	Calcular $\Delta \leftarrow H_{i_{\max}}^{aloc} - H_{i_{\min}}^{aloc}$
19	Selecionar tarefa $t^* \in S(i_{\max})$ com $D_{t^*} \approx \Delta$
20	**Se** $(H_{i_{\min}}^{aloc} + D_{t^*} \leq H_i)$ então:
21	Reatribuir t^* de $i_{\max}$ para $i_{\min}$
22	Atualizar $H_{i_{\max}}^{aloc}$ e $H_{i_{\min}}^{aloc}$
23	**Fim se**
24	**Fim enquanto**
25	Retornar $S_{\text{final}} = \{z_{ijak}\}$ e métricas de balanceamento

6 RESULTADOS OBTIDOS DOS EXPERIMENTOS NUMÉRICOS

O presente capítulo apresenta e discute os resultados obtidos a partir da aplicação do modelo matemático e da heurística de decomposição desenvolvidos neste trabalho, com foco em avaliar sua eficácia, robustez e aderência às restrições operacionais do Instituto CEMA. Inicialmente, são descritos os indicadores de desempenho relacionados à cobertura das tarefas, ao equilíbrio da carga de trabalho e à eliminação de horas extras, estabelecendo um comparativo direto com o cenário atualmente praticado pela instituição. Em seguida, analisa-se a qualidade da solução gerada pela heurística, incluindo a interpretação da função objetivo, a distribuição final das tarefas entre os profissionais e os efeitos concretos do processo de balanceamento. Por fim, são apresentados gráficos e sínteses que ilustram a coerência temporal das alocações e o comportamento global da solução, permitindo a compreensão de forma clara e detalhada os ganhos operacionais alcançados pela abordagem proposta.

6.1 Resultados da heurística de duas fases com etapa de balanceamento

Após a aplicação completa da heurística de duas fases – composta pela alocação inicial (fase longa-curta) e pelo processo iterativo de balanceamento – obteve-se uma solução final de excelente qualidade, com valor de função objetivo igual a 9̇622,12 comparado a uma *upper bound* de 9645,22. O tempo total de processamento até a convergência da heurística foi de 3h 47min 58s, incluindo a execução das duas fases e as iterações do módulo de reequilíbrio entre profissionais.

6.1.1 Desempenho global e cobertura de tarefas

O resultado final garantiu a cobertura integral das janelas de higienização, contemplando todas as 818 ocorrências requeridas (735 do tipo LC e 83 do tipo LT). Nenhuma tarefa permaneceu não alocada, e todas as restrições de capacidade individual foram respeitadas.

$$\text{Taxa de cobertura} = \frac{818}{818} = 100\%$$

A execução combinada da heurística permitiu explorar com maior eficiência as combinações viáveis de profissional–janela–tarefa, reduzindo substancialmente o tempo de resolução em relação à execução pura do modelo exato, sem comprometer a qualidade da solução.

6.1.2 Distribuição de esforço e etapa de balanceamento

A etapa de balanceamento – terceira fase da heurística – foi responsável por redistribuir pequenas cargas entre os profissionais, transferindo tarefas daqueles com maior acúmulo para aqueles com menor carga, desde que respeitados os limites de disponibilidade e janelas de

execução. Esse processo iterativo foi repetido até que a diferença entre o maior e o menor volume de horas por profissional fosse inferior à menor duração entre as tarefas existentes, conforme o pseudocódigo proposto no capitulo anterior.

A distribuição final de esforço é apresentada na Tabela 11.

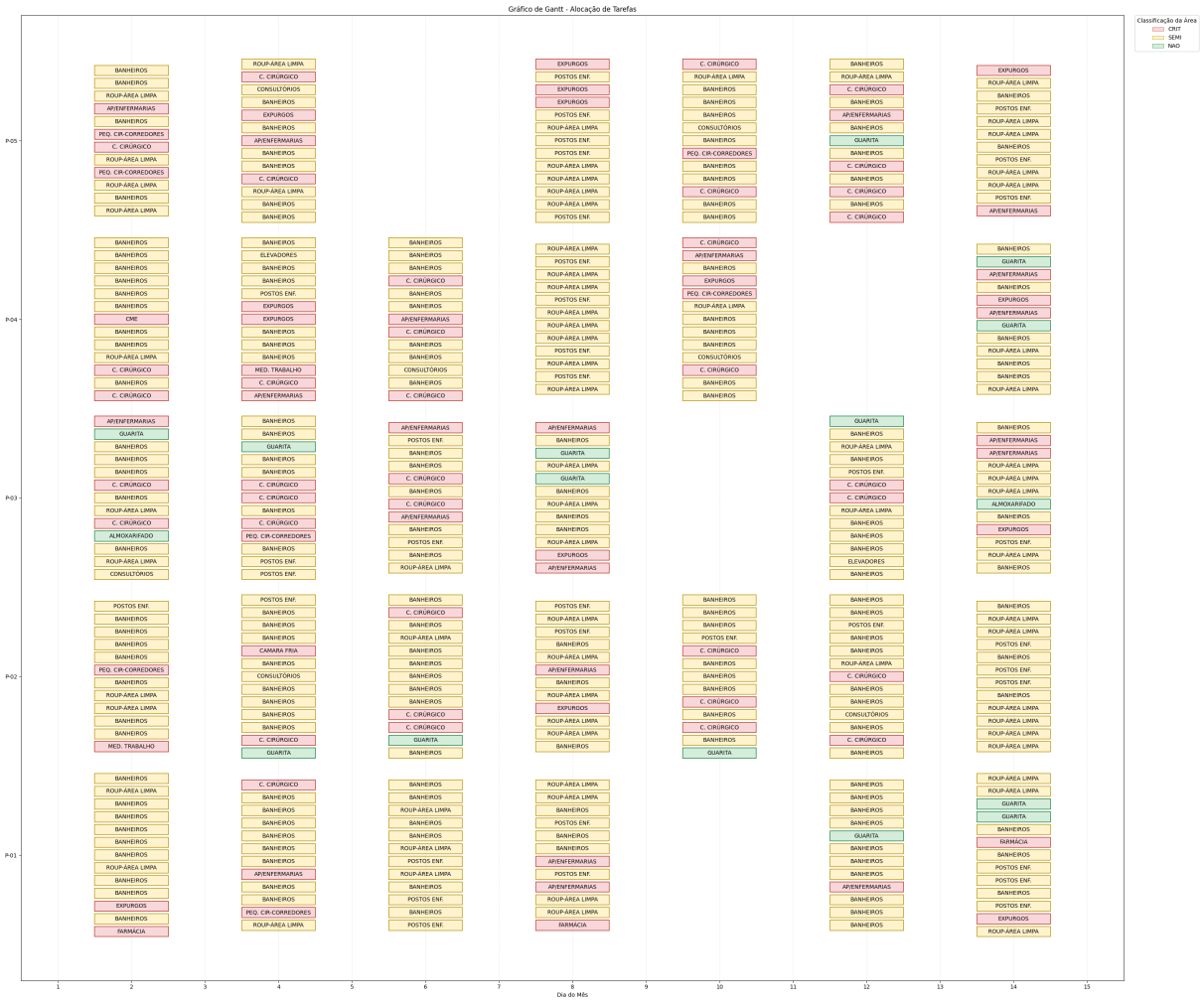
Tabela 11 – Resumo estatístico da carga horária após aplicação da heurística

Estatística	**Valor (h)**
Carga média por profissional	9,91
Carga mínima observada	7,50
Carga máxima observada	11,25
Desvio padrão das cargas	1,18
Diferença Máx–Mín (Δ)	3,75

A variação observada ($\Delta = 3{,}75$ h) evidencia o sucesso da etapa de reequilíbrio: o algoritmo conseguiu reduzir significativamente a discrepância entre os profissionais sem alterar a viabilidade global da solução. Como resultado, obteve-se uma alocação mais uniforme e operacionalmente justa.

