



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Agendamento de recursos acadêmicos em Instituições de Ensino Superior: Estudo de caso na Universidade de Brasília

Éber Júnio Borges Moreira

Dissertação apresentada como requisito parcial para conclusão do
Mestrado Profissional em Computação Aplicada

Orientador

Prof. Dr. Sergio Antônio Andrade De Freitas

Brasília
2024

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

Ba Borges Moreira, Éber Júnio
 Agendamento de recursos acadêmicos em Instituições de
 Ensino Superior: Estudo de caso na Universidade de Brasília
 / Éber Júnio Borges Moreira; orientador Sergio Antônio
 Andrade de Freitas. -- Brasília, 2024.
 164 p.

 Dissertação(Mestrado Profissional em Computação Aplicada)
 -- Universidade de Brasília, 2024.

 1. Agendamento de cursos. 2. Otimização de alocação de
 recursos;. 3. Sistemas de gestão acadêmica. 4. Modelo de
 agendamento adaptativo. 5. Estudo de caso. I. Freitas,
 Sergio Antônio Andrade de, orient. II. Título.



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Agendamento de recursos acadêmicos em Instituições de Ensino Superior: Estudo de caso na Universidade de Brasília

Éber Júnio Borges Moreira

Dissertação apresentada como requisito parcial para conclusão do
Mestrado Profissional em Computação Aplicada

Prof. Dr. Sergio Antônio Andrade De Freitas (Orientador)
Universidade de Brasília

Profa. Dra. Irene Pimenta Rodrigues Profa. Dra. Andrea Felipe Cabello
Universidade de Évora Universidade de Brasília

Profa. Dra. Edna Dias Canedo
Coordenador do Programa de Pós-graduação em Computação Aplicada

Brasília, 17 de dezembro de 2024

Dedicatória

Dedico este trabalho aos meus familiares, cuja presença e apoio foram fundamentais ao longo de toda a minha jornada.

Às minhas filhas Rafaela, Gabriela e à minha esposa Solange, que são minha maior fonte de inspiração e amor incondicional.

Aos meus pais, José Moreira e Maria Borges, por terem-me ensinado os valores da determinação e do esforço, essenciais para a realização deste trabalho.

Aos meus irmãos, Cely, Selma, Selmíria e Natanael, pelo carinho, apoio e incentivo que sempre me proporcionaram.

Com todo o meu carinho e gratidão.

Éber Júnio Borges Moreira

Agradecimentos

Agradeço, primeiramente, a Deus, por me dar força e perseverança ao longo de toda esta jornada acadêmica.

Expresso minha gratidão ao Prof. Dr. Sérgio Antônio Andrade de Freitas, pela orientação, paciência e inestimável contribuição para a realização deste trabalho.

À Profa. Dra. Irene Pimenta Rodrigues, à Profa. Dra. Andrea Felipe Cabello e ao Prof. Dr. André Luiz Peron Martins Lanna, pela análise e pelas valiosas sugestões que enriqueceram este trabalho.

Agradeço à Universidade de Brasília (UnB), especialmente ao Departamento de Ciência da Computação e à Secretaria de Tecnologia da Informação, por fornecer os recursos e o ambiente acadêmico que possibilitaram a execução deste projeto.

Aos meus familiares, pelo apoio incondicional, paciência e incentivo em todos os momentos. Sua presença foi essencial para que eu pudesse superar os desafios ao longo desta jornada.

Por fim, agradeço a todos que, de alguma forma, contribuíram para a realização deste trabalho. A colaboração e incentivo de cada um de vocês foram essenciais para a conclusão desta dissertação.

Resumo

O agendamento de recursos acadêmicos em instituições de ensino superior configura-se como um problema de elevada complexidade, envolvendo a alocação de turmas, professores, horários e espaços físicos, de modo a atender a múltiplas restrições e preferências institucionais. Este trabalho apresenta o modelo SARA (Sistema de Agendamento de Recursos Acadêmicos), concebido para oferecer flexibilidade e adaptabilidade às particularidades das instituições de ensino superior.

A pesquisa foi conduzida em duas etapas principais: a realização de uma revisão sistemática da literatura e a proposição do modelo SARA. A revisão sistemática possibilitou o mapeamento de estratégias, algoritmos e restrições amplamente utilizados na literatura, fornecendo subsídios teóricos para a concepção do modelo. A proposta do SARA integra características como personalização, integração com sistemas acadêmicos e a aplicação de algoritmos de otimização, promovendo uma abordagem abrangente e adaptável para a criação de agendamentos em ambientes universitários.

A validação do modelo foi realizada por meio de um estudo de caso na Universidade de Brasília (UnB), em um ambiente de grande porte e alta complexidade. Os resultados preliminares demonstraram a viabilidade e a eficácia do modelo em atender às demandas específicas da instituição, evidenciando seu potencial de aplicação em diferentes contextos institucionais. A pesquisa contribui para o avanço teórico e prático no campo da gestão acadêmica, apresentando uma solução que combina rigor metodológico com aplicabilidade prática.

Palavras-chave: Agendamento de cursos; Otimização de alocação de recursos; Sistemas de gestão acadêmica; Modelo de agendamento adaptativo; Estudo de caso

Abstract

The scheduling of academic resources in higher education institutions is a highly complex problem, involving the allocation of classes, professors, time slots, and physical spaces, while meeting multiple institutional constraints and preferences. This study introduces the SARA model (Academic Resource Scheduling System), designed to provide flexibility and adaptability to the specific needs of higher education institutions.

The research was conducted in two main stages: a systematic review of the literature and the proposal of the SARA model. The systematic review enabled the mapping of strategies, algorithms, and constraints widely utilized in the literature, providing a solid theoretical foundation for the model's conception. The SARA model incorporates features such as customization, integration with academic systems, and the application of optimization algorithms, promoting a comprehensive and adaptable approach to scheduling in university environments.

The model was validated through a case study at the University of Brasília (UnB), in a large-scale and highly complex environment. Preliminary results demonstrated the feasibility and effectiveness of the model in addressing the institution's specific demands, highlighting its potential applicability to other institutional contexts. This research contributes to theoretical and practical advancements in academic management, presenting a solution that combines methodological rigor with practical applicability.

Keywords: Course timetabling; Resource Allocation Optimization; Academic Management Systems; Adaptive Scheduling Framework; Case Study

Sumário

1	Introdução	1
1.1	Contextualização do Problema	1
1.2	Motivação	2
1.3	Metodologia	2
1.4	Objetivo Geral	3
1.5	Objetivos específicos	3
1.6	Estrutura do trabalho	3
2	Revisão da Literatura	4
2.1	Protocolo da revisão sistemática da literatura	4
2.1.1	Questões de pesquisa	5
2.1.2	<i>String</i> de busca	5
2.1.3	Fontes	5
2.1.4	CrITÉrios de seleÇo	6
2.2	ExecuÇo do protocolo	6
2.3	Respostas às questes de pesquisa	14
2.4	Trabalhos relacionados	20
3	ContextualizaÇo	24
3.1	Universidade de BrasÍlia	24
3.2	Prefeitura da UnB	28
3.3	Secretaria de Tecnologia da Informaço	28
3.4	Sistemas Institucionais Integrados de Gesto	29
3.5	Sistema de Ensalamento	30
3.6	Consideraçes finais	31
4	O modelo SARA	33
4.1	Modelo proposto	33
4.2	Partes do modelo	35
4.2.1	PersonalizaÇo	35

4.2.2	Agendamento	40
4.2.3	<i>Feedback</i> e avaliação	43
4.2.4	Comunicação e Notificação	44
4.2.5	Predição	44
4.2.6	Integração	45
4.2.7	Relatórios	47
4.3	Usuários	48
5	A implementação do SARA	49
5.1	Introduzindo o SARA	49
5.2	Características	51
5.2.1	Arquivos de Entrada de dados	51
5.2.2	<i>Constraints</i>	52
5.2.3	Algoritmo adotado	56
5.2.4	Crítica algorítmica	58
5.2.5	Usuários	59
5.2.6	Ações	60
5.3	Implementação	61
5.3.1	Tecnologias Utilizadas	61
6	Resultados	65
6.1	Configuração do ambiente experimental	65
6.2	Conjuntos de dados	66
6.3	Coleta de dados	66
6.4	Análise individual das instâncias	67
6.4.1	DS1: Características e Análise de Desempenho	67
6.4.2	DS2: Características e Análise de Desempenho	67
6.4.3	DS3: Características e Análise de Desempenho	68
6.4.4	DS4: Características e Análise de Desempenho	69
6.4.5	DS5: Características e Análise de Desempenho	70
6.4.6	DS6: Características e Análise de Desempenho	70
6.4.7	DS7: Características e Análise de Desempenho	71
6.4.8	DS8: Características e Análise de Desempenho	72
6.4.9	DS9: Características e Análise de Desempenho	72
6.4.10	DS10: Características e Análise de Desempenho	73
6.4.11	DS11: Características e Análise de Desempenho	74
6.5	Processos de pré-resolução	74
6.6	Análise do comportamento do solucionador	75

7 Conclusão e trabalhos futuros	77
7.1 Trabalhos futuros	78
Apêndice	88
A Constraints Identificadas nos Artigos Pesquisados	89
B SQL para extração de dados de espaços físicos	139
C SQL para extração de dados de eventos	140
D Guia Visual do SARA	142
D.1 Organização do Guia Visual	142
D.1.1 Página Pública	142
D.1.2 Tela de Login e Autenticação	143
D.1.3 Tela de Agendamentos	144
D.1.4 Tela Usuários	150
D.1.5 Página de Ajuda	150
Glossário	151

Lista de Figuras

2.1	Fluxo de seleção de estudos	7
3.1	Os <i>campi</i> da UnB estão distribuídos em várias cidades.	25
3.2	Organograma da STI.	29
4.1	Visão geral do modelo SARA.	34
5.1	Estrutura do SARA	50
6.1	Tempo para encontrar a solução ótima.	76
6.2	Número de soluções encontradas.	76
D.1	Página pública do SARA	143
D.2	Tela de Login do SARA	144
D.3	Tela de Agendamentos do SARA	145
D.4	Seção superior da tela de edição de agendamentos.	146
D.5	Seção inferior da tela de edição de agendamentos: quadro de horários.	147
D.6	Representação de evento no SARA.	148
D.7	Tela de Usuários	150

Lista de Tabelas

2.1	Resultado da busca atemporal nas bases de pesquisa.	8
2.2	Artigos mais citados	8
2.2	Artigos mais citados	9
2.2	Artigos mais citados	9
2.3	Número de estudos retornados na pesquisa no período de 2018 a 2023.	10
2.4	Estudos recentes escolhidos para a revisão da literatura.	10
2.4	Estudos recentes escolhidos para a revisão da literatura.	11
2.4	Estudos recentes escolhidos para a revisão da literatura.	12
2.4	Estudos recentes escolhidos para a revisão da literatura.	13
2.4	Estudos recentes escolhidos para a revisão da literatura.	14
2.5	Mapeamento das abordagens	18
2.6	<i>Hard constraints</i> mais utilizadas	20
2.7	<i>Soft constraints</i> mais utilizadas.	21
3.1	Panorama da UnB em 2022.	27
4.1	<i>Hard constraints</i> discutidas em McCollum et al. (2010)	41
4.2	<i>Soft constraints</i> discutidas em McCollum et al. (2010)	42
5.1	Descrição dos campos do arquivo de espaços físicos	52
5.2	Descrição dos Campos dos Eventos Acadêmicos	52
5.3	Descrição dos Parâmetros do Solver CP-SAT utilizados no SARA.	57
5.4	Estados do evento	59
6.1	Sumário dos conjuntos de dados	66
6.2	Sumário dos detalhes do modelo	75
A.1	Constraints identificadas nos artigos pesquisados	90
D.1	Legenda de cores dos eventos de agendamento	149

Lista de Abreviaturas e Siglas

CIC Departamento de Ciência da Computação.

CP-SAT Constraint Programming Solver with SAT.

IFES Instituições Federais de Ensino Superior.

SARA Sistema de Agendamento de Recursos Acadêmicos.

SIGAA Sistema Integrado de Gestão de Atividades Acadêmicas.

SITAB Sistema Unificado de Tabelas da UnB.

STI Secretaria de Tecnologia da Informação.

UnB Universidade de Brasília.

Capítulo 1

Introdução

1.1 Contextualização do Problema

As instituições de ensino superior enfrentam desafios logísticos na gestão de seus recursos acadêmicos (SIDDIQUI; RAZA; TARIQ, 2018), (HOUHAMDÍ et al., 2019), (VIANNA et al., 2020). Gerenciar estes recursos é uma tarefa complexa que envolve a alocação de salas de aula, professores e turmas, de forma a atender a uma série de requisitos e preferências. Esses requisitos e preferências são comumente referidos como *constraints* ou restrições, e são elementos necessários para a formulação e resolução do aqui denominado problema do agendamento universitário. De acordo com Lewis (2007), as *constraints* podem ser classificadas em duas categorias principais: *hard constraints* e *soft constraints*. As *hard constraints* são restrições rígidas que devem ser rigorosamente cumpridas para que uma solução seja considerada viável (BURKE et al., 1997). Elas representam condições não-negociáveis, como a disponibilidade de uma sala ou a disponibilidade de um professor em um determinado horário. Violar uma *hard constraint* resulta em uma solução inviável. Por outro lado, segundo Burke et al. (1997) as *soft constraints* são restrições suaves que, embora desejáveis, não são estritamente necessárias para a viabilidade da solução. Estas podem incluir, por exemplo, a preferência de um professor por um horário específico ou a proximidade de salas para turmas consecutivas. Violar uma *soft constraint* não torna a solução inviável, mas pode afetar sua qualidade ou aceitabilidade.

Observa-se que apesar da Universidade de Brasília (UnB) possuir mecanismos de gestão acadêmica que auxiliam na administração desses recursos, a estratégia adotada carece de um conjunto de regras e condições específicas para uma gestão mais eficiente dos seus recursos (UNB, 2023b). Além disso, esse cenário é tratado de forma rígida na UnB, pois a abordagem é empregada de forma centralizada não permitindo ser adaptada facilmente para melhor atendimento das particularidades de cada setor. Portanto, verifica-se que há

espaço para a busca por estratégias que sejam flexíveis e adaptáveis, contribuindo para um melhor tratamento deste problema na instituição.

1.2 Motivação

Considerando o panorama delineado, observa-se a necessidade de otimizar a gestão de recursos acadêmicos em instituições de ensino superior. Neste contexto, o foco da presente pesquisa reside na execução de uma revisão sistemática da literatura, que engloba a identificação de algoritmos e de restrições, comumente empregados como possibilidades na resolução do problema do agendamento universitário. A partir da pesquisa, o intuito é a formulação de um modelo que seja flexível para atender às especificidades e demandas de uma instituição de ensino público federal de grande porte, como é o caso da UnB.

Adicionalmente, o fato de haver a possibilidade de realizar testes na própria UnB, amplia o escopo motivacional deste trabalho. A disponibilidade dos recursos desta instituição viabiliza a realização de testes e validações em diferentes cenários. Assim, será possível fazer a validação do modelo em um ambiente de grande porte, como o das Instituições Federais de Ensino Superior(IFES).