A Figura 8 apresenta um recorte ilustrativo do cronograma final – restrito aos profissionais de 01 a 05 – resultante da aplicação completa da heurística de duas fases acrescida da etapa de balanceamento. O gráfico de Gantt permite visualizar, no eixo vertical, a distribuição dos profissionais e, no eixo horizontal, a evolução dos dias ao longo do horizonte de planejamento. A representação evidencia a alocação temporal das tarefas nas janelas permitidas, revelando a coerência entre periodicidade, disponibilidade e restrições operacionais. Observa-se que as atividades foram organizadas de forma a eliminar sobreposições indevidas e a garantir a continuidade dos serviços de higienização, respeitando os limites de turno e as janelas de elegibilidade de cada colaborador. Além disso, a uniformidade visual na ocupação das linhas de trabalho ilustra o impacto direto da fase de balanceamento, que redistribuiu tarefas entre os profissionais, produzindo um planejamento mais homogêneo, equitativo e operacionalmente eficiente.

Figura 8 – Alocação de tarefas final (parcial para demonstração)



Autoria própria.

6.1.3 Impacto na variação de esforço e eliminação de horas extras

No cenário real de operação do hospital, a escala manual de limpeza exige, em média, cerca de 67 horas extras mensais, necessárias para suprir lacunas de cobertura ou compensar ausências. Com a solução obtida pela heurística, todas as tarefas foram atendidas dentro do regime regular de trabalho, sem necessidade de qualquer hora adicional.

Essa eliminação total de horas excedentes representa uma redução de 100% no volume de horas extras historicamente praticadas, com impacto direto na redução de custos e na sustentabilidade operacional.

Adicionalmente, a distribuição de esforço entre os profissionais se mostrou muito mais equilibrada: a diferença entre o trabalhador mais e o menos carregado reduziu-se para menos de 4 horas ao longo de todo o horizonte de 15 dias. Esse comportamento contribui para mitigar a percepção de injustiça na divisão das tarefas, uma vez que o modelo passou a refletir, de forma mais transparente, a equivalência entre os níveis de esforço.

6.1.4 Síntese dos resultados obtidos

A Tabela 12 apresenta uma síntese dos principais indicadores obtidos a partir da aplicação integral da heurística de decomposição – englobando a alocação de tarefas longas, a inserção de tarefas curtas e o módulo de balanceamento – em comparação direta com o cenário real construído manualmente pelos profissionais do hospital. As métricas do cenário real foram obtidas a partir da escala operacional de um mês selecionado para estudo, à qual foram aplicados os mesmos parâmetros definidos na modelagem, como notas de esforço e ponderações de horas. Essa padronização foi necessária, pois tais medidas não eram formalmente registradas pela instituição até o momento, e possibilitou uma comparação justa e homogênea entre os cenários. Cada indicador apresentado reflete dimensões centrais avaliadas neste trabalho, incluindo qualidade da solução, viabilidade computacional e impacto operacional sobre a rotina de higienização. A partir dessa consolidação, desenvolve-se uma análise crítica que contrasta os resultados produzidos pela abordagem proposta com as práticas atualmente adotadas no Instituto CEMA, evidenciando ganhos de eficiência, maior consistência estrutural e melhoria na equidade entre profissionais decorrentes da aplicação das técnicas de otimização.

Tabela 12 – Síntese comparativa dos resultados após aplicação da heurística

Indicador	**Proposto**	**Real**	**Melhoria (%)**
Função Objetivo (FO)	9622,12	9621,16	0,01%
Gap final	0,24%	–	⟨⟩
Tempo total de execução	3h 47min 58s	5h	31%
Tarefas cumpridas	818/818	818/818	—
Horas extras necessárias	0h	67h	100%
Desvio padrão da carga (h)	1,18	2,44	48%
Diferença Máx–Mín (Δ h)	3,75	7,14	48%
Comentário síntese	Redução total de horas extras e maior equidade entre profissionais		

Nota: ⟨⟩ simboliza o valor que não pode ser calculado.

O valor obtido da função objetivo (FO = 9 622,13) representa o melhor equilíbrio entre três componentes essenciais: *(i)* atendimento máximo das janelas e tarefas mandatórias, *(ii)* priorização das áreas de maior criticidade e *(iii)* minimização dos desvios de carga e esforço entre os profissionais. Para fins de comparação, a aplicação da FO aos dados do cenário real atualmente utilizado pelo hospital resulta em FO = 9 621,16. Apesar da proximidade numérica, o cenário real apresenta um desbalanceamento substancialmente maior na distribuição das tarefas entre os profissionais, enquanto a heurística proposta produz uma solução estruturalmente mais equilibrada, convergente e operacionalmente coerente.

O *gap* final de apenas 0,24% demonstra que a solução obtida se encontra muito próxima da fronteira de otimalidade para o modelo reduzido empregado na Fase 1. Esse *gap* foi calculado segundo a definição clássica utilizada em problemas de Programação Inteira Mista, que compara o melhor valor primal encontrado (lower bound) com o melhor limite superior disponível (upper

bound) no encerramento da busca. No caso analisado, o solver registrou um *upper bound* de 9645,22 e uma solução primal de aproximadamente 9622,13, o que corresponde ao *lower bound*. Assim, o *gap* é calculado pela expressão 6.1

$$\text{GAP} = \frac{\text{UB} - \text{LB}}{\text{UB}} \times 100\% = \frac{9645{,}22 - 9622{,}13}{9645{,}22} \times 100\% \approx 0{,}24\%. \tag{6.1}$$

A diferença absoluta entre os limites — cerca de 23 pontos na função objetivo — é extremamente pequena diante da escala do problema, que envolve milhares de variáveis binárias e forte estrutura combinatória. Na literatura de MIP, gaps inferiores a 1% são amplamente reconhecidos como indicadores de soluções de excelente qualidade, especialmente em modelos com restrições de janela, capacidade e precedência. Esse desempenho confirma que a heurística proposta, ao isolar a parte mais densa do problema na Fase 1 e delegar os refinamentos posteriores a operadores mais leves, conseguiu equilibrar profundidade de busca, qualidade da solução e custo computacional. Portanto, o *gap* reduzido não apenas expressa solidez matemática, mas reforça a confiabilidade prática da solução produzida para uso operacional.

O tempo total de 3h 47min 58s inclui a resolução do problema reduzido, a inserção de tarefas curtas e as iterações da etapa de balanceamento. Quando comparado ao processo manual – que demanda aproximadamente 5 horas de dedicação contínua de supervisores experientes –, a heurística apresenta dois benefícios simultâneos: *(i)* redução efetiva de cerca de 31% no tempo dedicado e *(ii)* eliminação de inconsistências associadas ao julgamento humano em cenários com grande número de janelas, ambientes e regras complexas. Além disso, enquanto o processo manual precisa ser reconstruído a cada ciclo, a heurística assegura replicabilidade, auditabilidade e maior padronização.