1.3 Metodologia

A metodologia de pesquisa prevista para este trabalho consiste em uma revisão sistemática da literatura combinada com a proposição de um modelo e a realização de um caso de uso. A primeira etapa, abordada nesta proposta, envolve uma revisão sistemática da literatura conforme as diretrizes estabelecidas pela metodologia de Kitchenham e Charters (2007). Nesta etapa do trabalho, é conduzida uma busca por estudos nas bases de dados acadêmicas *ACM Digital Library*, *IEEE Digital Library* e *Scopus*, com o objetivo de identificar as estratégias e restrições frequentemente utilizadas frente o problema do agendamento universitário, que servem tanto para entendimento do problema quanto como fundamentação para o modelo proposto.

A segunda etapa desta pesquisa, programada para ocorrer após a qualificação, tem como foco a execução de testes e avaliações do modelo proposto, valendo-se de um caso de uso para isto. Neste contexto, a UnB, aproveitando a oportunidade oferecida por esta instituição, será o ambiente para a continuidade dos estudos. Poder conduzir este estudo na UnB é relevante, considerando a possibilidade de entendimento da complexidade e os desafios inerentes ao agendamento de cursos neste ambiente de grande porte.

1.4 Objetivo Geral

O objetivo geral desta pesquisa é propor, analisar e testar uma estratégia que permita a otimização da alocação de recursos acadêmicos, tais como espaços físicos, horários de professores e agendamento de turmas.

1.5 Objetivos específicos

1. Realizar uma revisão sistemática da literatura para compreender:
 - Quais são os algoritmos mais utilizados na alocação de recursos acadêmicos;
 - Quais algoritmos são adequados para o problema;
 - Quais modelagens do problema se alinham com as características da UnB; e
 - Mapear as *constraints*(*hard* e *soft*).
2. Propor um modelo que seja flexível e adaptável;
3. Criar um protótipo a partir do proposta; e
4. Testar e avaliar a proposta junto a comunidade da UnB.

1.6 Estrutura do trabalho

Para atingir os objetivos propostos, esta pesquisa será conduzida em duas fases distintas. A primeira fase, tratada neste documento, foca nos objetivos específicos 1 e 2, que estão relacionados à qualificação. Os demais objetivos serão abordados em fases subsequentes à qualificação. A estrutura desta dissertação é organizada da seguinte maneira:

Capítulo 1 - Introdução: Este capítulo contextualiza o problema de agendamento universitário, delinea a motivação para a pesquisa e especifica os objetivos deste trabalho.

Capítulo 2 - Revisão da literatura: Este capítulo detalha a metodologia de pesquisa adotada, incluindo aspectos como a definição e execução do protocolo; respostas às questões de pesquisa e trabalhos relacionados.

Capítulo 3 - Contexto institucional da UnB: Este capítulo fornece uma visão sobre a UnB, destacando órgãos e sistemas acadêmicos relevantes para o problema em estudo.

Capítulo 4 - Proposta: Este capítulo apresenta o modelo proposto, descreve as partes que o compõem e os perfis de usuários implicados nesta abordagem.

Capítulo 5 - Considerações finais e cronograma: Este capítulo descreve o caso de uso a ser realizado como continuação desta pesquisa e estabelece o cronograma para a implementação do trabalho da dissertação.

Capítulo 2

Revisão da Literatura

Neste trabalho, foi conduzida uma revisão sistemática da literatura, abrangendo tanto os principais trabalhos acadêmicos já publicados quanto as pesquisas mais recentes sobre o tema. A abordagem adotada, seguindo os moldes de Kitchenham e Charters (2007), auxiliou na identificação de algoritmos e restrições frequentemente empregados na resolução do problema. Assim, este capítulo apresenta o protocolo da revisão sistemática da literatura, a execução do protocolo, as respostas às questões de pesquisa e os trabalhos relacionados.

2.1 Protocolo da revisão sistemática da literatura

A criação de agendamento universitário é uma tarefa que envolve a coordenação de uma série de variáveis, incluindo a alocação de espaços físicos, a definição de horários de aulas e a gestão de recursos humanos (CHEN et al., 2021). Cada universidade enfrenta desafios únicos na elaboração de seus horários devido a diferentes requisitos, que podem ser influenciados por políticas institucionais e pelo sistema educacional de sua região. Dada a sua complexidade, vê-se a necessidade de se adotar uma abordagem bem fundamentada para conduzir pesquisas sobre esse problema, com o objetivo de identificar na literatura as estratégias utilizadas para resolvê-lo.

Neste sentido, para garantir uma revisão abrangente e objetiva da literatura disponível sobre o tema, adotou-se a metodologia de revisão sistemática proposta por Kitchenham e Charters (2007). Esta metodologia visa identificar, avaliar e interpretar todas as pesquisas relevantes disponíveis para uma pergunta de pesquisa específica, tópico de área ou fenômeno de interesse. Ela orienta os pesquisadores a definirem os critérios de inclusão e exclusão para a seleção de estudos, bem como a elaborarem um protocolo de revisão que detalha a estratégia de busca a ser adotada.

2.1.1 Questões de pesquisa

Segundo Kitchenham e Charters (2007), a formulação de questões de pesquisa é um dos elementos do planejamento, pois estabelece as motivações e orienta todo o processo da revisão. Neste estudo, foram estabelecidas quatro questões de pesquisa:

- QP1 - Quais são os algoritmos mais utilizados na alocação de recursos acadêmicos?
- QP2 - Quais são os algoritmos mais adequados para a instituição?
- QP3 - Quais são as modelagens do problema?
- QP4 - Quais são as *constraints* mais utilizadas?

2.1.2 *String* de busca

Para identificação e busca dos trabalhos, criou-se a seguinte *string* de busca com os principais termos encontrados a partir da leitura rápida de alguns artigos sobre o tema:

```
"Course scheduling"OR  
"Course timetabling"OR  
"Course timetabling problem"OR  
"Curriculum-based course timetabling"OR  
"Educational timetabling"OR  
"timetabling approaches"OR  
"timetabling optimization"OR  
"University class timetabling"OR  
"University Course Timetabling"OR  
"University Course Timetabling Issue"OR  
"University course timetabling-problems"OR  
"University Course Timetabling"OR  
"University Timetable Problem"OR  
"University timetabling"
```

2.1.3 Fontes

Para buscar estudos sobre o tema, optou-se pela utilização de bases de dados acadêmicas, especificamente a *ACM Digital Library*¹, a *IEEE Digital Library*² e o *Scopus*³. A escolha dessas bases foi motivada por sua abrangência e credibilidade no campo da ciência da computação e áreas afins.

¹<http://portal.acm.org>

²<http://ieeexplore.ieee.org>

³<http://www.scopus.com>

2.1.4 Critérios de seleção

Os critérios de seleção para a inclusão e exclusão de estudos foram delineados com o objetivo de garantir a qualidade e a relevância dos trabalhos selecionados. Foram incluídos artigos publicados em revistas científicas de alto impacto e trabalhos apresentados em conferências de prestígio na área, que contribuam para o campo do agendamento universitário.

Critérios de inclusão

Para a inclusão de estudos na pesquisa, foram estabelecidos os seguintes critérios:

1. **Contém requisitos:** artigos que apresentem requisitos relacionados ao tema, identificando necessidades ou condições essenciais para o desenvolvimento de soluções na área.
2. **Explica a abordagem:** estudos que detalham metodologias ou *frameworks* adotados para resolver o problema, proporcionando uma visão sobre o processo e a lógica aplicada.
3. **Explica o uso do algoritmo:** trabalhos que descrevem a aplicação de algoritmos, com foco em como foram utilizados ou adaptados para o problema.

Critérios de exclusão

Para a exclusão de estudos da pesquisa, foram definidos os seguintes critérios:

4. **Estudo não recuperado integralmente:** artigos cuja versão completa não pôde ser obtida.
5. **Não trata o tema:** estudos que não abordam o tema específico da pesquisa.
6. **Não trata o tema de forma relevante:** trabalhos que, embora mencionem o tema, não o abordam de maneira significativa para contribuir com a revisão.
7. **Similar a artigos mais relevantes:** estudos com conteúdo similar ou redundante em relação a outros artigos já considerados mais relevantes para a revisão.

2.2 Execução do protocolo

O processo de seleção de estudos para esta pesquisa foi estruturado para incluir artigos relevantes e de alta qualidade que contribuam para o tema. O fluxo de seleção de estudos, ilustrado na Figura 2.1, compreende as etapas:

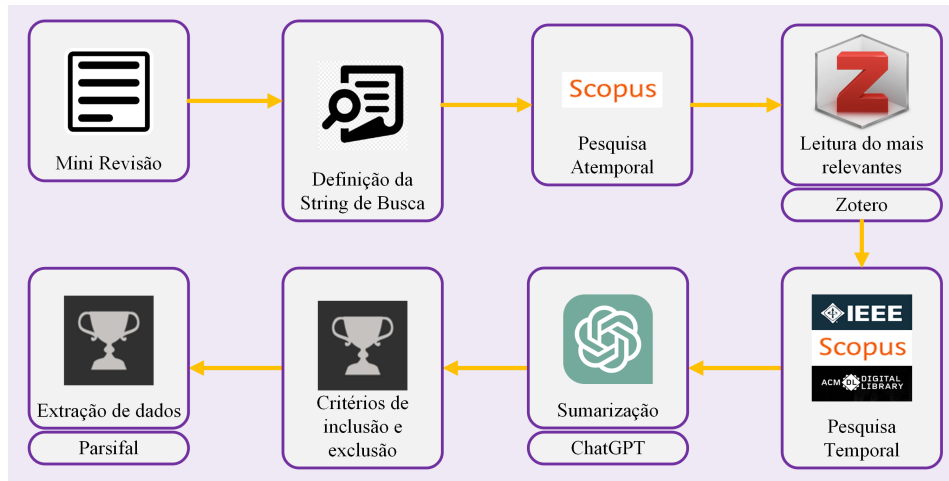


Figura 2.1: Fluxo de seleção de estudos

- **Mini Revisão:** Início do processo com uma breve revisão da literatura para identificar os termos de busca mais comuns relacionados ao tema.
- **Definição da String de Busca:** Com base na mini revisão, os termos identificados são organizados em uma *string* de busca específica para a pesquisa.
- **Pesquisa Atemporal na Scopus:** Realiza-se uma busca sem restrição temporal na base Scopus, com o objetivo de identificar os artigos mais citados.
- **Leitura dos Mais Relevantes (Zotero)⁴:** Os artigos mais citados são organizados e lidos com auxílio do software Zotero, onde são catalogados para facilitar o gerenciamento de referências.
- **Pesquisa Temporal nas Bases ACM, IEEE e Scopus:** Em seguida, realiza-se uma pesquisa temporal (de 2018 a 2023) nas bases ACM Digital Library, IEEE Digital Library e Scopus, para identificar estudos mais recentes.
- **Sumarização (ChatGPT)⁵:** A sumarização dos artigos selecionados é realizada com auxílio do ChatGPT para facilitar a síntese das principais contribuições de cada trabalho.
- **Critérios de Inclusão e Exclusão:** Aplicam-se os critérios definidos para incluir ou excluir estudos, garantindo a relevância e a qualidade dos trabalhos.
- **Extração de Dados (Parsifal):** Por fim, os dados dos artigos selecionados são extraídos e organizados com o auxílio da ferramenta Parsifal para facilitar a análise e o uso na pesquisa.

⁴<https://www.zotero.org/>

⁵<https://chat.com/>

Com a apresentação do fluxo de seleção de estudos, passa-se agora para a execução prática desse processo, onde são apresentados os resultados obtidos. Utilizando a *string* de busca definida, foi realizada uma pesquisa sem restrição temporal nas bases acadêmicas selecionadas, com o objetivo de avaliar a extensão do universo de estudos sobre o tema, conforme os resultados apresentados na Tabela 2.1.

Base de pesquisa	Estudos retornados	Período
<i>ACM Digital Library</i>	912	1968 - 2023
<i>IEEE Digital Library</i>	247	1990 - 2023
<i>Scopus</i>	1167	1977 - 2024
Total	2326	

Tabela 2.1: Resultado da busca atemporal nas bases de pesquisa.

Recorreu-se a duas estratégias para refinar a pesquisa. Primeiro, realizou-se uma busca atemporal na base *Scopus* com o objetivo de identificar os 20 (vinte) artigos mais citados, listados na Tabela 2.2. A coluna ID é utilizada para facilitar referências futuras em outras tabelas, como no mapeamento de *constraints*, por exemplo.

Tabela 2.2: Artigos mais citados

ID	Referência	Título
53	(BURKE et al., 1997)	Automated University timetabling The state of the art
54	(CARTER; LA-PORTE, 1998)	Recent Developments in Practical Course Timetabling
55	(SOCHA; KNOWLES; SAMPELS, 2002)	A MAX -MIN Ant System for the University Course Timetabling Problem
56	(BURKE; PETROVIC, 2002)	Recent research directions in automated timetabling
57	(ROSSI-DORIA et al., 2003)	A Comparison of the Performance of Different Metaheuristics on the Timetabling Problem
58	(SOCHA; SAMPELS; MANFRIN, 2003)	Ant Algorithms for the University Course Timetabling Problem with Regard to the State-of-the-Art
59	(DASKALAKI; BIRBAS; HOSOS, 2004)	An integer programming formulation for a case study in university timetabling

Tabela 2.2: Artigos mais citados

ID	Referência	Título
60	(DASKALAKI; BIRBAS, 2005)	Efficient solutions for a university timetabling problem through integer programming
61	(CHIARANDINI et al., 2006)	An effective hybrid algorithm for university course timetabling
62	(BURKE; PETROVIC; QU, 2006)	Case-based heuristic selection for timetabling problems
63	(BURKE et al., 2007)	A graph-based hyper-heuristic for educational timetabling problems
64	(ABDULLAH; BURKE; McCollum, 2007)	A hybrid evolutionary approach to the university course timetabling problem
65	(LEWIS, 2007)	A survey of metaheuristic-based techniques for University Timetabling problems
66	(QU; BURKE, 2009)	Hybridizations within a graph-based hyper heuristic framework for university timetabling problems
67	(Lü; HAO, 2010)	Adaptive Tabu Search for course timetabling
68	(McCollum et al., 2010)	Setting the Research Agenda in Automated Timetabling: The Second International Timetabling Competition
69	(AL-BETAR; KHADER, 2012)	A harmony search algorithm for university course timetabling
70	(AL-BETAR; KHADER; ZAMAN, 2012)	University Course Timetabling Using a Hybrid Harmony Search Metaheuristic Algorithm
71	(BABAEI; KARIMPOUR; HADIDI, 2015)	A survey of approaches for university course timetabling problem
72	(ABRAMSON, 1991)	Constructing School Timetables Using Simulated Annealing Sequential and Parallel Algorithms

Tabela 2.2: Artigos mais citados

Em seguida, realizou-se uma pesquisa temporal nas bases *ACM Digital Library*, *IEEE Digital Library* e *Scopus*, abrangendo o período de 2018 a 2023, a fim de incluir estudos mais recentes sobre o tema. O número de estudos identificados nesta etapa pode ser visto

na Tabela 2.3.

Base de pesquisa	Estudos retornados
ACM Digital Library	153
IEEE Digital Library	49
Scopus	320
Total	522

Tabela 2.3: Número de estudos retornados na pesquisa no período de 2018 a 2023.

Neste ponto, recorreu-se ao uso de ferramentas como *ChatGPT* para a sumarização e extração de informações acerca das abordagens e restrições empregadas. Além disso, foi feita uma tentativa de se conduzir uma análise de similaridade realizando o cruzamento dos estudos mais recentes com os artigos mais relevantes, visando a eliminação de trabalhos muito parecidos, porém sem sucesso devido a falta de padrão nas respostas retornadas.

Ao final da aplicação dos critérios de seleção, foram selecionados 52 estudos recentes, conforme detalhado na Tabela 2.4. Esses, somados aos 20 artigos mais relevantes, totalizam 72 estudos que formam a base para a realização desta pesquisa.

Tabela 2.4: Estudos recentes escolhidos para a revisão da literatura.

ID	Referência	Título
1	(SONG et al., 2018)	An Iterated Local Search Algorithm for the University Course Timetabling Problem
2	(ZARANDI et al., 2020)	A State of the Art Review of Intelligent Scheduling
3	(AKKAN; GüLCü, 2018)	A bi-criteria hybrid Genetic Algorithm with robustness objective for the course timetabling problem
4	(CHEN et al., 2021)	A Survey of University Course Timetabling Problem: Perspectives, Trends and Opportunities
5	(ASSI; HALAWI; HARATY, 2018)	Genetic Algorithm Analysis using the Graph Coloring Method for Solving the University Timetable Problem
6	(MUKLASON; IRIANTI; MAROM, 2019)	Automated Course Timetabling Optimization Using Tabu-Variable Neighborhood Search Based Hyper-Heuristic Algorithm
7	(GOH; KENDALL; SABAR, 2019b)	Simulated annealing with improved reheating and learning for the post enrolment course timetabling problem
8	(PILLAY; ÖZCAN, 2019)	Automated generation of constructive ordering heuristics for educational timetabling

Tabela 2.4: Estudos recentes escolhidos para a revisão da literatura.

ID	Referência	Título
9	(GOH et al., 2020)	An effective hybrid local search approach for the post enrolment course timetabling problem
10	(THEPPHAKORN; PONGCHAROEN, 2020)	Performance improvement strategies on Cuckoo Search algorithms for solving the university course timetabling problem
11	(GüLCü; AKKAN, 2020)	Robust university course timetabling problem subject to single and multiple disruptions
12	(BANBARA et al., 2019)	teaspoon : solving the curriculum-based course timetabling problems with answer set programming
13	(CHEN et al., 2022)	Investigation of Heuristic Orderings with a Perturbation for Finding Feasibility in Solving Real- World University Course Timetabling Problem
14	(SIDDIQUI; RAZA; TARIQ, 2018)	A web-based group decision support system for academic term preparation
15	(JAMILI; HAMID; GHAROUN, 2018)	Developing a comprehensive and multi-objective mathematical model for university course timetabling problem: A real case study
16	(SIEW et al., 2022)	Grouping and Heuristics for A Multi-stage Class Timetabling System
17	(GOZALI et al., 2020)	Solving university course timetabling problem using localized island model genetic algorithm with dual dynamic migration policy
18	(RAPPOS et al., 2022)	A Mixed-Integer Programming Approach for Solving University Course Timetabling Problems
19	(GOH; KENDALL; SABAR, 2019a)	Monte carlo tree search in finding feasible solutions for course timetabling problem
20	(TAVAKOLI et al., 2020)	Proposing a novel heuristic algorithm for university course timetabling problem with the quality of courses rendered approach; a case study
21	(JUNN et al., 2019)	A formal model of multi-agent system for university course timetabling problems

Tabela 2.4: Estudos recentes escolhidos para a revisão da literatura.

ID	Referência	Título
22	(BAGGER; Sørensen; STIDSEN, 2019)	Dantzig–Wolfe decomposition of the daily course pattern formulation for curriculum-based course timetabling
23	(MAURITSIUS et al., 2017)	Novel Local Searches for Finding Feasible Solutions in Educational Timetabling Problem
24	(MUKLASON et al., 2022)	Flexible Automated Course Timetabling System with Lecturer Preferences Using Hyper-Heuristic Algorithm
25	(CHEN; WU, 2021)	A Meta-Heuristic Approach for The Constraint Satisfaction Problem
26	(HOUHAMDI et al., 2019)	A multi-agent system for course timetable generation
27	(WAHID et al., 2019)	Constructing population of initial university timetable: Design and analysis
28	(VIANNA et al., 2020)	Hybrid VNS-TS heuristics for University Course Timetabling Problem
29	(DIMITSAS et al., 2022)	An Exact Based Approach for the Post Enrollment Course Timetabling Problem
30	(LEMO; MONTEIRO; LYNCE, 2020)	Minimal Perturbation in University Timetabling with Maximum Satisfiability
31	(BABAIE; KARIMPOUR; HADIDI, 2019)	Generating an optimal timetabling for multi-departments common lecturers using hybrid fuzzy and clustering algorithms
32	(THEPPHAKORN; PONGCHAROEN, 2023)	Modified and Hybridised Bi-Objective Firefly Algorithms for University Course Scheduling
33	(DUNKE; NICKEL, 2023)	A matheuristic for customized multi-level multi-criteria university timetabling
34	(REN; LI, 2022)	Computer Intelligent Course Scheduling System Based on Deep Learning
35	(JUNE et al., 2020)	Sequential Constructive Algorithm incorporate with Fuzzy Logic for Solving Real World Course Timetabling Problem

Tabela 2.4: Estudos recentes escolhidos para a revisão da literatura.

ID	Referência	Título
36	(THEPPHAKORN; SOONCHAROEN; PONGCHAROEN, 2020)	Academic Operating Costs Optimisation Using Hybrid MCPSO Based Course Timetabling Tool
37	(ABDALLA et al., 2019)	Performance comparison of sequential and cooperative integer programming search methodologies in solving curriculum-based university course timetabling problems (CB-UCT)
38	(NGUYEN; NGUYEN, 2021)	An SHO-based approach to timetable scheduling: a case study
39	(ALEXA; ZMARANDA; MOISI, 2021)	Modeling and Solving the University Timetabling Problem using the Constraints Satisfaction Model: A Case Study
40	(CARAMIA; GIORDANI, 2021)	Curriculum-Based Course Timetabling with Student Flow, Soft Constraints, and Smoothing Objectives: an Application to a Real Case Study
41	(ZHANG; WANG, 2021)	Intelligent Course Scheduling System Based on Case-based Reasoning
42	(WIJAYA, 2020)	Two stages best first search algorithm using hard and soft constraints heuristic for course timetabling
43	(HAFSA et al., 2023)	Solving a multiobjective professional timetabling problem using evolutionary algorithms at Mandarin Academy
44	(Mei Ching Chen et al., 2021)	A Hybrid of Heuristic Orderings and Variable Neighbourhood Descent for a Real Life University Course Timetabling Problem
45	(AINI; SAPTAWIJAYA; AMINAH, 2017)	Bringing answer set programming to the next level: A real case on modeling course timetabling
46	(ZHENG et al., 2022)	Course Scheduling Algorithm Based on Improved Binary Cuckoo Search
47	(EKE et al., 2019)	A Case Study of Model-Driven Engineering for Automated Timetabling

Tabela 2.4: Estudos recentes escolhidos para a revisão da literatura.

ID	Referência	Título
48	(SHUAI, 2021)	Design of Automatic Course Arrangement System for Electronic Engineering Teaching Based on Monte Carlo Genetic Algorithm
49	(BASHAB et al., 2023)	Optimization Techniques in University Timetabling Problem: Constraints, Methodologies, Benchmarks, and Open Issues
50	(NUGROHO; HERMAWAN, 2018)	Solving University Course Timetabling Problem Using Memetic Algorithms and Rule-based Approaches
51	(THEPPHAKORN; SOONCHAROEN; PONGCHAROEN, 2021)	Particle Swarm Optimisation Variants and Its Hybridisation Ratios for Generating Cost-Effective Educational Course Timetables
52	(BABAEI; KARIMPOUR; HADIDI, 2018)	Applying Hybrid Fuzzy Multi-Criteria Decision-Making Approach to Find the Best Ranking for the Soft Constraint Weights of Lecturers in UCTP

2.3 Respostas às questões de pesquisa

QP1 - Quais são os algoritmos mais utilizados na alocação de recursos acadêmicos?

As heurísticas construtivas como, por exemplo *Largest degree first*, *Largest degree with tournament selection*, *Colour degree* e *Saturation degree*, vem sendo empregadas como estratégia para a geração do agendamento. No estudo realizado por Burke, Petrovic e Qu (2006), desenvolveu-se um sistema que utiliza o raciocínio baseado em casos (CBR) para adaptar heurísticas pré-existentes a novos problemas, empregando características e pesos específicos no cálculo da similaridade entre os casos. No trabalho de Burke et al. (2007), investiga-se uma hiper-heurística simples e genérica que permuta diversas heurísticas para a criação de agendamentos aplicáveis tanto a exames quanto a aulas. Qu e Burke (2009) apresentaram um *framework* híbrido que combina métodos de busca local com uma hiper-heurística baseada em grafos, criando sequências de heurísticas para obter a melhor solução. Mais recentemente, Pillay e Özcan (2019) abordaram a geração automatizada de heurísticas de ordenação construtiva para a programação educacional.

Outra estratégia que é comumente utilizada é o *Simulated Annealing*. O estudo de Abramson (1991) investiga o uso desta técnica, oferecendo uma alternativa às abordagens mais convencionais, que frequentemente utilizavam programação linear ou métodos heurísticos àquela época. A técnica era relativamente nova no campo da otimização e se revelou eficaz, porém apontando para a necessidade de estudos futuros. Chiarandini et al. (2006) combinou técnicas como a busca tabu e heurísticas construtivas com o *Simulated Annealing* na construção de um algoritmo híbrido que se mostrou eficiente em comparação com *benchmarks*. Song et al. (2018) conseguiu resultados competitivos ao propor um algoritmo de busca local iterado para encontrar uma solução viável para o problema do agendamento universitário. Uma vez que um cronograma inicial é construído, o *Simulated Annealing* combinado com um procedimento de diversificação que traz perturbação moderada ou até mesmo melhoria para a solução atual são realizados de maneira iterativa até que uma condição de parada seja atendida. No trabalho de Goh, Kendall e Sabar (2019b), foi introduzido o algoritmo *Simulated Annealing with Improved Reheating and Learning* (SAIRL), que visa encontrar soluções viáveis em uma primeira fase e aprimorá-las em uma segunda fase. Diferentemente do *Simulated Annealing* convencional, o SAIRL ajusta seus parâmetros automaticamente e incorpora aprendizado por reforço, oferecendo competitividade e escalabilidade em testes com conjuntos de dados *benchmark*.

No campo da programação inteira (IP), Daskalaki, Birbas e Housos (2004) propôs um modelo que permite a incorporação de regras e restrições específicas ao agendamento de cursos, professores e salas de aula. Quanto aos resultados, o modelo demonstrou capacidade de gerar horários sem problemas significativos relacionados ao tempo de computação. Neste outro estudo, Neste outro estudo, Daskalaki e Birbas (2005) aplica a programação inteira com relaxamento para otimizar a eficiência computacional e a qualidade dos horários. A metodologia é dividida em duas fases distintas. Na fase inicial, seu modelo relaxa as restrições de consecutividade para sessões que ocorrem em múltiplos períodos, facilitando, assim, a resolução do problema. Na fase subsequente, essas restrições são progressivamente reintroduzidas, assegurando a necessária consecutividade nas sessões de múltiplos períodos. A estratégia de relaxamento adotada no modelo demonstrou eficiência computacional, possibilitando até a sua utilização de forma interativa. O modelo provou ser suficientemente flexível para incorporar requisitos adicionais, como pausas para almoço e distribuição equilibrada de horários, sem comprometer a qualidade dos horários produzidos, além de contemplar as preferências dos docentes e a possibilidade de ajustes interativos. O estudo de Rappos et al. (2022), emprega a programação inteira mista para criar agendamentos. Para isto, utiliza uma estrutura de grafos que representam os conflitos entre as aulas. O problema é resolvido usando um algoritmo iterativo que começa com um conjunto pequeno de restrições e gradualmente adiciona mais até que uma solução

viável seja encontrada. Uma vez que uma solução viável é encontrada, uma heurística de melhoria é aplicada para aumentar a qualidade da solução.

A pesquisa neste domínio também tem explorado combinações de diferentes heurísticas e meta-heurísticas. Abordagens híbridas que integram, por exemplo, busca tabu, *Simulated Annealing* e algoritmos genéticos têm mostrado sucesso tanto em alcançar soluções viáveis quanto em minimizar violações de restrições, porém observa-se importância da escolha e configuração adequadas dos parâmetros algorítmicos (CHIARANDINI et al., 2006), (ABDULLAH; BURKE; McCollum, 2007). No trabalho de Goh et al. (2020), é feita a proposta de uma abordagem híbrida de busca local, combinando o *Two-Stage Swap Procedure* (TSSP) com o *Iterated Local Search* (ILS), para formar o TSSP-ILS. Este algoritmo é utilizado tanto para construir soluções viáveis quanto para aprimorá-las. Além disso, o *Simulated Annealing with Reheating* (SAR) foi introduzido para oferecer melhorias adicionais nas soluções. No estudo de Thepphakorn, Sooncharoen e Pongcharoen (2021), foi introduzida uma ferramenta de agendamento que usa Otimização por Enxame de Partículas Híbridas (HPSOT), combinando diferentes variantes da Otimização por Enxame de Partículas (PSO). Esta abordagem híbrida mostrou-se eficaz tanto em otimizar custos operacionais acadêmicos como em acelerar o tempo de computação. Neste outro estudo, Akkan e Gülcü (2018) apresentam uma abordagem que adiciona uma métrica de robustez, além do critério tradicional de penalidade por violação de restrições. Para alcançar esses objetivos dualistas, eles utilizaram um Algoritmo Genético Multiobjetivo Híbrido (MOGA), que integra técnicas como *Hill Climbing* e *Simulated Annealing*. Os *frameworks* híbridos como hiper-heurísticas de grafos (GHH) têm permitido a exploração eficaz do espaço de soluções através da aplicação sequencial de diferentes métodos de pesquisa (QU; BURKE, 2009). O desenvolvimento de algoritmos adaptativos, como a Busca Tabu Adaptativa (ATS), representa outra vertente, oferecendo mecanismos dinâmicos para a minimização de restrições e para a adaptação da busca no espaço de soluções (Lü; HAO, 2010).

Além das estratégias descritas anteriormente, outras técnicas vêm sendo utilizadas para abordagem deste problema:

- Lógica *Fuzzy*

O trabalho de Zarandi et al. (2020), destaca a lógica *fuzzy* como uma técnica proeminente na inteligência computacional para lidar com incertezas, podendo ser utilizadas para a geração de agendamentos. Babaei, Karimpour e Hadidi (2018) implementaram uma abordagem de tomada de decisão multicritério fundamentada em técnicas *fuzzy* para otimizar as preferências dos docentes na criação de horários, obtendo resultados que indicam uma maior satisfação entre os envolvidos. Paralelamente, Babaei, Karimpour e Hadidi (2019) concentram-se na otimização dos

horários de professores que atuam em múltiplos departamentos, utilizando um algoritmo híbrido que integra métodos *fuzzy* e técnicas de *clustering*. Esta abordagem foi validada com dados reais e demonstrou eficácia na minimização do uso excessivo de recursos pelos departamentos. Em um estudo adicional, June et al. (2020) propõem uma integração do Algoritmo Construtivo Sequencial (SCA) com a lógica *fuzzy*, permitindo a combinação de diversas heurísticas em uma única, o que resulta em uma melhoria considerável na qualidade dos horários.