O modelo atingiu 100% de cobertura das 818 ocorrências previstas – 735 de limpeza concorrente e 83 de limpeza terminal –, alocando plenamente a equipe sem violar restrições de jornada, disponibilidade ou elegibilidade. Nesse sentido, a cobertura total supera o planejamento manual atualmente empregado, que frequentemente demanda ajustes emergenciais ou redistribuição de pessoal no decorrer do mês. Com a solução proposta, todas as tarefas foram distribuídas dentro das horas regulares, eliminando a necessidade das 67 horas extras mensais atualmente utilizadas. Além do impacto direto na redução de custos, o resultados contribui para menor fadiga física, maior aderência às normas trabalhistas e redução de riscos ergonômicos e ocupacionais.

O balanceamento obtido após a terceira fase evidenciou uma melhora substancial na distribuição das cargas. O desvio padrão da carga horária reduziu-se de 2,44 h para 1,18 h (aproximadamente 48%), indicando uma distribuição significativamente mais homogênea entre os profissionais. A diferença Máx–Mín (7,14 h no cenário manual contra 3,75 h após a heurística) confirma que o algoritmo conseguiu mitigar concentrações de carga que geram percepção de injustiça e favorecem o acúmulo de fadiga. Essa diferença final, inferior à duração de uma tarefa

terminal típica, indica que o sistema alcançou o limite estrutural de redistribuição possível sem violar janelas ou regras de elegibilidade.

Os resultados consolidados demonstram que a heurística desenvolvida oferece uma combinação eficaz de rigor matemático, eficiência prática e aderência ao ambiente real da operação. Entre os principais benefícios observados destacam-se: *(i)* cobertura absoluta de tarefas, *(ii)* equidade significativa entre profissionais, *(iii)* eliminação completa de horas extras, *(iv)* solução matematicamente próxima do ótimo e *(v)* tempo de processamento compatível com usos reais. Quando comparada ao comportamento dos modelos exatos, a heurística supera a inviabilidade computacional e entrega resultados adequados em termos de distribuição de carga e robustez operacional. Dessa forma, configura-se como ferramenta viável para a gestão de higienização hospitalar.

7 CONCLUSÃO

O capítulo final reuniu as principais contribuições deste trabalho, destacando avanços, limitações e possibilidades de continuidade. O objetivo central foi analisar se técnicas de Pesquisa Operacional poderiam apoiar um processo atualmente manual – a programação das atividades de higienização hospitalar – e se modelos matemáticos e heurísticas de decomposição conseguiriam produzir resultados utilizáveis na realidade do Instituto CEMA. Ao longo do estudo, ficou claro que a abordagem exata, embora importante como referência conceitual, não é aplicável em escala real. A instância completa do hospital, composta por 59 profissionais, horizonte mínimo de 15 dias e centenas de janelas de execução, permaneceu irresolvida mesmo após 24 horas de processamento contínuo, chegando apenas a um *upper bound* sem certificação de otimalidade. Isso evidencia que o modelo exato, no formato tradicional, não pode ser empregado como ferramenta operacional de rotina.

A heurística de decomposição em três fases mostrou-se, dentro de suas limitações, uma alternativa mais realista e utilizável. Ela não garante soluções ótimas, nem pretende substituir a complexidade intrínseca do processo, mas organiza a tomada de decisão em etapas coerentes com a dinâmica da operação. Ao separar tarefas longas, tarefas curtas e uma etapa de balanceamento, a heurística cria uma estrutura que reduz o número de variáveis binárias envolvidas, tornando o problema tratável. Essa hierarquização das decisões reflete de maneira fiel o fluxo operacional do hospital, em que se priorizam primeiro as áreas críticas e as atividades mais longas, para depois acomodar rotinas complementares e, por fim, ajustar diferenças residuais entre profissionais. Além disso, a aderência prática do método, particularmente devido à integração transparente com planilhas Excel, facilita sua adoção por equipes de hotelaria, que podem compreender, validar e ajustar as decisões geradas de forma mais intuitiva.

Os resultados obtidos foram positivos, mas devem ser interpretados de forma equilibrada. A heurística conseguiu alocar 100% das 818 tarefas previstas no horizonte, sem violar janelas e sem gerar horas extras – o que representa, no contexto real, a eliminação de cerca de 67 horas mensais de sobrecarga. Também foi possível reduzir significativamente o desbalanceamento de carga entre os profissionais, promovendo uma distribuição mais homogênea, embora não isenta de assimetrias. Esses ganhos não devem ser vistos como solução definitiva, mas como um avanço incremental e consistente: a metodologia melhora a qualidade da alocação e reduz a dependência de julgamentos manuais, sem eliminar completamente a necessidade de supervisão.

Do ponto de vista institucional, o trabalho contribui ao transformar um processo empírico em um procedimento sistematizado, reproduzível e mais transparente. Essa transição não substitui o papel da experiência humana, mas cria um ambiente mais previsível e menos sujeito a vieses, reforçando a confiabilidade das operações de higienização – que têm impacto direto na segurança do paciente e na eficiência hospitalar.

Quanto às extensões futuras, há espaço significativo para evolução. Uma linha evidente é a incorporação de incertezas, especialmente relacionadas às limpezas imediatas e ao absenteísmo, por meio de modelos estocásticos ou abordagens híbridas com simulação. Trabalhos como o de Wu *et al.* (2023), que integram incertezas em problemas de escalonamento hospitalar, oferecem base conceitual promissora para essa evolução. Outras oportunidades incluem a integração do modelo com sistemas de informação hospitalares, permitindo atualização automática de janelas e demandas; o uso de técnicas de aprendizado de máquina para prever durações reais das tarefas e flutuações de demanda; e o desenvolvimento de heurísticas adaptativas que refinam iterativamente a solução sem aumentar significativamente o tempo de processamento.

Em síntese, o trabalho demonstrou que métodos de otimização e heurísticas de decomposição, quando aplicados de forma criteriosa e adaptada à realidade operacional, podem tornar o processo de higienização hospitalar mais estruturado, equilibrado e sustentável. A solução proposta não resolve integralmente o problema, mas representa um avanço prático e aplicável, evidenciando o potencial da Engenharia de Produção para apoiar decisões em ambientes complexos de saúde.

REFERÊNCIAS

ANVISA. **Boletim Segurança do Paciente e Qualidade em Serviços de Saúde nº 32 – Avaliação Nacional dos indicadores de IRAS e RM - 2024**. 2024. Painel Interativo – Power BI. Acesso em: 29 jul. 2025. Disponível em: https://www.gov.br/anvisa/pt-br/centraisdeconteudo/publicacoes/servicosdesaude/boletins-e-relatorios-das-notificacoes-de-iras-e-outros-eventos-adversos-1/boletins-e-relatorios-das-notificacoes-de-iras-e-outros-eventos-adversos.

AZAIEZ, N.; SHARIF, S. A 0-1 goal programming model for nurse scheduling. **Computers Operations Research**, v. 32, p. 491–507, 03 2005.

BILGIN, B. *et al.* Local search neighbourhoods for dealing with a novel nurse rostering model. **Annals OR**, v. 194, p. 33–57, 04 2012.