- Sistemas Especialistas

Conforme ilustrado por Zarandi, os sistemas especialistas (SE) são uma ramificação importante da inteligência artificial e têm mostrado notável eficácia em problemas de agendamento, particularmente nas indústrias de manufatura. Estes sistemas operam com raciocínio inferencial complexo, utilizando uma base de conhecimento, um motor de inferência e uma interface de usuário (METAXIOTIS et al., 2004; LIAO, 2005). No âmbito do agendamento universitário, Nugroho e Hermawan (2018) adotam uma abordagem híbrida, mesclando algoritmos meméticos e regras específicas do domínio. Essa abordagem inicia com uma solução preliminar gerada por um algoritmo genético, seguida da aplicação de técnicas de busca local e da introdução de regras para refinar e corrigir inconsistências na solução. A validação experimental da estratégia sugere que ela é capaz de obter agendamentos de alta qualidade.

- Aprendizado de Máquina

A utilização de métodos de aprendizado de máquina para aprimorar sistemas de agendamento universitário é uma área de pesquisa emergente. Estes métodos podem ser categorizados em duas principais vertentes: Redes Neurais e Raciocínio Baseado em Casos (CBR), cada um com suas próprias vantagens e desafios, como discutido em Zarandi et al. (2020).

- Redes Neurais As redes neurais podem ser empregadas para auxiliar automatização e a otimização no processo de criação de agendamento. O estudo de Dunke e Nickel (2023) propõe um planejamento multi-nível usando um algoritmo genético e uma meta-modelagem de rede neural para personalizar os horários e melhor avaliar os planos gerados. Em contrapartida, o estudo de Ren e Li (2022) faz uso de tecnologias de *deep learning* combinadas com algoritmos genéticos para considerar múltiplas restrições complexas, como disponibilidade de professores e salas, obtendo resultados otimizados com taxas de utilização mais elevadas.

- Raciocínio Baseado em Casos (CBR) O CBR é uma técnica que visa resolver novos problemas de agendamento com base em soluções para problemas similares do passado. No estudo de Burke, Petrovic e Qu (2006), um sistema com CBR foi desenvolvido para adaptar heurísticas antigas a novos problemas, usando características e pesos específicos para calcular a similaridade entre casos. Complementarmente, Zhang e Wang (2021) explora uma abordagem que foca na recuperação e reutilização de casos anteriores para gerar soluções otimizadas, usando métricas de similaridade baseadas na proporção do subgrafo comum máximo.

Como visto, várias abordagens têm sido propostas na literatura científica para encontrar soluções otimizadas para o problema do agendamento universitário. Um mapeamento dessas estratégias é apresentado na Tabela 2.5.

Tabela 2.5: Mapeamento das abordagens

Id.	Abordagem	Artigos Referenciados	Total
1	Construtive Heuristics	[8],[13],[19],[23],[27],[32],[35],[44],[62],[63],[66]	11
2	Simulated annealing	[1],[7],[9],[11],[21],[29],[30],[38],[72],[61]	10
3	Integer Programming	[14],[18],[22],[29],[30],[37],[40],[59],[60]	9
4	Iterated local search	[1],[9],[16],[24],[25],[31],[50],[64],[66]	9
5	Tabu Search	[5],[7],[9],[19],[28],[61],[63],[67]	8
6	Genetic Algorithm	[3],[8],[17],[33],[34],[43],[48]	7
7	Hill climbing	[23],[24],[62],[69]	4
8	Variable Neighborhood Search	[28],[30],[44],[61]	4
9	Fuzzy Logic	[31],[35],[52]	3
10	Greedy algorithm	[5],[16],[25]	3
11	Memetic algorithm	[50],[64],[69]	3
12	Multi-agent system	[21],[26],[37]	3
13	Particle Swarm Optimization	[35],[51],[69]	3
14	Answer Set Programming	[12],[45]	2
15	Ant System	[55],[58]	2
16	Case-based Reasoning	[41],[62]	2
17	Cuckoo Search	[10],[46]	2
18	Harmonic Search Algorithm	[69],[70]	2
19	Hyper-heuristics	[8],[24]	2
20	Neural network	[33],[34]	2

Tabela 2.5 – continuação da página anterior

Id.	Abordagem	Artigos Referenciados	Total
21	Belief propagation algorithm	[47]	1
22	Best First Search Algorithm	[42]	1
23	Clustering	[31]	1
24	Constraint Satisfaction Model	[39]	1
25	Epsilon-constraint method	[15]	1
26	Firefly algorithm	[32]	1
27	First Fit Decreasing	[16]	1
28	Great Deluge	[21]	1
29	MATLAB	[20]	1
30	Monte Carlo tree search	[19]	1
31	Spotted Hyena Optimizer	[38]	1
32	TOPSIS	[15]	1

QP2 - Quais são os algoritmos mais adequados para a instituição? Observou-se que há formas diferentes de se resolver o problema do agendamento universitário e que não há um algoritmo mais adequado para todas as situações. Neste sentido, considera-se que as abordagens que podem ser as mais adequadas para a instituição estejam no universo das heurísticas construtivas como nos trabalhos de Burke et al. (2007), Qu e Burke (2009), Pillay e Özcan (2019), *Simulated Annealing* presente nos trabalhos de Abramson (1991), Chiarandini et al. (2006) e Song et al. (2018) e programação inteira nos estudos de Daskalaki, Birbas e Housos (2004), Daskalaki e Birbas (2005) e Siddiqui, Raza e Tariq (2018) e Rappos et al. (2022), por serem as mais frequentemente utilizadas.

QP3 - Quais são as modelagens do problema?

Uma estratégia comum para a modelagem do problema é a representação matemática, frequentemente empregada como problema de otimização, seja ela linear ou inteira. Esta abordagem define as variáveis, as restrições e a função objetivo de forma matemática, como evidenciado nos trabalhos de Daskalaki, Birbas e Housos (2004), Daskalaki e Birbas (2005) e Lü e Hao (2010). Em contrapartida, há estudos que optam por representações computacionais, utilizando bancos de dados para representar os elementos como salas de aula, horários e recursos, exemplificados por Ren e Li (2022), Siddiqui, Raza e Tariq (2018) e Houhamdi et al. (2019). Além dessas, abordagens alternativas têm sido exploradas, como a utilização de algoritmos de Busca Harmônica por Al-Betar e Khader (2012), onde os elementos são representados por matrizes, ou ainda, a modelagem por meio de programas lógicos no modelo de Aini, Saptawijaya e Aminah (2017), que utiliza *Answer*

Set Programming e o uso de *arrays* de classes com algoritmos genéticos (ASSI; HALAWI; HARATY, 2018).

QP4 - Quais são as *constraints* mais utilizadas?

Os requisitos e preferências para a criação do agendamento são comumente referidos como *constraints* ou restrições. De acordo com Lewis (2007), as *constraints* podem ser classificadas em duas categorias principais:

As *hard constraints* representam restrições que devem ser estritamente satisfeitas para que uma solução seja reconhecida como viável no contexto do problema do agendamento universitário (BURKE et al., 1997). Nesta pesquisa, verificou-se que os estudos analisados demonstram elevada conformidade com as *constraints* elencadas por McCollum et al. (2010), estabelecendo-as assim como as mais empregadas no domínio (ver Tabela 2.6).

Tabela 2.6: *Hard constraints* mais utilizadas

Regra	Descrição
Aulas	Todas as aulas de um curso devem ser agendadas e devem ser atribuídas a períodos distintos.
Ocupação da sala	Duas aulas não podem acontecer na mesma sala no mesmo período.
Conflitos	As aulas de cursos do mesmo currículo ou ministradas pelo mesmo professor devem ser todas agendadas em períodos diferentes
Disponibilidade	Se o professor do curso não estiver disponível para ministrar esse curso em um determinado período, a aula não poderá ser agendada nesse período

Por outro lado, as *soft constraints* são restrições suaves que, embora desejáveis, não são estritamente necessárias para a viabilidade da solução (BURKE et al., 1997). Estas podem incluir, por exemplo, a preferência de um professor por um horário específico ou a proximidade de salas para turmas consecutivas. Violar uma *soft constraint* não torna a solução inviável, mas pode afetar sua qualidade ou aceitabilidade. As preferências mais utilizadas foram apresentadas por McCollum et al. (2010) e podem ser vistas na tabela 2.7.

Além das *constraints* apresentadas aqui, foi feito um levantamento mais amplo sobre estes elementos. Uma tabela mais detalhes constando todas as *constraints* identificadas nesta pesquisa é apresentada no apêndice A.

2.4 Trabalhos relacionados

O estudo de Siddiqui, Raza e Tariq (2018), introduz um sistema de suporte à decisão em grupo para a preparação do período acadêmico em uma universidade do Oriente Médio.

Tabela 2.7: *Soft constraints* mais utilizadas.

Regra	Descrição
Mínimo de dias de trabalho	As aulas de cada curso devem ser distribuídas nos dias mínimos especificados.
Compactação curricular	As aulas pertencentes a um currículo devem ser adjacentes umas às outras (ou seja, em períodos consecutivos).
Capacidade da sala	Para cada aula, o número de alunos que frequentam o curso deve ser inferior ou igual ao número de assentos de todas as salas que hospedam suas aulas.
Estabilidade da sala	Todas as aulas de um curso devem ser ministradas na mesma sala.

Este sistema automatiza a tarefa de criação de horários acadêmicos usando um modelo de programação inteira mista multiobjetivo.

De forma semelhante a proposta desta pesquisa, o sistema de Siddiqui permite a participação de diferentes perfis de usuários no processo de criação de agendamento, que inclui departamentos acadêmicos, unidade de geração de agendamento, professores e gestores. Outra característica comum entre ambos é a capacidade de gerar relatórios que facilitam a análise e avaliação das propostas de agendamento e dos resultados alcançados.

O trabalho de Siddiqui, possui particularidades na abordagem como por exemplo a ênfase ao seccionamento dos estudantes, fazendo uma distinção clara entre homens e mulheres, uma característica que não está presente nesta pesquisa. Além disso, ao passo que a flexibilidade na criação de contextos independentes é uma funcionalidade presente nesta proposta, em seu modelo não é contemplado. Enquanto a abordagem proposta neste trabalho oferece a possibilidade de integração com outras estratégias de gestão acadêmica, a de Siddiqui não apresenta essa flexibilidade, limitando-se a uma metodologia mais rígida e preestabelecida.

A abordagem baseada em um Sistema Multiagente (MAS) é explorada por Houhamdi et al. (2019). Em seu estudo, Houhamdi propôs um sistema que se estrutura de forma hierárquica, abrangendo desde o nível da universidade até o nível individual de um curso. Três agentes colaborativos trabalham em conjunto para elaborar os horários, considerando critérios como a demanda dos alunos e a disponibilidade de instrutores.

Houhamdi, assim como esta pesquisa, também reconhece a necessidade de adaptabilidade e flexibilidade nos sistemas de agendamento, dada a dinâmica e as múltiplas variáveis envolvidas no ambiente universitário. Em concordância, nossos trabalhos consideram a importância da integração com outras estratégias de gestão acadêmica e a necessidade de considerar as preferências e restrições específicas de cada instituição.

Porém, em sentido diferente, nosso modelo oferece uma abordagem mais abrangente incluindo elementos como *Feedback* e Avaliação, que permitem ajustes e melhorias contí-

nuas no modelo. Além disso, nosso estudo introduz a Predição, que utiliza dados históricos e tendências para prever necessidades futuras de agendamento, algo que não é explicitamente abordado por Houhamdi. Deste modo, esta proposta apresenta uma estrutura mais flexível e adaptável que pode ser personalizada de acordo com as necessidades específicas da instituição.

O desafio do agendamento de professores em múltiplos departamentos é o foco do estudo de Babaei, Karimpour e Hadidi (2019). Os autores propõe um algoritmo híbrido que combina técnicas de lógica *fuzzy* e algoritmos de agrupamento para agendar aulas em comum entre departamentos.

O modelo de Babaei, Karimpour e Hadidi possui similaridades com a estrutura do modelo proposto nesta pesquisa. Ambos são compostos por elementos distintos que permitem interação com o usuário e otimização, por exemplo. Em ambas as propostas, a interação humana é necessária. No modelo deles, os professores são convidados a expressar suas preferências, enquanto coordenadores são responsáveis por validar tais entradas. Essa dinâmica de interação entre diferentes atores no ambiente universitário confere um grau de realismo e aplicabilidade aos modelos, permitindo que eles se adaptem mais eficientemente a cenários reais.

Por outro lado, nosso modelo é mais abrangente, pois agrega elementos que permitem integração, comunicação, predição e contextualização, além da possibilidade da geração de relatórios. Tais características não foram observadas na pesquisa de Babaei, Karimpour e Hadidi.

Chen et al. (2021) apresenta um sistema de agendamento de aulas que integra algoritmos genéticos. Este sistema opera em uma arquitetura de três camadas e oferece tanto ajustes manuais quanto automáticos no agendamento, proporcionando flexibilidade para acomodar diversas restrições.

O trabalho de Chen enfatiza a importância de se considerar as preferências dos docentes e alunos, bem como a otimização do uso dos recursos disponíveis, como salas de aula e laboratórios. A nossa proposta, também aborda a questão da indicação de preferências, mas com um foco particular mais voltado para adaptabilidade e flexibilidade, permitindo que o modelo seja ajustado para atender às necessidades específicas de diferentes instituições. O nosso modelo destaca a importância de se manter contextos distintos para facilitar a adaptação do modelo às variadas demandas e complexidades das instituições acadêmicas. Similarmente, o apresentado por Chen também permite ajustes com base nas necessidades e preferências dos usuários, embora com um enfoque mais voltado para a otimização.

Nosso estudo apresenta uma abordagem distintiva ao problema do agendamento universitário, focando na modularidade e adaptabilidade. Ele introduz o conceito de contexto

como um elemento para a flexibilidade do modelo. Cada contexto é composto por um conjunto de elementos, como usuários, espaços físicos e eventos, e deve ter um responsável pela sua manutenção. Este enfoque permite que o modelo seja adaptado para atender às necessidades específicas de diferentes departamentos ou programas acadêmicos, algo que não foi identificado no trabalho de Chen. Além disso, nosso trabalho oferece uma visão mais detalhada sobre a personalização, permitindo que diferentes categorias de usuários, como Coordenadores, Gestores e Administradores, interajam e adaptem os parâmetros do modelo às suas realidades específicas.

Capítulo 3

Contextualização

Fundada em 1962, a Universidade de Brasília (UnB) é uma instituição pública de ensino superior que se destaca por sua trajetória de inovação e pelo compromisso com o ensino, a pesquisa e a extensão. Ao longo dos anos, a UnB consolidou-se como um dos principais centros de ensino superior do Brasil, oferecendo uma ampla gama de cursos, centros de pesquisa e projetos de impacto social. Este capítulo visa fornecer uma visão abrangente da UnB, incluindo sua estrutura administrativa, características institucionais e aspectos históricos que moldaram sua evolução. Esses elementos são fundamentais para compreender as necessidades específicas da universidade, bem como os desafios e oportunidades que impactam o desenvolvimento de um sistema de agendamento eficiente e adaptado ao contexto universitário. O ambiente institucional da UnB serve, assim, como base para esta pesquisa, proporcionando um contexto real e aplicável para o desenvolvimento de um sistema de agendamento universitário.

3.1 Universidade de Brasília

Conforme seu portal institucional UnB (2023a), Brasília, a capital do Brasil, tinha apenas dois anos de existência quando a UnB foi inaugurada em 21 de abril de 1962. A universidade nasceu com a promessa de reinventar a educação superior no Brasil, engajando-se na transformação socioeconômica e cultural do país.