BOULDING, W. *et al.* A relationship-centered model of patient satisfaction: patients' perspectives and the patient-physician relationship. **Medical Care Research and Review**, SAGE Publications, v. 68, n. 2, p. 156–180, 2011.

BRASIL. **Norma Regulamentadora nº 32 (NR-32): Segurança e Saúde no Trabalho em Serviços de Saúde**. Brasília, DF: Ministério do Trabalho e Previdência, 2022. Atualizada pela Portaria MTP nº 806, de 13 de abril de 2022.

CAUSMAECKER, P. D.; BERGHE, G. V. A categorisation of nurse rostering problems. **Journal of Scheduling**, v. 14, n. 1, p. 3–16, 2011.

COELLO, C. A. C. A comprehensive survey of evolutionary-based multiobjective optimization techniques. **Knowledge and Information Systems**, v. 1, n. 3, p. 269–308, 1999.

DIAS, T. M. *et al.* Constructing nurse schedules at large hospitals. **International Transactions in Operational Research**, v. 10, p. 245–265, 2003. ISSN 1475-3995. Disponível em: http://dx.doi.org/10.1111/1475-3995.00406.

DOHN, A.; KOLIND, E.; CLAUSEN, J. The manpower allocation problem with time windows and job-teaming constraints: A branch-and-price approach. **Computers Operations Research**, v. 36, n. 4, p. 1145–1157, 2009. ISSN 0305-0548. Disponível em: https://www.sciencedirect.com/science/article/pii/S030505480700278X.

DONSKEY, C. J. Does improving surface cleaning and disinfection reduce health care-associated infections? **American Journal of Infection Control**, v. 41, n. 5, Supplement, p. S12–S19, 2013. Disponível em: https://doi.org/10.1016/j.ajic.2012.12.010.

DUENAS, A.; TüTüNCü, G.; CHILCOTT, J. A genetic algorithm approach to the nurse scheduling problem with fuzzy preferences. **IMA Journal of Management Mathematics**, v. 20, 10 2009.

ELAHIPANAH, M.; DESAULNIERS, G.; LACASSE-GUAY Ève. A two-phase mathematical-programming heuristic for flexible assignment of activities and tasks to work shifts. **Journal of Scheduling**, Springer Science+Business Media New York, v. 16, n. 4, p. 443–460, 2013. Disponível em: https://doi.org/10.1007/s10951-013-0324-2.

FANG, K. *et al.* Flow shop scheduling with peak power consumption constraints. **Annals of Operations Research**, v. 206, p. 115–145, 2013.

GAREY, M. R.; JOHNSON, D. S. **Computers and Intractability: A Guide to the Theory of NP-Completeness**. New York: W. H. Freeman and Company, 1979.

GLICKMAN, S. W. *et al.* Patient satisfaction and its relationship with clinical quality and inpatient mortality in acute myocardial infarction. **Circulation: Cardiovascular Quality and Outcomes**, American Heart Association, v. 3, n. 2, p. 188–195, mar. 2010. Epub 2010 Feb 23.

GOMES, M. I.; RAMOS, T. R. P. Modelling and (re-)planning periodic home social care services with loyalty and non-loyalty features. **European Journal of Operational Research**, v. 277, n. 1, p. 284–299, 2019.

GRAHAM, R. L. *et al.* Optimization and approximation in deterministic sequencing and scheduling: a survey. **Annals of Discrete Mathematics**, v. 4, p. 287–326, 1979.

INSTITUTO CEMA. **Manual de Hotelaria**. São Paulo, 2023. Documento interno.

ISERMANN, H. Linear lexicographic optimization. **OR Spektrum**, v. 4, n. 4, p. 223–228, 1982.

KRAMER, A.; SCHWEBKE, I.; KAMPF, G. How long do nosocomial pathogens persist on inanimate surfaces? a systematic review. **BMC Infectious Diseases**, v. 6, n. 130, 2006. Disponível em: https://doi.org/10.1186/1471-2334-6-130.

LIM, A.; RODRIGUES, B.; SONG, L. Manpower allocation with time windows. **Journal of the Operational Research Society**, v. 55, 06 2004.

MANSOURI, S. A.; AKTAS, E.; BESIKCI, U. Green scheduling of a two-machine flowshop: Trade-off between makespan and energy consumption. **European Journal of Operational Research**, v. 248, n. 3, p. 772–788, 2016.

MOSEK APS. Mosek modeling cookbook. Copenhagen, Denmark, p. 110–111, 2025.

PINEDO, M.; ZACHARIAS, C.; ZHU, N. Scheduling in the service industries: An overview. **Journal of Systems Science and Systems Engineering**, Systems Engineering Society of China and Springer-Verlag Berlin Heidelberg, v. 24, n. 1, p. 1–48, mar. 2015. ISSN 1004-3756.

PINEDO, M. L. **Scheduling: Theory, Algorithms, and Systems**. 5th. ed. Cham: Springer, 2016. ISBN 978-3-319-26578-0. Disponível em: https://doi.org/10.1007/978-3-319-26580-3.

WEINBROUM, A. A.; EKSTEIN, P.; EZRI, T. Efficiency of the operating room suite. **American Journal of Surgery**, v. 185, n. 3, p. 244–250, mar. 2003.

WORLD HEALTH ORGANIZATION. **Report on the Burden of Endemic Health Care-Associated Infection Worldwide**. Geneva: World Health Organization, 2011.

WU, Z. *et al.* Two scenario-based heuristics for stochastic shift design problem with task-based demand. **Applied Sciences**, MDPI, v. 13, n. 18, p. 10070, 2023. Disponível em: https://doi.org/10.3390/app131810070.

APÊNDICES

APÊNDICE A – CÓDIGO DO MODELO

```
import os
import pandas as pd
from gurobipy import Model, GRB, quicksum

# ========================
# 1) Caminho do Excel
# ========================
xlsx_path = r"content/OFICIAL 15d.xlsx"

# ========================
# 1.1) Perfil de parâmetros (otimização Gurobi)
# ========================
def set_fast_params(m: Model, foco: str = "gap"):
    """
    Conjunto de parâmetros com bom custo/benefício.
    foco: 'viavel' | 'gap' | 'equilibrado'
    """
    # CPU e paralelismo
    m.Params.OutputFlag    = 1
    m.Params.Threads       = 0
    m.Params.ConcurrentMIP = 1

    # Presolve e agregação
    m.Params.Presolve      = 2
    m.Params.Aggregate     = 2
    m.Params.PreSparsify   = 1
    m.Params.PrePasses     = 10

    # Cortes & simetria
    m.Params.Cuts          = 2
    m.Params.ZeroHalfCuts  = 2
    m.Params.CliqueCuts    = 2
    m.Params.CoverCuts     = 2
    m.Params.Symmetry      = 2

    # Heurísticas e foco
    m.Params.Heuristics    = 0.2
    m.Params.MIPFocus      = {"viavel": 2, "gap": 1, "equilibrado": 0}.get(
        foco, 1)

    # Memória (spill da árvore)
    m.Params.NodefileStart = 0.5  # em GB

# ========================
# 2) Leitura e normalização
# ========================
def check_cols(df, required, sheet):
    miss = [c for c in required if c not in df.columns]
    if miss:
        raise ValueError(f"Aba '{sheet}' faltando colunas: {miss}")

# --- Profissionais
prof = pd.read_excel(xlsx_path, sheet_name="Profissionais")
check_cols(prof, ["professional_id", "H_i", "shift_type"], "Profissionais")
```

```
prof["professional_id"] = prof["professional_id"].astype(str).str.strip()
prof["H_i"] = pd.to_numeric(prof["H_i"], errors="coerce").fillna(0.0).
    astype(float)
prof["shift_type"] = prof["shift_type"].astype(str).str.strip().str.upper()

# --- Disponibilidade
disp = pd.read_excel(xlsx_path, sheet_name="Disponibilidade")
check_cols(disp, ["professional_id", "day", "works"], "Disponibilidade")
disp["professional_id"] = disp["professional_id"].astype(str).str.strip()
disp["day"] = pd.to_numeric(disp["day"], errors="coerce").fillna(1).astype(
    int)
disp["works"] = pd.to_numeric(disp["works"], errors="coerce").fillna(0).
    astype(int).clip(0, 1)

# --- Áreas (subárea -> classe + numero_ambiente=MULTIPLICIDADE)
areas = pd.read_excel(xlsx_path, sheet_name="Areas")
check_cols(areas, ["subarea_id", "area_class"], "Areas")  # numero_ambiente
    é opcional
if "numero_ambiente" not in areas.columns:
    areas["numero_ambiente"] = 1  # default multiplicidade
areas["subarea_id"] = areas["subarea_id"].astype(str).str.strip().str.upper
    ()
areas["area_class"] = areas["area_class"].astype(str).str.strip().str.upper
    ()
areas["numero_ambiente"] = pd.to_numeric(areas["numero_ambiente"], errors="
    coerce").fillna(1).astype(int).clip(lower=1)

# Mapas básicos (por subárea base)
S_base = sorted(areas["subarea_id"].unique().tolist())
sub2class_base = {r["subarea_id"]: r["area_class"] for _, r in areas.
    iterrows()}
sub2mult_base  = {r["subarea_id"]: int(r["numero_ambiente"]) for _, r in
    areas.iterrows()}

# --- Tarefas
tasks = pd.read_excel(xlsx_path, sheet_name="Tarefas")
check_cols(tasks, ["area_id", "task_type_id", "duration_hours", "
    effort_weight", "priority_weight"], "Tarefas")
for c in ["forbid_manha", "forbid_noite"]:
    if c not in tasks.columns:
        tasks[c] = 0

# Normalização
tasks["area_id"]      = tasks["area_id"].astype(str).str.strip().str.upper
    ()
tasks["task_type_id"] = tasks["task_type_id"].astype(str).str.strip().str.
    upper()
tasks["duration_hours"]  = pd.to_numeric(tasks["duration_hours"],  errors="
    coerce").fillna(0.0)
tasks["effort_weight"]   = pd.to_numeric(tasks["effort_weight"],   errors="
    coerce").fillna(0.0)
tasks["priority_weight"] = pd.to_numeric(tasks["priority_weight"], errors="
    coerce").fillna(0.0)
tasks["forbid_manha"]    = pd.to_numeric(tasks["forbid_manha"],    errors="
    coerce").fillna(0).astype(int).clip(0,1)
tasks["forbid_noite"]    = pd.to_numeric(tasks["forbid_noite"],    errors="
    coerce").fillna(0).astype(int).clip(0,1)

# Dicionários: (1) por SUBÁREA base; (2) por CLASSE
```

```
D_sub = {}; E_sub = {}; P_sub = {}; FORB_sub = {}
for _, r in tasks.iterrows():
    aid = r["area_id"]; k = r["task_type_id"]
    if aid in S_base:
        D_sub[(aid, k)] = float(r["duration_hours"])
        E_sub[(aid, k)] = float(r["effort_weight"])
        P_sub[(aid, k)] = float(r["priority_weight"])
        FORB_sub[(aid, k)] = (int(r["forbid_manha"]), int(r["forbid_noite"
            ]))

classes = set(areas["area_class"].unique().tolist())
D_cls = {}; E_cls = {}; P_cls = {}; FORB_cls = {}
for _, r in tasks.iterrows():
    aid = r["area_id"]; k = r["task_type_id"]
    if aid in classes:
        D_cls[(aid, k)] = float(r["duration_hours"])
        E_cls[(aid, k)] = float(r["effort_weight"])
        P_cls[(aid, k)] = float(r["priority_weight"])
        FORB_cls[(aid, k)] = (int(r["forbid_manha"]), int(r["forbid_noite"
            ]))

# Helpers (subárea base → pega param; unidades herdam)
def getD(s_base, k):
    if (s_base, k) in D_sub: return D_sub[(s_base, k)]
    ac = sub2class_base[s_base]
    if (ac, k) in D_cls: return D_cls[(ac, k)]
    raise KeyError(f"Sem duração definida para (subarea={s_base}, tarefa={k
        }).")

def getE(s_base, k):
    if (s_base, k) in E_sub: return E_sub[(s_base, k)]
    ac = sub2class_base[s_base]
    if (ac, k) in E_cls: return E_cls[(ac, k)]
    raise KeyError(f"Sem esforço definido para (subarea={s_base}, tarefa={k
        }).")

def getP(s_base, k):
    if (s_base, k) in P_sub: return P_sub[(s_base, k)]
    ac = sub2class_base[s_base]
    if (ac, k) in P_cls: return P_cls[(ac, k)]
    return 0.0

def getFORB(s_base, k):
    if (s_base, k) in FORB_sub: return FORB_sub[(s_base, k)]
    ac = sub2class_base[s_base]
    return FORB_cls.get((ac, k), (0, 0))

# --- Janelas
wins = pd.read_excel(xlsx_path, sheet_name="Janelas")
check_cols(wins, ["subarea_id", "task_type_id", "required"], "Janelas")
for c in ["start_day", "end_day", "forbid_manha", "forbid_noite", "
    crew_size"]:
    if c not in wins.columns:
        wins[c] = 0
wins["subarea_id"]    = wins["subarea_id"].astype(str).str.strip().str.
    upper()
wins["task_type_id"]  = wins["task_type_id"].astype(str).str.strip().str.
    upper()
wins["required"]      = pd.to_numeric(wins["required"],  errors="coerce").
```

```
    fillna(0).astype(int).clip(0, None)
wins["start_day"]     = pd.to_numeric(wins["start_day"], errors="coerce").