A UnB foi concebida por Darcy Ribeiro, Anísio Teixeira e Oscar Niemeyer. Darcy Ribeiro, um antropólogo de renome, delineou as bases ideológicas da instituição. O educador Anísio Teixeira contribuiu com o modelo pedagógico, enquanto o arquiteto Oscar Niemeyer foi responsável pelo design do campus. Os fundadores desejavam criar uma experiência educadora que unisse o que havia de mais moderno em pesquisas tecnológicas com uma produção acadêmica capaz de melhorar a realidade brasileira.

A sua estrutura administrativa foi organizada como uma Fundação¹, um conceito inovador na época, visando garantir sua autonomia frente ao burocratismo ministerial. Essa autonomia continua sendo uma característica fundamental da gestão universitária atual.

Desde o seu início, a UnB buscou integrar inovações tecnológicas com produção acadêmica direcionada à melhoria das condições brasileiras. Seu Plano Orientador de 1962, serviu como uma espécie de Carta Magna, definindo sua missão que e ainda permanece em vigor (RIBEIRO; TEIXEIRA, 1962):

"Ser uma universidade inovadora e inclusiva, comprometida com as finalidades essenciais de ensino, pesquisa e extensão, integradas para a formação de cidadãos e cidadãs éticos e qualificados para o exercício profissional e empenhados na busca de soluções democráticas para questões nacionais e internacionais, por meio de atuação de excelência."

Esta foi a base para a criação do Projeto Político-Pedagógico Institucional de 2018, que delinea sua estrutura administrativa e princípios orientadores para a gestão universitária, em continuidade com sua missão original estabelecida em 1962. Entre os princípios fundamentais estão a descentralização, transparência, legitimidade, legalidade, impessoalidade, publicização e probidade, os quais buscam assegurar uma governança democrática da instituição. Estes princípios norteiam as ações e atividades da sua estrutura administrativa, das unidades acadêmicas, dos centros e núcleos de pesquisa e ensino em seus *campi* (UNB, 2018).

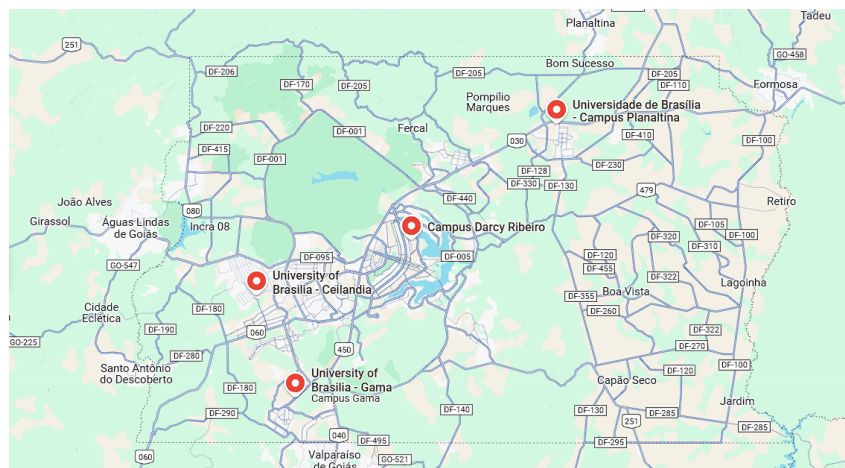


Figura 3.1: Os *campi* da UnB estão distribuídos em várias cidades.

Após formalizar sua participação no Programa de Apoio a Planos de Reestruturação e Expansão das Universidades Federais (Reuni) no ano de 2007, a UnB empreendeu uma

¹No contexto jurídico brasileiro, uma fundação é uma pessoa jurídica de direito privado instituída com um patrimônio destinado a um fim específico, geralmente de interesse social, cultural ou científico, e que não possui fins lucrativos. Esse modelo de organização permite maior autonomia administrativa e financeira em comparação com outras estruturas governamentais.

expansão institucional ao estabelecer três novos *campi*: Planaltina, Gama e Ceilândia. Estes vieram a complementar o já existente Campus Darcy Ribeiro, ampliando assim o alcance e a diversidade educacional da instituição.

O *campus* da UnB em Planaltina foi inaugurado em maio de 2006, integrando-se a um projeto mais amplo de expansão da universidade (UNB, 2023g). Especializado na oferta de cursos nas áreas de Ciências Naturais e Agrárias, bem como em licenciatura em Educação do Campo, ele está localizado a 37 km do *campus* Darcy Ribeiro (UNB, 2023b). Mantendo uma relação estreita com a comunidade local, que possui um perfil voltado para atividades rurais e uma rica tradição cultural, o *campus* conta com um quadro de 117 professores e 51 servidores técnico-administrativos, distribuídos em cinco áreas de conhecimento, Ciências Exatas; Ciências da Vida e da Terra; Educação e Linguagens; Ciências Sociais e Humanas; e Ciências Sociais Aplicadas e Tecnologia. A oferta acadêmica inclui cinco cursos de graduação interdisciplinares Licenciatura em Ciências Naturais (diurno); Licenciatura em Ciências Naturais (noturno); Licenciatura em Educação do Campo; Bacharelado em Gestão Ambiental; e Bacharelado em Gestão do Agromercado. Além dos cursos de graduação, são ofertados seis programas de pós-graduação, Ciência de Materiais (PPGCIMA); Meio Ambiente e Desenvolvimento Rural (PPGMA-DER); Ciências Ambientais (PPGCA); Gestão Pública (PPGGP); Gestão e Regulação de Recursos Hídricos (Prof.Água); e Sustentabilidade junto a Povos e Territórios Tradicionais (PPGMESPT).

O *campus* da UnB no Gama foi inaugurado em agosto de 2008, dando continuidade ao seu programa de expansão (UNB, 2023f). Especializado na oferta de cursos de Engenharias, ele está localizado a 37 km do *campus* Darcy Ribeiro (UNB, 2023b). Este *campus* conta com um quadro de 133 professores e 43 servidores técnico-administrativos. A oferta acadêmica de graduação inclui diversos cursos como Engenharia Aeroespacial, Engenharia Automotiva, Engenharia Eletrônica, Engenharia de Energia e Engenharia de Software. Além dos cursos de graduação, são ofertadas pós-graduação em Engenharia Clínica e Modelagem de Sistemas Complexos, bem como mestrados em Engenharia Biomédica e Integridade de Materiais da Engenharia, ampliando assim a pluralidade e o alcance acadêmico do *campus*.

O *campus* da UnB em Ceilândia também foi inaugurado em agosto de 2008. Tem como especialidade a oferta de cursos na área da Saúde (UNB, 2023e). Localizada a uma distância de 31 km do *campus* Darcy Ribeiro, este *campus* se beneficia da experiência acumulada pela Faculdade de Ciências da Saúde da mesma instituição (UNB, 2023b). Este *campus* conta com um quadro de 173 de professores e 71 servidores técnico-administrativos. Sua oferta acadêmica de graduação abrange diversos cursos como Enfermagem; Farmácia; Fisioterapia; Fonoaudiologia; Saúde Coletiva; e Terapia Ocupacional. Além dos cursos

de graduação, são ofertadas pós-graduação em Análises Clínicas; Fisioterapia Traumatológica; Fisioterapia Cardiorrespiratória; e Terapia Ocupacional em Contexto Hospitalar, bem como os programas de pós-graduação em Ciências da Reabilitação (PPGCR) e Ciências e Tecnologias em Saúde (PPGCTS).

A UnB possui em sua estrutura 12 institutos e 14 faculdades, 55 departamentos, 682 laboratórios, uma Biblioteca Central e cinco bibliotecas setoriais, uma fazenda e duas unidades do Hospital Veterinário, 7 centros vinculados à Reitoria e 14 órgãos complementares e auxiliares (UNB, 2023b). Os Campi são equipados com instalações que englobam salas de aula, laboratórios multidisciplinares e laboratórios de informática, criando assim um ambiente propício para atividades de ensino, pesquisa e extensão. Um panorama resumido do que foi apresentado no Plano de Desenvolvimento Institucional de 2023-2028 (UNB, 2023b), pode ser visto na tabela 3.1.

Tabela 3.1: Panorama da UnB em 2022.

Item	Quantidade
Discentes de graduação	47.966
Discentes de pós-graduação	9.636
Docentes	2608
Grupos de Pesquisa	660
Ações de Extensão	2.252
Cursos Ofertados de Graduação	132
Cursos Ofertados de pós-graduação	168
Cursos Ofertados de Residência Médica	49

A sua estrutura acadêmico-administrativa é tripartida composta por institutos voltados para pesquisa e pós-graduação, faculdades focadas em profissionalização e órgãos complementares que fornecem serviços de apoio. Cada unidade acadêmica oferece cursos de graduação e pós-graduação, bem como atividades de extensão e pesquisa. O regime curricular é semi-seriado, com um teto máximo de 70% de disciplinas obrigatórias e 30% compostas por disciplinas optativas e módulos livres. A estrutura promove a convergência e integração entre modalidades de ensino presencial e a distância e permite flexibilidade na integralização, aproveitamento e equivalência de disciplinas. Cada curso ou programa é coordenado por um colegiado específico e a pesquisa é orientada por grandes linhas prioritárias, mas também estimula iniciativas individuais (UNB, 2023b). Dessa forma, a UnB busca responder às demandas dinâmicas da comunidade universitária e da sociedade, englobando ensino, pesquisa e extensão.

3.2 Prefeitura da UnB

A Prefeitura da UnB (PRC) desempenha um papel central na administração e manutenção das infraestruturas acadêmicas e administrativas da universidade (UNB, 2023h). Suas responsabilidades variam desde a gestão de edifícios e áreas comuns até a zeladoria e paisagismo. Focada em sustentabilidade e segurança patrimonial, a PRC também é encarregada da manutenção de espaços comerciais dentro dos campi. Além disso, ela atua no estímulo ao uso consciente de espaços comuns pela comunidade universitária, monitora o consumo de água e energia para promover práticas sustentáveis e é responsável pela sinalização predial e urbana.

Atualmente, a PRC tem como atribuições a geração e a manutenção do agendamento acadêmico no campus Darcy Ribeiro. A entidade busca otimizar a ocupação dos espaços físicos designados para atividades acadêmicas, bem como assegurar a disseminação dos cronogramas de aulas para toda a comunidade acadêmica, visando uma gestão mais eficiente dos recursos institucionais. Nesse contexto, a atuação da PRC vai além da manutenção de infraestruturas, posicionando-se como um agente ativo na promoção da celeridade e eficiência nos serviços sob sua responsabilidade (UNB, 2023c).

3.3 Secretaria de Tecnologia da Informação

A Secretaria de Tecnologia da Informação (STI), antes conhecida como Centro de Processamento de Dados (CPD), foi estabelecida em 1991 e atua como órgão meio responsável pela parte tecnológica da UnB. A STI visa fornecer soluções de TI que assegurem a disponibilidade, integridade, confiabilidade e autenticidade das informações nos sistemas informatizados da universidade. Entre seus objetivos centrais estão o suporte ao ensino, pesquisa e extensão, bem como a promoção do compartilhamento de recursos computacionais com redes de pesquisa (inter)nacionais. A STI também se ocupa do desenvolvimento, implementação e manutenção de sistemas em diversas plataformas (UNB, 2023j).

Conforme pode ser visto na Figura 3.2, a STI é composta por uma Direção, uma Vice-Direção e um Conselho Consultivo, cada um com responsabilidades bem definidas (UNB, 2023).

A Direção é responsável pelo planejamento e supervisão das atividades de TI e segurança, enquanto a Vice-Direção age como substituta do Diretor e auxilia na coordenação interna. O Conselho Consultivo assessora em aspectos políticos e estratégicos. Além disso, há outras três unidades no organograma: a Diretoria de Operações e Serviços (DOS), a Coordenação de Gestão e Planejamento (CGESP) e a Diretoria de Sistemas da Informação (DSI). Esta última possui uma subdivisão específica, a Coordenação de Sistemas Acadê-

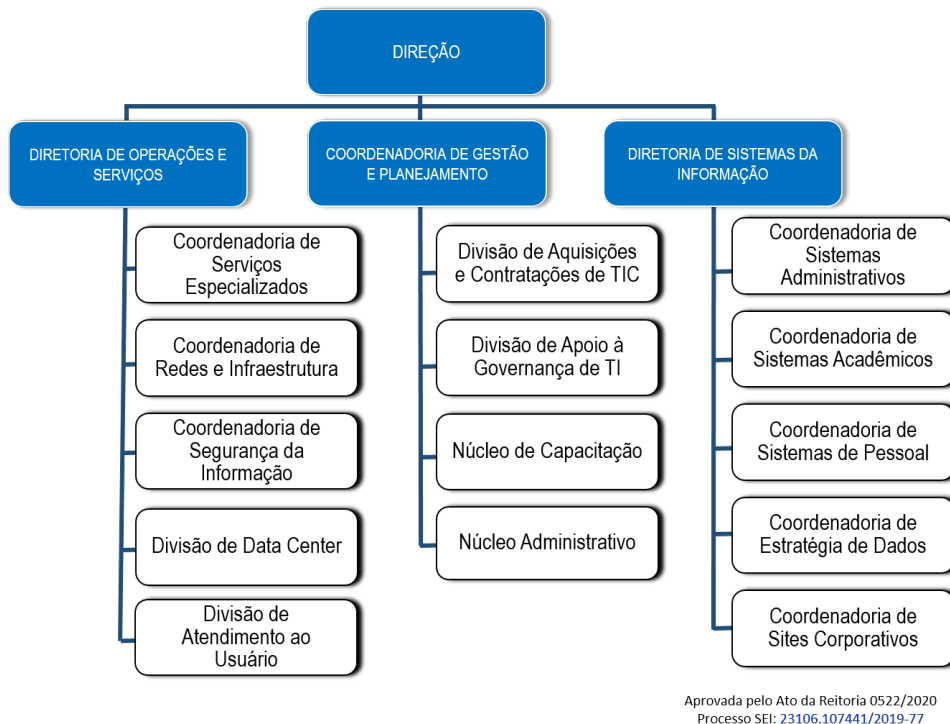


Figura 3.2: Organograma da STI.

micos, que é encarregada do desenvolvimento e manutenção dos sistemas acadêmicos da universidade.

Dentre os vários sistemas mantidos pela DSI estão o Sistema Integrado de Gestão das Atividades Acadêmicas (SIGAA) e o Sistema de Ensalamento (SIGEN). O SIGAA é um sistema que faz parte de um pacote de ferramentas *on-line* (SIG-UnB), adotado para unificar e modernizar os vários sistemas internos de gestão da UnB (UNB, 2023i), já o SIGEN é a ferramenta tecnológica utilizada atualmente para resolver o problema de alocação de turmas a espaços físicos.

3.4 Sistemas Institucionais Integrados de Gestão

Os Sistemas Institucionais Integrados de Gestão (SIG) criados pela Universidade Federal do Rio Grande do Norte (UFRN) são um conjunto de ferramentas tecnológicas destinadas a gerenciar diversos aspectos administrativos e acadêmicos da instituição (UFRN, 2023b). Estes sistemas visam fornecer uma infraestrutura para a realização de tarefas acadêmicas, administrativas, financeiras, controle de recursos humanos, entre outros. Atualmente, o SIG é adotado por mais de trinta Instituições Federais de Ensino Superior, demonstrando sua relevância no contexto acadêmico brasileiro (UFRN, 2023a). Dentre os sistemas adotados na UnB destacam-se os Sistema Integrado de Gestão de Recursos

Humanos (SIGRH), Sistema Integrado de Gestão de Patrimônio, Administração e Contratos (SIPAC), Sistema de Gestão Administrativa (SIGAdmin) e o Sistema Integrado de Gestão de Atividades Acadêmicas (SIGAA).