    fillna(0).astype(int)
wins["end_day"]       = pd.to_numeric(wins["end_day"],   errors="coerce").
    fillna(0).astype(int)
wins["forbid_manha"]  = pd.to_numeric(wins["forbid_manha"], errors="coerce"
    ).fillna(0).astype(int).clip(0,1)
wins["forbid_noite"]  = pd.to_numeric(wins["forbid_noite"], errors="coerce"
    ).fillna(0).astype(int).clip(0,1)
wins["crew_size"]     = pd.to_numeric(wins["crew_size"], errors="coerce").
    fillna(1).astype(int).clip(1, None)

# =========================
# 3) Conjuntos, horizonte, disponibilidade
# =========================
I = prof["professional_id"].unique().tolist()

max_end = int(max(wins["end_day"].max(), 0)) if not wins.empty else 0
max_day = max([disp["day"].max() if not disp.empty else 0, max_end])
max_day = int(max(1, max_day))
J = sorted(set(disp["day"]).union(set(range(1, max_day + 1))))

Hi = {r["professional_id"]: float(r["H_i"]) for _, r in prof.iterrows()}
SHIFT = {r["professional_id"]: r["shift_type"] for _, r in prof.iterrows()}

# Disponibilidade W[(i,j)]
W = {(i, j): 0 for i in I for j in J}
for _, r in disp.iterrows():
    i, j, wv = r["professional_id"], int(r["day"]), int(r["works"])
    if i in I and j in J:
        W[(i, j)] = 1 if wv >= 1 else 0

# =========================
# 3.1) Expandir subáreas por multiplicidade
# =========================
S_units = []
unit2base = {}
unit2idx  = {}
class_unit = {}
mult_rows = []

for s_base in S_base:
    M = sub2mult_base[s_base]
    for u in range(1, M+1):
        s_unit = f"{s_base}{u}"
        S_units.append(s_unit)
        unit2base[s_unit] = s_base
        unit2idx[s_unit]  = u
        class_unit[s_unit] = sub2class_base[s_base]
        mult_rows.append({"base_subarea_id": s_base, "ambiente_idx": u})

# =========================
# 3.2) Expansão de janelas por UNIDADE (multiplicidade)
# =========================
crit_alias = {"CRIT", "CRÍTICA", "CRITICA", "CRÍTICO", "CRITICO"}

Wins = {}
minJ = min(J) if J else 1
maxJ = max(J) if J else 1
```

```
def dow_1_sun(j: int) -> int:
    return ((int(j) - 1) % 7) + 1

working_days = [j for j in J if dow_1_sun(j) in {2, 3, 4, 5, 6}]
all_days = J[:]

for _, r in wins.iterrows():
    s_base = r["subarea_id"]
    k      = r["task_type_id"]
    fm     = int(r.get("forbid_manha", 0))
    fn     = int(r.get("forbid_noite", 0))
    crew   = int(r.get("crew_size", 1))

    start = int(r.get("start_day", 0)) or minJ
    end   = int(r.get("end_day",   0)) or maxJ
    req   = int(r.get("required",  0))

    M = sub2mult_base.get(s_base, 1)
    for u in range(1, M+1):
        s_unit = f"{s_base}{u}"
        Wins.setdefault(s_unit, [])

        if k == "LC":
            area_cls = class_unit[s_unit]
            if area_cls in crit_alias:
                for j in all_days:
                    Wins[s_unit].append({
                        "start": j, "end": j, "req": 1,
                        "forbid_m": fm, "forbid_n": fn,
                        "task_type": k, "crew": crew
                    })
            else:
                for j in working_days:
                    Wins[s_unit].append({
                        "start": j, "end": j, "req": 0,
                        "forbid_m": fm, "forbid_n": fn,
                        "task_type": k, "crew": crew
                    })
        else:
            Wins[s_unit].append({
                "start": start, "end": end, "req": req,
                "forbid_m": fm, "forbid_n": fn,
                "task_type": k, "crew": crew
            })

# Prévia das janelas expandidas
preview_rows = []
for s_unit, lst in Wins.items():
    for w in lst:
        preview_rows.append({
            "base_subarea_id": unit2base[s_unit],
            "ambiente_idx": unit2idx[s_unit],
            "area_class": class_unit[s_unit],
            "task_type_id": w["task_type"],
            "window_start": w["start"],
            "window_end": w["end"],
            "window_req": w["req"],
            "forbid_manha": w["forbid_m"],
```

```
            "forbid_noite": w["forbid_n"],
            "crew_size": w["crew"]
        })

preview_df = pd.DataFrame(preview_rows)
mult_df_pre = pd.DataFrame(mult_rows)

preview_out = xlsx_path.replace(".xlsx", "_preview_janelas_expandidas.xlsx"
    )
os.makedirs(os.path.dirname(preview_out) or ".", exist_ok=True)
with pd.ExcelWriter(preview_out, engine="openpyxl") as writer:
    preview_df.to_excel(writer, sheet_name="Wins_Expandidas", index=False)
    mult_df_pre.to_excel(writer, sheet_name="Ambientes_Expandidos_Pre",
        index=False)
print(f"\ n     Prévia de janelas expandidas salva em: {os.path.abspath(
    preview_out)}", flush=True)

# =========================
# 3.3) Validação
# =========================
missing = []
for s_unit in S_units:
    s_base = unit2base[s_unit]
    for w in Wins.get(s_unit, []):
        k = w["task_type"]
        try:
            _ = getD(s_base, k); _ = getE(s_base, k); _ = getP(s_base, k);
                _ = getFORB(s_base, k)
        except KeyError:
            missing.append((s_base, k))
        L = int(w["end"]) - int(w["start"]) + 1
        if int(w["req"]) > L:
            raise ValueError(
                f"'required' ({w['req']}) maior que o tamanho da janela ({L
                    }) "
                f"em {s_unit}, tarefa {w['task_type']}, {w['start']}..{w['
                    end']}."
            )

if missing:
    raise ValueError(
        "Parâmetros ausentes em 'Tarefas' para: "
        + ", ".join([f"({s},{k})" for s, k in sorted(set(missing))])
    )

# =========================
# 3.4) Helpers de turno / proibições
# =========================
def eff_shift(i):
    s = str(SHIFT.get(i, "MANHA")).upper()
    if s in ["5X2", "ANY", "AMBOS", "BOTH", "*"]:
        return "ANY"
    if s in ["NOITE", "NIGHT"]:
        return "NOITE"
    return "MANHA"

def is_forbidden_task(s_unit, k, s_eff):
    if s_eff == "ANY":
        return False
```

```
    s_base = unit2base[s_unit]
    forb_m, forb_n = getFORB(s_base, k)
    return (s_eff == "MANHA" and forb_m == 1) or (s_eff == "NOITE" and
        forb_n == 1)

def is_forbidden_window(win, s_eff):
    if s_eff == "ANY":
        return False
    return (s_eff == "MANHA" and win["forbid_m"] == 1) or (s_eff == "NOITE"
        and win["forbid_n"] == 1)

# =========================
# 3.5) Auditoria & prints
# =========================
def pretty(n):
    try:
        return f"{n:,.0f}".replace(",", ".")
    except Exception:
        return str(n)

def compute_pre_kpis():
    SJk_tmp = []
    for s_unit in S_units:
        for w in Wins.get(s_unit, []):