O SIGAA é uma plataforma desenvolvida com o objetivo de integrar diversas atividades acadêmicas em um único ambiente virtual (UFRN, 2023b). Ele tem a capacidade de informatizar uma gama de processos na esfera acadêmica, abrangendo desde a graduação até a pós-graduação, em níveis *stricto* e *lato sensu*. Adicionalmente, ele gerencia atividades em ensino técnico, médio e infantil. A plataforma também é equipada com um Ambiente Virtual de Aprendizado, que facilita o ensino a distância. Esta plataforma oferece portais específicos para uma variedade de *stakeholders* dentro da instituição, incluindo reitoria, professores, alunos, tutores de ensino a distância, e coordenações de programas de graduação e pós-graduação (UFRN, 2023b).

Entretanto, apesar da abrangência e modularidade do SIGAA, há uma lacuna significativa relacionada à resolução automatizada do problema do agendamento universitário. A plataforma não oferece um módulo ou funcionalidade que automatize ou otimize o processo de alocação de salas, horários e outros recursos para as turmas. A única opção disponível é um campo onde administradores podem inserir manualmente a informação de localização quando criam uma turma. Esta abordagem manual não só é susceptível a erros, como também não considera variáveis complexas, como a disponibilidade de salas, conflitos de horários entre turmas e professores, e outras restrições pertinentes ao ambiente acadêmico. Para mitigar essa lacuna, UnB desenvolveu o SIGEN, uma ferramenta projetada para tratar especificamente das complexidades associadas ao agendamento universitário na instituição. No entanto, esta solução, que foi recentemente implementada, ainda apresenta uma série de limitações que restringem sua eficiência e abrangência.

3.5 Sistema de Ensalamento

Visando preencher a lacuna deixada pelo SIGAA no que diz respeito ao agendamento de turmas, a UnB desenvolveu recentemente um sistema próprio denominado Sistema de Ensalamento (SIGEN) (UNB, 2023d). O processo atual de agendamento na UnB representa um esforço significativo para centralizar e organizar a alocação de horários e espaços físicos para as atividades acadêmicas. No entanto, a solução ainda apresenta uma série de limitações que restringem sua eficiência e abrangência.

Atualmente, o sistema está operacional, porém atende apenas parte do *campus* Darcy Ribeiro, deixando de fora os outros *campi* da UnB. Esta limitação geográfica impede uma integração completa e inclusiva de todas as unidades acadêmicas da universidade, criando desequilíbrios e possíveis gargalos no processo de agendamento.

Além disso, o sistema atual não foi projetado para ser genérico. Ele está fortemente acoplado aos sistemas existentes da UnB contendo em sua lógica aspectos específicos da universidade. Esta característica impede que a solução seja facilmente adaptada e implementada em outras instituições acadêmicas, limitando seu potencial de escalabilidade e adaptabilidade a diferentes contextos e necessidades institucionais.

Outra limitação significativa é a ausência de uma abordagem baseada em contextos. O processo de agendamento é executado de uma só vez para todos os cursos que participam do processo, sem a possibilidade de segmentar e personalizar o agendamento com base em diferentes contextos e requisitos específicos de cada curso ou departamento. Esta abordagem monolítica pode levar a soluções subótimas, onde as necessidades individuais e nuances de diferentes cursos não são adequadamente atendidas.

O sistema atual não emprega algoritmos de otimização, uma lacuna significativa que impede a realização de um agendamento verdadeiramente eficiente e otimizado. A otimização poderia permitir uma alocação de recursos mais racionalizada, levando em consideração uma série de variáveis, como a disponibilidade de espaços físicos, as preferências dos professores e alunos, entre outros fatores críticos que poderiam melhorar significativamente a qualidade do agendamento.

3.6 Considerações finais

Diante deste cenário, observa-se que, embora a estratégia utilizada para gerar agendamento na UnB represente um passo na direção certa, ainda há um caminho a percorrer para se alcançar uma solução verdadeiramente integrada, eficiente e inclusiva. Neste sentido, a necessidade de se ter um melhor entendimento das limitações atuais da sua estratégia se torna evidente.

Para superar essas barreiras, esta proposta visa a criação de um modelo mais flexível e adaptável, que possa não apenas atender às necessidades diversificadas da comunidade acadêmica da UnB, mas também oferecer uma abordagem que possa ser adaptada a outras realidades. A integração de algoritmos de otimização e a concepção de uma abordagem baseada em contextos podem ser passos importantes nesta direção, pavimentando o caminho para um melhor modelo de agendamento.

Assim, tendo em vista as limitações identificadas no SIGEN e considerando a ampla implementação do SIGAA em diversas instituições de ensino superior no Brasil (UFRN, 2023a), a presente proposta de pesquisa ganha relevância devido ao seu potencial de adaptação a diferentes cenários. Sua adaptabilidade e flexibilidade a tornam propícias a adesão em ambientes acadêmicos que já empregam o SIGAA, como é o caso da UnB. Essa característica expande a abrangência e eleva o impacto potencial da pesquisa, com pos-

sibilidade de catalisar melhorias sistêmicas nas estratégias de agendamento universitário em âmbito nacional.

Capítulo 4

O modelo SARA

O problema do agendamento universitário envolve múltiplos fatores e exige uma abordagem cuidadosa, dado o seu impacto direto e indireto na gestão e na qualidade da experiência educacional em uma instituição acadêmica (DUNKE; NICKEL, 2023), (ASSI; HALAWI; HARATY, 2018). Manter o agendamento não é apenas determinar a alocação de espaço e tempo para atividades educacionais, mas também entrar em interação com uma variedade de restrições e preferências, incluindo, mas não limitado a, disponibilidade de recursos, horários dos docentes, e eventos especiais (SONG et al., 2018). Este capítulo apresenta o modelo do Sistema de Agendamento de Recursos Acadêmicos (SARA), desenvolvido para estruturar o problema de agendamento universitário, introduzindo um mecanismo de apoio à gestão acadêmica.

4.1 Modelo proposto

À medida que as instituições acadêmicas evoluem, suas necessidades e prioridades também se transformam. Portanto, o modelo de agendamento adotado necessita ser adaptável e robusto (AKKAN; GüLCü, 2018). O SARA foi concebido visando adaptabilidade, permitindo modificações e ajustes sem a necessidade de grandes revisões ou reconfigurações. Reconhecendo a necessidade de interconexão, este modelo permite integrar-se com outros mecanismos utilizados na gestão acadêmica.

O objetivo do SARA é auxiliar na resolução do problema do agendamento universitário. Visando alcançar esse objetivo, são apresentados os seguintes elementos presentes na Figura 4.1:

- **Personalização:** Permite que as instituições façam configurações de acordo com suas necessidades específicas, garantindo que o agendamento reflita as particularidades de cada universidade.

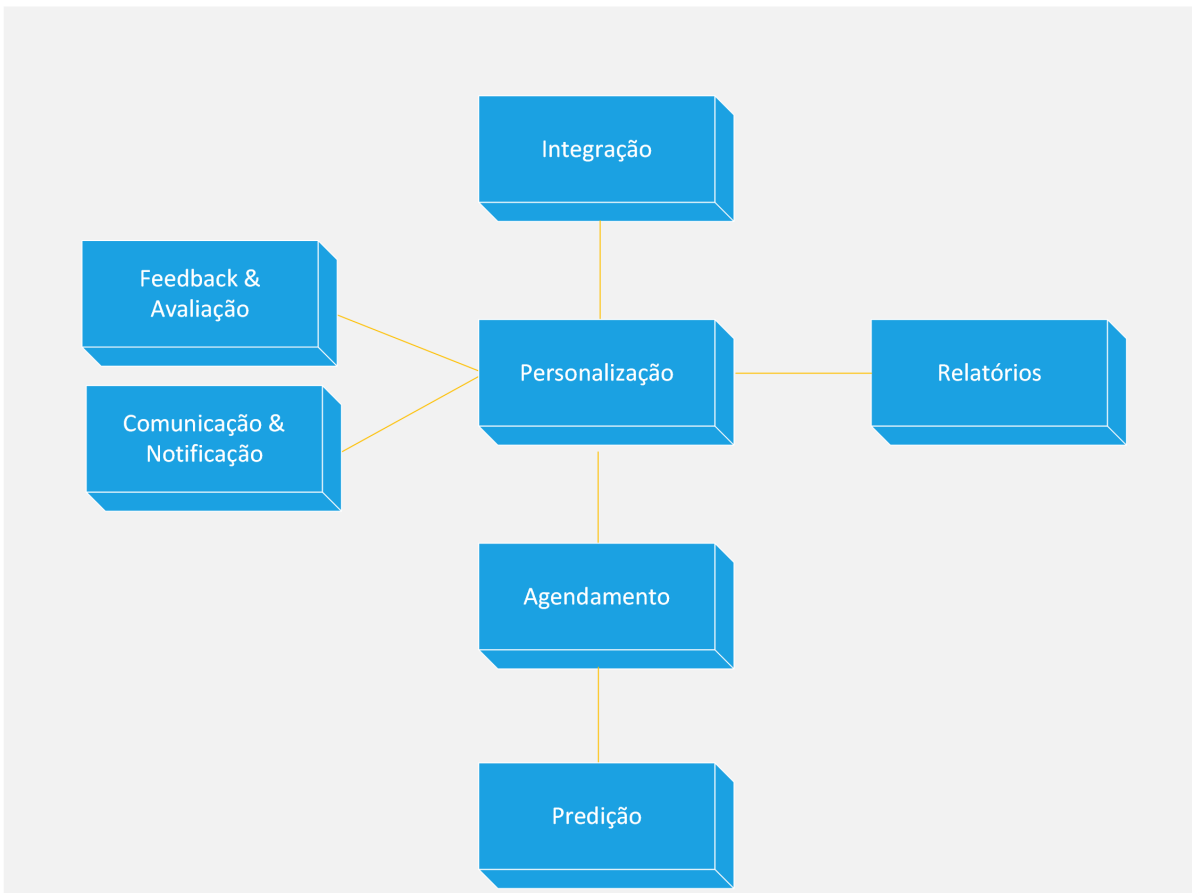


Figura 4.1: Visão geral do modelo SARA.

- **Agendamento:** Onde os horários são definidos com base em algoritmos de otimização, considerando todas as *constraints* e restrições.
- **Feedback e Avaliação:** Viabiliza o fornecimento de percepções sobre o agendamento, permitindo ajustes e melhorias contínuas.
- **Predição:** Utiliza agendamento anteriores e tendências para prever necessidades futuras, ajudando na tomada de decisões proativas.
- **Integração:** Assegura que o modelo possa se comunicar com outros mecanismos de gestão da instituição.
- **Comunicação e Notificação:** Mantém todos os *stakeholders* informados sobre agendamentos, mudanças e outros eventos relevantes.
- **Relatórios:** Geração de relatórios sobre o agendamento, fornecendo *insights* para a gestão.

4.2 Partes do modelo

Por meio dos elementos apresentados anteriormente, este modelo pode ser adaptado para diferentes realidades, tornando viável a criação de agendamentos para diversas instituições. A seguir, apresenta-se uma descrição mais detalhada das partes que o compõem.

4.2.1 Personalização

A personalização é a parte do SARA que é alimentada com informações específicas de uma instituição. Essas informações, que podem variar conforme as necessidades institucionais, incluem detalhes sobre os espaços físicos, turmas, disciplinas e currículo, sendo essenciais para que o modelo seja capaz de gerar agendamentos para diferentes cenários.

Nesta parte do modelo os usuários podem realizar as suas atividades, interagindo e adaptando os seus parâmetros às suas realidades de forma semelhante ao proposto em Siddiqui, Raza e Tariq (2018). Para isto, o SARA dispõe de categorias de usuários, como Coordenadores, Gestores e Administradores, permitindo uma melhor concentração e segregação de características e atividades específicas dos indivíduos envolvidos no planejamento do agendamento da instituição, assim como mencionado no trabalho de Burke et al. (1997).

Cada instituição possui uma cultura, uma história e um conjunto de valores únicos que influenciam diretamente a maneira como o agendamento é percebido e implementado. A Personalização, ao permitir uma interação dos usuários com o modelo, também serve como uma ponte para incorporar esses aspectos intangíveis no agendamento. Isso garante que, além de atender às necessidades logísticas e operacionais, o modelo também respeite e reflita a identidade e os valores da instituição.

Deste modo, as características da Personalização são descritas da seguinte forma:

Eventos

No contexto deste modelo, um evento pode ser definido como uma aula, palestra, seminário, maratona ou qualquer outra reunião que requeira a reserva de um espaço físico por um período determinado (BURKE et al., 2007). Esta característica envolve a organização e manutenção desses encontros, abrangendo também aqueles que não foram programados como ofertas de vagas em turmas, tais como Ações de Extensão, Palestras, Seminários, Maratonas, entre outros. Os eventos podem ser realizados independentemente das aulas regulares e podem abarcar ocorrências não planejadas. Assim, contempla-se a possibilidade de integrar essas ocorrências no planejamento do agendamento institucional. Além disso, é considerada a dinâmica e a diversidade dos eventos acadêmicos, onde cada ocorrência tem suas próprias necessidades e requisitos. Um seminário pode necessitar de

equipamentos audiovisuais avançados, enquanto uma maratona pode exigir espaços mais amplos e abertos. A capacidade de adaptar-se a essas variáveis é importante que cada evento seja bem-sucedido e atenda às expectativas dos participantes.

Gerenciar estes eventos implica na coordenação entre diferentes departamentos e partes interessadas. Por exemplo, a reserva de um laboratório para uma aula prática pode exigir a coordenação com o departamento responsável pelo laboratório, enquanto a organização de uma palestra pode envolver a colaboração com palestrantes externos e patrocinadores. Esta atividade proporciona uma alocação apropriada e que todos os recursos necessários, humanos ou materiais, estejam disponíveis e coordenados.

Lidar com Eventos envolve a capacidade de responder e adaptar-se a mudanças e imprevistos, já que eles podem ser cancelados, reagendados ou modificados com base em uma variedade de fatores, como condições climáticas, indisponibilidade de palestrantes ou mudanças no currículo. A capacidade de gerenciar essas mudanças e comunicá-las a todas as partes interessadas é importante para a continuidade das operações e minimizar interrupções e inconvenientes.

Preferências

Esta característica envolve o fornecimento de preferências específicas de localização, professores e discentes (SIDDIQUI; RAZA; TARIQ, 2018). Essas preferências são utilizadas para a otimização da logística do agendamento, como, por exemplo, na minimização do deslocamento de professores entre aulas. Além disso, esta atividade pode se estender a outros aspectos do ambiente acadêmico. Por exemplo, professores, podem ter preferências relacionadas a horários específicos do dia devido a compromissos pessoais ou outras atividades acadêmicas. Alunos, por sua vez, podem ter necessidades especiais que requerem acomodações, como salas acessíveis ou equipamentos específicos. Ao considerar essas nuances, o SARA pode criar um ambiente mais inclusivo e adaptado às necessidades individuais e alinhada com as necessidades e particularidades de cada instituição acadêmica.

Outro aspecto é a capacidade de priorizar e balancear as preferências. Em um ambiente universitário, nem sempre é possível atender a todas as preferências devido a conflitos e limitações de recursos. Portanto, o modelo é capaz de ponderar diferentes preferências e tomar decisões que maximizem o bem-estar geral. Isso pode envolver a utilização de algoritmos para determinar a melhor forma de alocar recursos e horários. Ao fazer isso, as preferências tornam o agendamento otimizado e justo para todos os envolvidos, já que esta característica poderá ser traduzida em maior satisfação entre os usuários.

Agrupamentos

Esta característica permite realizar o agrupamento de elementos conforme as necessidades específicas do agendamento. Por exemplo, pode haver a necessidade de agrupar indivíduos para compor a audiência de um evento, ou criar agrupamentos de turmas ou sub-turmas como no trabalho de Siew et al. (2022). Ela visa garantir que os eventos possam ocorrer em espaços físicos adequados, com a capacidade necessária para acomodar todos os participantes. Esta habilidade contribui para a flexibilidade do modelo, permite uma organização mais refinada e alinhada com as demandas e particularidades apresentadas.

Os agrupamentos envolvem uma compreensão das dinâmicas e interações entre diferentes grupos dentro de uma instituição. Certas turmas ou sub-turmas podem ter necessidades específicas que exigem agrupamentos especiais, como aulas práticas em laboratórios ou sessões de tutoria. Da mesma forma, eventos especiais, como *workshops* ou conferências, podem exigir a combinação de vários grupos para maximizar a participação e o engajamento.

A capacidade de manter agrupamentos de forma dinâmica permite que a instituição se adapte a mudanças ou situações imprevistas. Se um espaço físico específico não estiver disponível devido a reparos ou outras circunstâncias, o modelo pode reagrupar eventos ou turmas para acomodar essa mudança.

Eles também contribuem com a análise de dados. Ao monitorar e avaliar diferentes agrupamentos, a instituição pode identificar padrões, tendências e áreas de melhoria. Isso pode levar a *insights* sobre como otimizar a alocação de recursos e melhorar a satisfação dos participantes. Em última análise, a habilidade de manter agrupamentos contribui para a adaptabilidade do modelo, pois colabora para que ele possa atender às necessidades de uma instituição acadêmica.

Relatórios

Esta característica possibilita a disponibilização de relatórios com dados dos agendamentos já realizados, entre outros (SIDDIQUI; RAZA; TARIQ, 2018). Por meio dela, é possível exibir informações específicas de cada agendamento, oferecendo diferentes perspectivas e análises. Isso inclui, por exemplo, a visualização de agendamentos relacionados a um professor específico, uma sala, uma turma, entre outros aspectos. A capacidade de disponibilizar relatórios enriquece a análise e o acompanhamento do processo de agendamento, contribui com a transparência do processo e permite uma gestão mais informada.

Ter acesso a relatórios detalhados e precisos pode ser a diferença entre um semestre bem-sucedido e um repleto de desafios logísticos. Ao analisar relatórios de utilização de salas, a instituição pode identificar espaços subutilizados e, assim, redistribuir recursos

para maximizar o uso do espaço disponível, por exemplo. Eles também podem ser usados para identificar tendências ao longo do tempo incluindo padrões de inscrição em turmas, preferências de horários entre os professores, entre outros. Os *insights* gerados podem ser úteis para o planejamento futuro, ajudando a instituição a se antecipar às necessidades e ajustar sua estratégia de agendamento.

Outra capacidade desta característica é a de responsabilização. Em caso de disputas ou confusões sobre agendamentos, os relatórios servem como um registro imutável do que foi decidido, garantindo que todas as partes estejam cientes das decisões tomadas. Além disso, ao fornecer uma visão transparente do processo de agendamento, a instituição pode garantir que os *stakeholders*, sejam eles professores, alunos ou administradores, tenham confiança no sistema e em sua capacidade de atender às suas necessidades.

Contextos

No âmbito deste modelo, o contexto refere-se a uma parte específica da modelagem para onde será gerado um agendamento de forma independente (BABAEI; KARIMPOUR; HADIDI, 2015). Um contexto é composto por um conjunto de elementos, como usuários, espaços físicos e eventos, que serão combinados ao longo do fluxo de agendamento. Cada contexto deve ter um responsável pela sua manutenção, permitindo que seus elementos sejam organizados de forma separada. Isso possibilita a subdivisão das necessidades de agendamento de uma instituição em partes menores e independentes, promovendo flexibilidade. A capacidade manter contextos distintos facilita a adaptação do modelo às variadas demandas e complexidades das instituições acadêmicas, assegurando que o processo de agendamento seja ágil e flexível.

Em ambientes acadêmicos, onde diferentes departamentos, cursos ou programas podem ter requisitos e prioridades distintos, a capacidade de criar contextos separados é uma vantagem. Por exemplo, um departamento de ciências pode ter necessidades específicas relacionadas a laboratórios e equipamentos especializados, enquanto um departamento de artes pode priorizar estúdios ou salas de performance. Ao gerenciar esses departamentos como contextos separados, o modelo pode atender às suas necessidades individuais sem comprometer agendamentos de outros contextos.

Além disso, ao designar um responsável para cada contexto, o modelo garante que as decisões de agendamento sejam tomadas por indivíduos que têm um entendimento profundo das necessidades e prioridades de seu respectivo contexto. Isso melhora a precisão e do agendamento e promove a responsabilidade e a propriedade do processo.

A capacidade de isolar e resolver problemas ou conflitos que possam surgir de forma separada é outra vantagem provida pelo contexto. Se um problema de agendamento ocorre em um contexto específico, ele pode ser tratado sem perturbar o agendamento

em outros contextos. Isso permite que a instituição continue operando sem interrupções significativas, mesmo quando enfrenta desafios.

Propostas de Agendamento

A proposta de agendamento representa um estágio preliminar na criação do agendamento, sendo o resultado obtido após a aplicação de algoritmos de otimização específicos. Nesta fase, o agendamento ainda não é oficializado e pode ser submetido a ajustes e revisões. Deste modo, assegura-se que a proposta de agendamento esteja em perfeita conformidade com as necessidades e expectativas da instituição, sendo ainda possível refinar e aprimorar o agendamento antes de sua finalização e publicação.

Deste modo, esta característica enfatiza a natureza iterativa e colaborativa do modelo. Reconhecendo que a primeira solução gerada pode não ser a ideal, este estágio permite um espaço para reflexão, análise e *feedback*. Os *stakeholders*, sejam eles administradores, professores ou até mesmo estudantes, podem ter a oportunidade de revisar o agendamento e fornecer *insights* que podem ter sido negligenciados na primeira passagem.

Além disso, o ambiente universitário é dinâmico, com mudanças frequentes que podem afetar o agendamento. Seja uma mudança no número de alunos inscritos em uma turma, a indisponibilidade repentina de um espaço devido a reparos ou a introdução de um novo curso ou módulo, a proposta de agendamento deve ser flexível o suficiente para acomodar essas mudanças.

Por fim, esta característica também contribui com a transparência e a comunicação aberta. Ao permitir revisões e ajustes, a instituição pode demonstrar seu compromisso em criar um agendamento que não apenas atenda às suas necessidades logísticas, mas que também respeite e valorize as contribuições e preocupações dos membros da comunidade acadêmica.

Manter Agendamentos

A necessidade de alterações no agendamento após sua oficialização é algo que pode acontecer, podendo ser motivada por razões diversas como manutenção, dedetização, ou ajustes solicitados pelas partes envolvidas (AKKAN; GüLCü; KUŞ, 2022). Esta característica permite a realização das alterações necessárias no agendamento, garantindo que as necessidades de organização de eventos da instituição sejam atendidas da melhor maneira possível. Esse controle dinâmico contribui para que o modelo seja flexível e adaptável às circunstâncias imprevistas ou mudanças nas necessidades.

Esta característica também serve como uma ferramenta de *feedback* para o modelo. Ao monitorar as alterações frequentemente solicitadas, os administradores podem identificar padrões ou áreas de melhoria. Por exemplo, se uma sala específica está constantemente

sendo realocada devido a problemas de infraestrutura, isso pode sinalizar a necessidade de reparos ou atualizações naquela sala.

4.2.2 Agendamento

O agendamento universitário é uma tarefa que envolve a organização de vários eventos (como aulas, laboratórios, entre outros) em horários específicos e espaços físicos determinados (como salas de aula, laboratórios, etc.), considerando uma série de restrições que geralmente são divididas em duas categorias *hard constraints*, que nunca podem ser violadas, e as preferências *soft constraints*, que devem ser satisfeitas ao máximo (BURKE et al., 1997), o que geralmente torna este problema muito difícil de se resolver em circunstâncias reais (BURKE et al., 2007). O objetivo é encontrar um método que consiga alocar todos os eventos em horários e espaços pré-definidos, onde todas as restrições sejam atendidas (BABAEI; KARIMPOUR; HADIDI, 2015). A complexidade deste problema cresce exponencialmente com o aumento do número de variáveis, como salas de aula, horários, professores e cursos e geralmente ele se torna um problema NP-Difícil e não pode ser resolvido em tempo polinomial (ROSSI-DORIA et al., 2003), (ASSI; HALAWI; HARATY, 2018).

Deste modo, esta é parte do modelo dedicada à criação de agendamento. O núcleo deste elemento é um algoritmo de otimização que avalia os requisitos específicos da instituição, incluindo as preferências expressas pelos coordenadores. Além disso, são utilizadas informações e dados específicos, que podem ser adquiridos tanto por meio da Integração quanto fornecidos diretamente na Personalização, facilitando a composição de contextos distintos. Seu objetivo é desenvolver um agendamento que respeite todas as *hard constraints*, satisfaça ao máximo as *soft constraints*, e que esteja em harmonia com as singularidades de cada contexto, espelhando as necessidades e particularidades da instituição acadêmica, tudo isso dentro de um tempo viável e através de uma abordagem modular e flexível.

Neste sentido, o algoritmo de otimização empregado visa encontrar uma solução viável e identificar a melhor solução dentro do conjunto de possibilidades. Para fazer isso, são considerados os seguintes parâmetros:

- *Hard constraints*: referem-se às restrições obrigatórias que não podem ser violadas. Estas definem os requisitos que o agendamento deve atender obrigatoriamente. As *hard constraints* apresentadas no trabalho de McCollum et al. (2010), que são abordadas em múltiplos estudos, podem ser vistas na Tabela 4.1.

Estas restrições são, em muitos aspectos, o esqueleto do agendamento, fornecendo a estrutura básica que mantém a integridade do modelo. Sem essas restrições fir-

Tabela 4.1: *Hard constraints* discutidas em McCollum et al. (2010)

Regra	Descrição
Aulas	Todas as aulas de um curso devem ser agendadas e devem ser atribuídas a períodos distintos.
Ocupação da sala	Duas aulas não podem acontecer na mesma sala no mesmo período. Duas aulas na mesma sala no mesmo período representam uma violação.
Conflitos	As aulas de cursos do mesmo currículo ou ministradas pelo mesmo professor devem ser todas agendadas em períodos diferentes.
Disponibilidade	Se o professor do curso não estiver disponível para ministrar esse curso em um determinado período, nenhuma aula do curso poderá ser agendada nesse período.

memente estabelecidas, o risco de inconsistências e falhas no agendamento aumenta exponencialmente. Além de evitar conflitos e sobreposições, as *hard constraints* também desempenham um papel importante na manutenção da qualidade da experiência educacional, já que evitam que uma sala de aula seja superlotada proporcionando um ambiente de aprendizado adequado para os alunos. Da mesma forma, evita que um professor seja alocado em dois lugares ao mesmo tempo respeitando tempo e o compromisso do educador (BURKE et al., 1997).

Outra característica das *hard constraints* é sua capacidade de simplificar o processo de tomada de decisão. Ao definir claramente o que é e o que não é permitido, elas eliminam uma série de opções inviáveis, permitindo que os algoritmos de otimização se concentrem em soluções viáveis e de alta qualidade, além de favorecer a transparência do processo como um todo.

- *Soft constraints*: representam restrições suaves que, embora sejam desejáveis, podem ser adaptadas ou flexibilizadas conforme a necessidade, já que geralmente é impossível atender todas elas (BURKE et al., 1997). Estas restrições não são mandatórias, mas servem como diretrizes para otimizar a qualidade do agendamento. Exemplos dessas preferências foram apresentadas McCollum et al. (2010) e podem ser vistas na Tabela 4.2.

Deste modo, as *soft constraints* atuam como refinadores do processo de agendamento, introduzindo nuances e detalhes que tornam o resultado final mais alinhado com as expectativas e desejos da instituição. Enquanto as *hard constraints* estabelecem os limites rígidos do que é permitido, as *soft constraints* trabalham dentro desses limites para criar um agendamento que funcione e seja ideal em termos de conforto, conveniência e eficácia pedagógica.

Tabela 4.2: *Soft constraints* discutidas em McCollum et al. (2010)

Regra	Descrição
Mínimo de dias de trabalho	As aulas de cada curso devem ser distribuídas nos dias mínimos especificados.
Compactação curricular	As aulas pertencentes a um currículo devem ser adjacentes umas às outras (ou seja, em períodos consecutivos).
Capacidade da sala	Para cada aula, o número de alunos que frequentam o curso deve ser inferior ou igual ao número de assentos de todas as salas que hospedam suas aulas.
Estabilidade da sala	Todas as aulas de um curso devem ser ministradas na mesma sala.

A natureza adaptável das *soft constraints* permite que as instituições acadêmicas personalizem o processo de agendamento de acordo com suas prioridades e valores. Por exemplo, uma universidade que valoriza a saúde mental e o bem-estar de seus professores pode introduzir uma *soft constraint* que evite a programação de aulas muito cedo ou muito tarde para um professor específico. Da mesma forma, uma instituição que deseja promover a interdisciplinaridade pode ter uma *soft constraint* que favoreça a programação consecutiva de aulas de diferentes disciplinas no mesmo espaço, incentivando a interação entre estudantes de diferentes campos.

É importante notar que, embora as *soft constraints* sejam flexíveis por natureza, elas não são menos importantes do que as *hard constraints*. Na verdade, muitas vezes são essas restrições mais flexíveis que determinam a satisfação dos usuários finais com o agendamento. A capacidade de equilibrar eficazmente as *hard constraints* e *soft constraints* é o que define a aceitação da abordagem.

Deste modo, a habilidade de integrar e balancear tanto as *hard constraints* quanto as *soft constraints* é importante para o sucesso do algoritmo de otimização. Esta capacidade de equilíbrio capacita o Agendamento a produzir propostas que sejam alinhadas às necessidades, objetivos e preferências específicas da instituição. Esta abordagem assegura que, mesmo quando confrontado com a complexidade das demandas acadêmicas, o modelo SARA permaneça resiliente fornecendo soluções que são robustas e adaptáveis, pavimentando o caminho para uma gestão de agendamento universitário que é simultaneamente coesa e alinhada com os ideais e metas institucionais.

O algoritmo utilizado é baseado em técnicas de otimização, que podem incluir métodos heurísticos, meta-heurísticos ou até mesmo aprendizado de máquina. Estas técnicas são escolhidas por sua capacidade de explorar e avaliar rapidamente o espaço de soluções, identificando soluções de alta qualidade em um tempo computacionalmente razoável.

À medida que novas restrições ou objetivos são introduzidos, o algoritmo pode ser ajustado ou reconfigurado para considerá-los. Isso garante que o modelo permaneça relevante, mesmo diante de mudanças nas necessidades ou prioridades da instituição.