            for j in range(w["start"], w["end"] + 1):
                SJk_tmp.append((s_unit, j, w["task_type"]))
    SJk_tmp = sorted(list(set(SJk_tmp)))

    req_by_type = {}
    hours_task_by_type = {}
    hours_person_by_type = {}
    count_SJk_by_type = {}

    for s_unit in S_units:
        s_base = unit2base[s_unit]
        for w in Wins.get(s_unit, []):
            k = w["task_type"]
            dur = getD(s_base, k)
            req = int(w["req"])
            crew = int(w["crew"])
            req_by_type[k] = req_by_type.get(k, 0) + req
            hours_task_by_type[k] = hours_task_by_type.get(k, 0.0) + req *
                dur
            hours_person_by_type[k] = hours_person_by_type.get(k, 0.0) +
                req * dur * crew

    for (_, _, k) in SJk_tmp:
        count_SJk_by_type[k] = count_SJk_by_type.get(k, 0) + 1

    H_total_pre = sum(Hi[i] * sum(1 for j in J if W[(i, j)] == 1) for i in
        I) or 1.0

    shift_map = {"MANHA": 0, "NOITE": 0, "ANY": 0}
    for i in I:
        shift_map[eff_shift(i)] = shift_map.get(eff_shift(i), 0) + 1

    print("\n========== AUDITORIA PRÉ-MODELO ==========", flush=True)
    print(f"- Ambientes base: {len(S_base)} | Ambientes expandidos: {len(
        S_units)}", flush=True)
```

```
    print(f"- Horizonte: {min(J)}..{max(J)} (|J|={len(J)})", flush=True)
    print(f"- Linhas     Profissionais: {len(prof)}, Disponibilidade: {len(
        disp)}, Áreas: {len(areas)}, Tarefas: {len(tasks)}, Janelas: {len(
        wins)}", flush=True)
    n_lc = req_by_type.get("LC", 0)
    n_lt = req_by_type.get("LT", 0)
    print(f"- Ocorrências requeridas     LC: {pretty(n_lc)} | LT: {pretty(
        n_lt)} | Outros: {pretty(sum(v for k,v in req_by_type.items() if k
        not in ['LC','LT']))}", flush=True)
    tot_hours_task = sum(hours_task_by_type.values())
    tot_hours_person = sum(hours_person_by_type.values())
    print(f"- Horas (tarefa): {tot_hours_task:,.2f}".replace(",", "."),
        flush=True)
    print(f"- Horas-pessoa (com crew): {tot_hours_person:,.2f}".replace(","
        , "."), flush=True)
    print(f"- Capacidade H_total: {H_total_pre:,.2f}".replace(",", "."),
        flush=True)
    print(f"- Profissionais por turno → MANHA: {shift_map.get('MANHA',0)} |
         NOITE: {shift_map.get('NOITE',0)} | ANY: {shift_map.get('ANY',0)}",
         flush=True)
    print(f"- (s,j,k) possíveis (antes de disponibilidade): {pretty(len(
        SJk_tmp))}", flush=True)
    for k in sorted(count_SJk_by_type.keys()):
        print(f"    - {k}: {pretty(count_SJk_by_type[k])} combinações (s,j)"
            , flush=True)

compute_pre_kpis()

# ==========================
# 4) Índices e domínios viáveis (com disponibilidade/turno)
# ==========================
SJk = []
win_map = {}
for s_unit in S_units:
    for w in Wins.get(s_unit, []):
        for j in range(w["start"], w["end"] + 1):
            t = (s_unit, j, w["task_type"])
            SJk.append(t)
            win_map[t] = w
SJk = sorted(list(set(SJk)))

# x só quando é possível (i trabalha e turno permitido)
x_keys = []
allowed = {}
for (s_unit, j, k) in SJk:
    w = win_map[(s_unit, j, k)]
    tmp = []
    for i in I:
        if W[(i, j)] != 1:
            continue
        s_eff = eff_shift(i)
        if is_forbidden_task(s_unit, k, s_eff) or is_forbidden_window(w,
            s_eff):
            continue
        tmp.append(i)
        x_keys.append((i, s_unit, j, k))
    allowed[(s_unit, j, k)] = tmp

# ===== Poda estrutural: (s,j,k) inviáveis se crew > |allowed| =====
```

```
feasible_SJk = []
x_keys_filtered = []
for (s_unit, j, k) in SJk:
    crew = int(win_map[(s_unit, j, k)]["crew"])
    al = allowed[(s_unit, j, k)]
    if len(al) >= crew and len(al) > 0:
        feasible_SJk.append((s_unit, j, k))
        for i in al:
            x_keys_filtered.append((i, s_unit, j, k))
    else:
        allowed[(s_unit, j, k)] = []

SJk = feasible_SJk
x_keys = x_keys_filtered

# Agrupar para capacidade (i,j) e somatórios por i
by_ij = {}
for (i, s_unit, j, k) in x_keys:
    by_ij.setdefault((i, j), []).append((s_unit, k))

by_i = {}
for (i, s_unit, j, k) in x_keys:
    by_i.setdefault(i, []).append((s_unit, j, k))

# Print tamanho do modelo após filtros
print("\n========== TAMANHO DO MODELO ==========", flush=True)
print(f"- |SJk| (variáveis y): {pretty(len(SJk))}", flush=True)
print(f"- |x| (variáveis de alocação): {pretty(len(x_keys))}", flush=True)
print(f"- Restrições de cobertura: {pretty(len(SJk))}", flush=True)
print(f"- Restrições de capacidade por (i,j): {pretty(len(by_ij))}", flush=
    True)

# =========================
# 5) Modelo
# =========================
m = Model("Higienizacao_Janelas_Multiplicidade")

# Variáveis
y = m.addVars(SJk, vtype=GRB.BINARY, name="y")
x = m.addVars(x_keys, vtype=GRB.BINARY, name="z")

delta   = m.addVars(I, lb=0.0, name="delta")
epsilon = m.addVars(I, lb=0.0, name="epsilon")
Hbar = m.addVar(lb=0.0, name="Hbar")
Ebar = m.addVar(lb=0.0, name="Ebar")

# Cobertura com TAMANHO DE EQUIPE simultânea: sum_i x = crew * y
for (s_unit, j, k) in SJk:
    crew = int(win_map[(s_unit, j, k)]["crew"])
    m.addConstr(
        quicksum(x[i, s_unit, j, k] for i in allowed[(s_unit, j, k)]) ==
            crew * y[s_unit, j, k],
        name=f"cover_{s_unit}{j}{k}"
    )

# Capacidade por (i,j)
for (i, j), pairs in by_ij.items():
    m.addConstr(
        quicksum(x[i, s_unit, j, k] * getD(unit2base[s_unit], k) for (
```

```
            s_unit, k) in pairs) <= Hi[i],
        name=f"cap_{i}_{j}"
    )

# Balanceamento (média e desvios absolutos)
tot_hours  = quicksum(x[i, s_unit, j, k] * getD(unit2base[s_unit], k) for (
    i, s_unit, j, k) in x_keys)
tot_effort = quicksum(x[i, s_unit, j, k] * getE(unit2base[s_unit], k) for (
    i, s_unit, j, k) in x_keys)
m.addConstr(tot_hours  == len(I) * Hbar, name="link_Hbar")
m.addConstr(tot_effort == len(I) * Ebar, name="link_Ebar")

for i in I:
    pairs = by_i.get(i, [])
    work_i = quicksum(x[i, s_unit, j, k] * getD(unit2base[s_unit], k) for (
        s_unit, j, k) in pairs) if pairs else 0.0
    eff_i  = quicksum(x[i, s_unit, j, k] * getE(unit2base[s_unit], k) for (
        s_unit, j, k) in pairs) if pairs else 0.0
    m.addConstr(work_i - Hbar <= delta[i], name=f"balH_pos_{i}")
    m.addConstr(Hbar - work_i <= delta[i], name=f"balH_neg_{i}")
    m.addConstr(eff_i - Ebar <= epsilon[i], name=f"balE_pos_{i}")
    m.addConstr(Ebar - eff_i <= epsilon[i], name=f"balE_neg_{i}")

# Requisitos por janela (mínimo)
for s_unit in S_units:
    for w in Wins.get(s_unit, []):
        rng = [j for j in range(w["start"], w["end"] + 1)]
        req = int(w["req"])
        cand = [j for j in rng if (s_unit, j, w["task_type"]) in y]
        if req > 0 and cand:
            m.addConstr(quicksum(y[s_unit, j, w["task_type"]] for j in cand
                ) >= req,
                        name=f"win_min_{s_unit}{w['start']}{w['end']}")

# H_total / E_total p/ KPIs
H_total = sum(Hi[i] * sum(1 for j in J if W[(i, j)] == 1) for i in I) or
    1.0
E_total = sum(
    w["req"] * getE(unit2base[s_unit], w["task_type"])
    for s_unit in S_units for w in Wins.get(s_unit, [])
) or 1.0

# Função-Objetivo
obj = quicksum(getP(unit2base[s_unit], k) * y[s_unit, j, k] for (s_unit, j,
    k) in SJk)
m.setObjective(obj, GRB.MAXIMIZE)

# ========================
# 5.1) Parâmetros + otimização robusta (INF_OR_UNBD)
# ========================
print("\n========== OTIMIZAÇÃO ==========", flush=True)
set_fast_params(m, foco="gap")
m.optimize()

if m.Status == GRB.INF_OR_UNBD:
    print("- Status INF_OR_UNBD     reotimizando com DualReductions=0...",
        flush=True)
    m.setParam("DualReductions", 0)
    m.optimize()
```

```
# ========================
# 5.2) Logs de resultado
# ========================
try:
    solcount = getattr(m, "SolCount", 0)
    print("\n========== RESULTADO ==========", flush=True)
    print(f"- Status do modelo: {m.Status}", flush=True)
    print(f"- Solutions found (SolCount): {solcount}", flush=True)
    if m.Status in [GRB.OPTIMAL, GRB.INTERRUPTED] and solcount > 0:
        print(f"- Valor da função-objetivo: {m.objVal:,.6f}".replace(",", "
            ."), flush=True)
except Exception:
    pass

# ========================
# 6) Exportação p/ Excel
# ========================
def export_excel():
    out_path = xlsx_path.replace(".xlsx", "_resultado.xlsx")
    os.makedirs(os.path.dirname(out_path) or ".", exist_ok=True)

    def turn_of(i):
        s = str(SHIFT.get(i, "MANHA")).upper()
        if s in ["5X2", "ANY", "AMBOS", "BOTH", "*"]:
            return "ANY"
        if s in ["NOITE", "NIGHT"]:
            return "NOITE"
        return "MANHA"

    has_sol = (m.Status in [GRB.OPTIMAL, GRB.INTERRUPTED]) and (getattr(m,
        "SolCount", 0) > 0)

    aloc, y_rows, carga, carga_dia, esforco, desvios = [], [], [], [], [],
        []

    for (s_unit, j, k) in SJk:
        yv = float(y[s_unit, j, k].X) if has_sol else None
        s_base = unit2base[s_unit]
        if has_sol and yv and yv > 0.5:
            for i in allowed[(s_unit, j, k)]:
                if x[i, s_unit, j, k].X > 0.5:
                    aloc.append({
                        "base_subarea_id": s_base,
                        "ambiente_idx": unit2idx[s_unit],
                        "area_class": sub2class_base[s_base],
                        "day": j,
                        "task_type_id": k,
                        "professional_id": i,
                        "shift_allocated": turn_of(i),
                        "duration_hours": float(getD(s_base, k)),
                        "effort_weight": float(getE(s_base, k)),
                        "priority_weight": float(getP(s_base, k)),
                        "crew_size_day": win_map[(s_unit, j, k)]["crew"]
                    })
        w = win_map[(s_unit, j, k)]
        y_rows.append({
            "base_subarea_id": s_base,
            "ambiente_idx": unit2idx[s_unit],
```

```
                "area_class": sub2class_base[s_base],
                "day": j, "task_type_id": k,
                "y_value": yv,
                "crew_size": w["crew"],
                "window_start": w["start"],
                "window_end": w["end"],
                "window_req": w["req"]
            })

    if has_sol:
        for i in I:
            pairs = by_i.get(i, [])
            h = sum(x[i, s_unit, j, k].X * getD(unit2base[s_unit], k) for (
                s_unit, j, k) in pairs) if pairs else 0.0
            e = sum(x[i, s_unit, j, k].X * getE(unit2base[s_unit], k) for (
                s_unit, j, k) in pairs) if pairs else 0.0
            carga.append({"professional_id": i, "shift_type": turn_of(i), "
                hours_total": float(h)})
            esforco.append({"professional_id": i, "shift_type": turn_of(i),
                 "effort_total": float(e)})
            desvios.append({"professional_id": i, "delta_hours": float(
                delta[i].X), "epsilon_effort": float(epsilon[i].X)})
            for j in J:
                pairs_ij = by_ij.get((i, j), [])
                h_ij = sum(x[i, s_unit, j, k].X * getD(unit2base[s_unit], k
                    ) for (s_unit, k) in pairs_ij) if pairs_ij else 0.0
                carga_dia.append({"professional_id": i, "day": j, "
                    shift_type": turn_of(i), "hours": float(h_ij)})

    mult_df = pd.DataFrame([{"base_subarea_id": unit2base[s], "ambiente_idx
        ": unit2idx[s]} for s in S_units])

    KPIs = pd.DataFrame([{
        "model_status": m.Status,
        "objective_value": float(m.objVal) if has_sol else None,
        "H_total": float(H_total),
        "E_total": float(E_total),
        "sum_delta": float(sum(delta[i].X for i in I)) if has_sol else None
            ,
        "sum_epsilon": float(sum(epsilon[i].X for i in I)) if has_sol else
            None,
        "num_tasks_required": int(sum(w["req"] for s in S_units for w in
            Wins.get(s, []))),
        "num_decision_days": int(len(SJk)),
        "num_x_vars": int(len(x_keys))
    }])

    with pd.ExcelWriter(out_path, engine="openpyxl") as writer:
        pd.DataFrame(aloc).to_excel(writer, sheet_name="Alocacao", index=
            False)
        pd.DataFrame(y_rows).to_excel(writer, sheet_name="Y_Tarefas", index
            =False)
        pd.DataFrame(carga).to_excel(writer, sheet_name="CargaHoras", index
            =False)
        pd.DataFrame(carga_dia).to_excel(writer, sheet_name="CargaHorasDia"
            , index=False)
        pd.DataFrame(esforco).to_excel(writer, sheet_name="Esforco", index=
            False)
        pd.DataFrame(desvios).to_excel(writer, sheet_name="Desvios", index=
```

```
            False)
        KPIs.to_excel(writer, sheet_name="KPIs", index=False)
        mult_df.to_excel(writer, sheet_name="Ambientes_Expandidos", index=
            False)

    print(f"\ n   Planilha gerada em: {os.path.abspath(out_path)}", flush=
        True)

# Exporta sempre (mesmo inviável)
if m.Status == GRB.INFEASIBLE:
    print("\nModelo inviável     gerando IIS...", flush=True)
    m.computeIIS()
    m.write("modelo_iis.ilp")
    m.write("modelo_completo.ilp")
    export_excel()
else:
    export_excel()
```