Em termos práticos, o algoritmo funciona em várias etapas:

- Coleta de Dados: Reúne todas as informações necessárias, como disponibilidade de salas de aula, horários dos professores, inscrições de estudantes e preferências.
- Definição de Restrições e Objetivos: Estabelece as *hard constraints* e *soft constraints*, bem como os objetivos de otimização, como maximizar a satisfação das preferências ou minimizar os conflitos de horário.
- Exploração do Espaço de Soluções: Utiliza técnicas para explorar o espaço de soluções possíveis, identificando aquelas que atendem às restrições e se aproximam dos objetivos definidos.
- Avaliação e Seleção: Avalia as soluções identificadas com base em critérios predefinidos e seleciona a solução mais ótima.
- Iteração e Refinamento: Se necessário, o algoritmo pode iterar e refinar a solução, fazendo ajustes com base no *feedback* ou em novos dados.

4.2.3 *Feedback* e avaliação

Feedback e Avaliação é a parte do modelo que permite aprimorar continuamente o modelo, permitindo que todas as partes interessadas — incluindo alunos, professores e administradores — forneçam suas percepções e experiências em relação ao agendamento. Este elemento coleta *feedback* de forma estruturada por meio de questionários online e avaliações de satisfação. Além de avaliações quantitativas, como classificações em escala, também permite a coleta de comentários qualitativos para fornecer um contexto mais rico. O *feedback* poderá ser agregado para análise, onde os dados coletados poderão ser submetidos a técnicas de análise de dados para identificar padrões, tendências e espaços para melhorias. Esta parte é complementada por um conjunto de métricas predefinidas que auxiliam na avaliação do agendamento em termos de utilização de recursos, satisfação do usuário e conformidade com as políticas institucionais. Ele inclui a característica de alerta para identificar e tratar prontamente quaisquer problemas críticos que recebam *feedback* negativo. Os *insights* gerados a partir deste processo serão então usados para informar atualizações periódicas do sistema, permitindo ajustes e otimizações que refletem as necessidades e preferências dos usuários. Além disso, relatórios de avaliação serão gerados e disponibilizados para os principais tomadores de decisão dentro da instituição.

4.2.4 Comunicação e Notificação

A Comunicação e Notificação é uma parte integral deste modelo, atuando como um canal de comunicação direta entre ele e seus usuários, semelhante ao feito por Houhamdi et al. (2019). Este elemento visa manter todas as partes interessadas informadas sobre aspectos do agendamento, desde a publicação inicial dos horários até quaisquer mudanças ou cancelamentos subsequentes. Ele tem como característica a possibilidade de gerar notificações automáticas. Sempre que uma mudança significativa é feita no agendamento, como a realocação de uma sala de aula ou a mudança de horário de uma aula, notificações automáticas são enviadas aos afetados.

Além de notificações sobre mudanças no agendamento, esta parte poderá enviar alertas sobre prazos importantes. Por exemplo, coordenadores e gestores podem receber notificações lembrando-os dos prazos para a submissão de preferências de horário ou para a alocação de recursos. Da mesma forma, uma vez que o Agendamento é concluído e os horários são finalizados, o modelo pode enviar uma notificação em massa para anunciar a publicação dos resultados, permitindo que os alunos e professores ajustem seus planos de acordo. Estas notificações programadas ajudam a manter o processo em andamento, assegurando que todas as atividades relacionadas ao agendamento sejam realizadas de forma oportuna.

Esta parte do modelo facilita a disseminação de informações, mas também poderá auxiliar orquestração de tarefas, lembrando os usuários de suas responsabilidades dentro do processo de agendamento. Ao fazer isso, ele contribui para a eficiência do modelo, colaborando para que todos os envolvidos estejam cientes de suas agendas, prazos e quaisquer mudanças que possam afetá-los.

4.2.5 Predição

A Predição introduz uma inteligência que vai além da alocação de recursos. Utilizando estratégias inteligentes, este elemento recupera situações históricas similares, incluindo padrões de utilização de salas, preferências de horários de professores e alunos, e até mesmo variáveis sazonais como feriados acadêmicos ou períodos de exames, para resolver o problema em questão, de maneira parecida a apresentada por Burke, Petrovic e Qu (2006). A partir disso, o modelo pode prever necessidades futuras de agendamento, identificando potenciais gargalos, otimizando a utilização de recursos e até mesmo antecipando mudanças nas preferências dos usuários. Por exemplo, poderia prever um aumento na demanda por laboratórios de ciências durante um semestre específico ou antecipar a necessidade de salas adicionais devido a um aumento nas matrículas.

Este elemento também oferece a possibilidade de realização de simulações que permitirão aos administradores testar diferentes cenários de agendamento antes de sua implementação. Isso pode ser útil para se avaliar o impacto de mudanças, como a introdução de novos cursos, a reestruturação de departamentos acadêmicos ou a construção de novas instalações. Os *insights* obtidos a partir dos resultados dessas simulações podem ser usados para refinar ainda mais as estratégias de agendamento, permitindo que elas sejam resilientes a variações e incertezas.

Deste modo, a Predição serve para a realização de previsões e simulações, podendo ser utilizado para auxílio no planejamento estratégico, fornecendo aos tomadores de decisão as condições necessárias para que sejam criados melhores agendamentos.

4.2.6 Integração

A Integração atua como um elemento para assegurar que o modelo possa ser compatível e sinérgico com estratégias de gestão acadêmica adotadas (EKE et al., 2019). Considerando a rede de operações que caracterizam o ambiente universitário — que frequentemente envolve mecanismos distintos para matrículas, gestão de pessoal, controle de ativos e planejamento acadêmico — este elemento foi criado para facilitar a interoperabilidade entre modelo e a instituição.

Ela possui a capacidade de integração em tempo real, permitindo seja feitas atualizações em outros elementos conectados. Isso é importante para decisões e operações sensíveis ao tempo, como agendamento de aulas ou alocação de recursos. Este elemento é adaptável, podendo ser modificado para atender às necessidades específicas da instituição.

Deste modo, uma vez que o agendamento esteja concluído outros mecanismos utilizados na gestão poderão ser atualizados. Isso permite que alunos, professores e outros *stakeholders* sejam informados sobre seus respectivos horários, locais de aula e quaisquer outras informações pertinentes. A flexibilidade e a abrangência deste elemento de integração garantem que o modelo possa ser uma adição ao ecossistema de gestão da instituição acadêmica.

As seguintes características compõe integração:

Órgão

Órgão designa as entidades administrativas e acadêmicas que compõem a hierarquia institucional, desempenhando papéis específicos na governança e operação da universidade. Estes órgãos podem variar em escopo e responsabilidade, desde institutos, que geralmente abrangem várias disciplinas ou áreas de estudo relacionadas, faculdades, que podem ser

especializadas em campos específicos do conhecimento, até departamentos, que se concentram em subáreas ou disciplinas específicas dentro de um instituto ou faculdade.

Campus

Um *Campus* é definido como uma extensão territorial onde uma instituição acadêmica estabelece suas instalações, infraestruturas e serviços para a realização de atividades educacionais, de pesquisa e extensão. Ele pode abrigar diversos prédios, laboratórios, salas de aula, áreas de convivência e outras facilidades que atendem às necessidades de alunos, professores e demais membros da comunidade acadêmica. Cada campus possui características únicas, refletindo a missão, visão e objetivos da instituição em um contexto geográfico específico

Projeção

Uma Projeção é entendida como uma edificação ou estrutura que agrupa diversos espaços físicos, como salas de aula, laboratórios, auditórios e demais ambientes destinados a atividades acadêmicas e administrativas. Uma projeção pode ser um prédio, um bloco ou qualquer outra construção que organize e centralize funções específicas dentro do campus. Cada projeção tem sua singularidade, refletindo a natureza das atividades que abriga e as necessidades dos usuários que a frequentam.

Espaço Físico

Um espaço físico é definido como uma unidade individualizada dentro de uma projeção, destinada a acomodar atividades específicas, sejam elas acadêmicas, administrativas ou de suporte. Exemplos de espaços físicos incluem salas de aula, laboratórios, escritórios, auditórios, entre outros. Cada espaço físico possui características próprias, como capacidade de ocupação, recursos disponíveis e finalidade de uso. Estas características são essenciais para determinar sua adequação a diferentes atividades e para otimizar o processo de agendamento.

Tipo de Espaço Físico

Refere-se à categorização dos espaços com base em sua finalidade e características predominantes. Estes tipos ajudam a identificar rapidamente a natureza e a utilidade de um espaço, facilitando o processo de agendamento e alocação. Exemplos comuns de tipos de espaço físico incluem 'Sala de Aula', destinada principalmente para aulas teóricas; 'Laboratório', para atividades práticas ou experimentais; e 'Sala de Informática', equipada com computadores e destinada a atividades digitais.

Horário

Horário é conceituado como a especificação temporal que determina quando determinadas atividades acadêmicas, como aulas ou eventos, ocorrerão. Cada horário pode ser estruturado considerando fatores como o órgão responsável (por exemplo, Departamento ou Faculdade), o nível acadêmico (Graduação, Pós-graduação, etc.) e segue uma sequência lógica e estruturada para garantir a fluidez das atividades diárias. A organização e definição precisa dos horários são importantes para evitar sobreposições e garantir a melhor utilização dos recursos da instituição.

Evento

Evento é conceituado como qualquer atividade ou encontro programado que ocorra dentro da instituição, podendo ser as aulas de uma turma, palestra, seminário, *workshop* ou qualquer outra ocasião que demande a utilização de um espaço físico durante um período específico. Cada evento tem características distintas, como duração, público-alvo, recursos necessários e localização. A correta catalogação e gerenciamento dos eventos são importantes para evitar sobreposições ou conflitos de agendamento.

4.2.7 Relatórios

Relatórios servem como uma estratégia analítica dentro do SARA, fornecendo *insights* sobre a abordagem como um todo. Este elemento vai além da geração de relatórios tabulares ou gráficos básicos, podendo ser utilizado para realizar análises que auxiliem na tomada de decisões. Por meio dele podem ser feitas análises sobre a utilização de recursos, gerando entendimento sobre quais salas de aula ou laboratórios estarão sendo subutilizados e quais estarão em alta demanda. Isso pode ser útil para o planejamento de recursos a longo prazo. Além disso, é possível fornecer análises sobre padrões de agendamento, como a frequência de conflitos de horário para alunos ou professores, permitindo que a instituição faça ajustes para melhorar a satisfação dos usuários.

Outra característica deste elemento é a capacidade de personalizar visões de acordo com as necessidades específicas dos usuários. Administradores podem querer relatórios que focam características operacionais, enquanto coordenadores acadêmicos podem estar mais interessados em como o agendamento afeta o desempenho acadêmico ou a satisfação dos alunos. Os Relatórios podem ser configurados para gerar visões que atendam a essas necessidades variadas, oferecendo filtros e parâmetros que os usuários podem ajustar para obter as percepções que realmente importam para eles.

4.3 Usuários

A segmentação dos usuários em diferentes categorias é fundamental para o modelo, devido a quantidade de pessoas impactadas (BURKE et al., 1997). Cada categoria tem responsabilidades e funções específicas, conforme descrito a seguir:

- **Coordenadores:** Esta categoria é indicada aos indivíduos que representam uma unidade acadêmica, uma vez que compete a eles gerenciar as atividades do programa e representá-lo. Essas pessoas geralmente participam de atividades como a criação de turmas que serão ofertadas, definição do número de vagas que serão abertas, além da indicação da localização onde se deseja que as aulas sejam realizadas.
- **Gestores:** Esta categoria é indicada aos indivíduos responsáveis pela gestão dos espaços físicos da instituição. Sua função principal inclui o controle dos espaços disponíveis para uso, especialmente aqueles que são compartilhados entre as unidades acadêmicas. Eles organizam a logística de agendamento, garantindo que múltiplos cursos possam ocorrer de forma paralela e sem conflitos. Geralmente, essas pessoas são as responsáveis por criar e manter o agendamento das aulas, coordenando a utilização dos espaços da instituição por diferentes unidades acadêmicas.
- **Administradores:** Esta categoria é destinada aos usuários com responsabilidades administrativas abrangentes, sendo responsáveis pelas atividades de incorporação de informações específicas e gestão de usuários. Sua função pode incluir suporte na criação de agendamentos, já que possuem a capacidade de realizar todas as atividades atribuídas aos outros usuários. Além disso, a categoria de Administradores supervisiona e controla as operações, assegurando que o processo de criação de agendamento funcione de maneira eficaz e esteja em conformidade com os requisitos e padrões estabelecidos pela instituição.

Capítulo 5

A implementação do SARA

Neste capítulo, apresentamos a implementação do Sistema de Agendamento de Recursos Acadêmicos (SARA), detalhando sua estrutura e funcionalidades. Inicialmente, fornecemos uma breve introdução ao sistema, explicando seu propósito e contexto de uso na organização de recursos educacionais. Em seguida, discutimos suas principais características, incluindo as restrições (*constraints*) consideradas e o algoritmo utilizado para otimizar o agendamento. Também abordamos os diferentes perfis de usuários e como cada um interage com o sistema.

Posteriormente, exploramos os detalhes de implementação do SARA, descrevendo as tecnologias empregadas e a arquitetura adotada para garantir uma solução modular e escalável. Discutimos as decisões arquitetônicas e a integração dos componentes, além de explicar o fluxo de dados, desde a extração até a entrada no sistema.

5.1 Introduzindo o SARA

A elaboração de uma grade horária de aulas, que satisfaça professores, alunos, e otimize os recursos institucionais, apresenta diversas possibilidades de exploração, sendo necessário que haja o equilíbrio de fatores para que se atendam ao máximo às necessidades de todas as partes envolvidas, respeitando as restrições impostas.

Os professores têm interesse na criação da grade horária das aulas, pois afetam diretamente o planejamento da sua jornada de trabalho e podem impactar sua vida pessoal. Cada docente possui horários específicos para lecionar e preferências pessoais que devem ser respeitadas sempre que possível, contribuindo assim, para melhor satisfação.

A otimização da utilização da infraestrutura é outro ponto relevante. Elementos como o respeito à capacidade das salas de aula, disponibilidade de equipamentos, gestão de laboratórios e minimização de mudanças de sala ao longo do dia contribuem para diminuir conflitos e melhorar a gestão dos recursos.

Além disso, considerar a flexibilidade para ajustes de última hora contribui para manter um cronograma funcional ao longo do semestre. Esta flexibilidade pode ser viabilizada por meio de algoritmos que permitem uma rápida reconfiguração dos horários e alocações de recursos, minimizando o impacto de imprevistos.

Diante do exposto, para testar o modelo apresentado anteriormente, alguns dos seus elementos foram implementados e podem ser visualizados na Figura 5.1. A implementação teve como objetivo auxiliar na criação de grades horárias, alocando eventos acadêmicos em instituições de ensino superior. Utilizando algoritmos de otimização, o SARA oferece flexibilidade para ajustes manuais na grade horária gerada e incorpora um mecanismo de crítica algorítmica em tempo real, o que auxilia na prevenção de conflitos decorrentes das modificações realizadas no agendamento.

O fluxo básico de uso inicia-se com a importação de arquivos que contêm as informações necessárias para a criação da grade horária e a alocação em espaços físicos. Em seguida, o algoritmo de otimização é acionado, gerando o agendamento preliminar das aulas com base nas restrições definidas. Esse agendamento é disponibilizado para que um gestor possa verificar, editar e realizar ajustes necessários antes da publicação final na instituição.

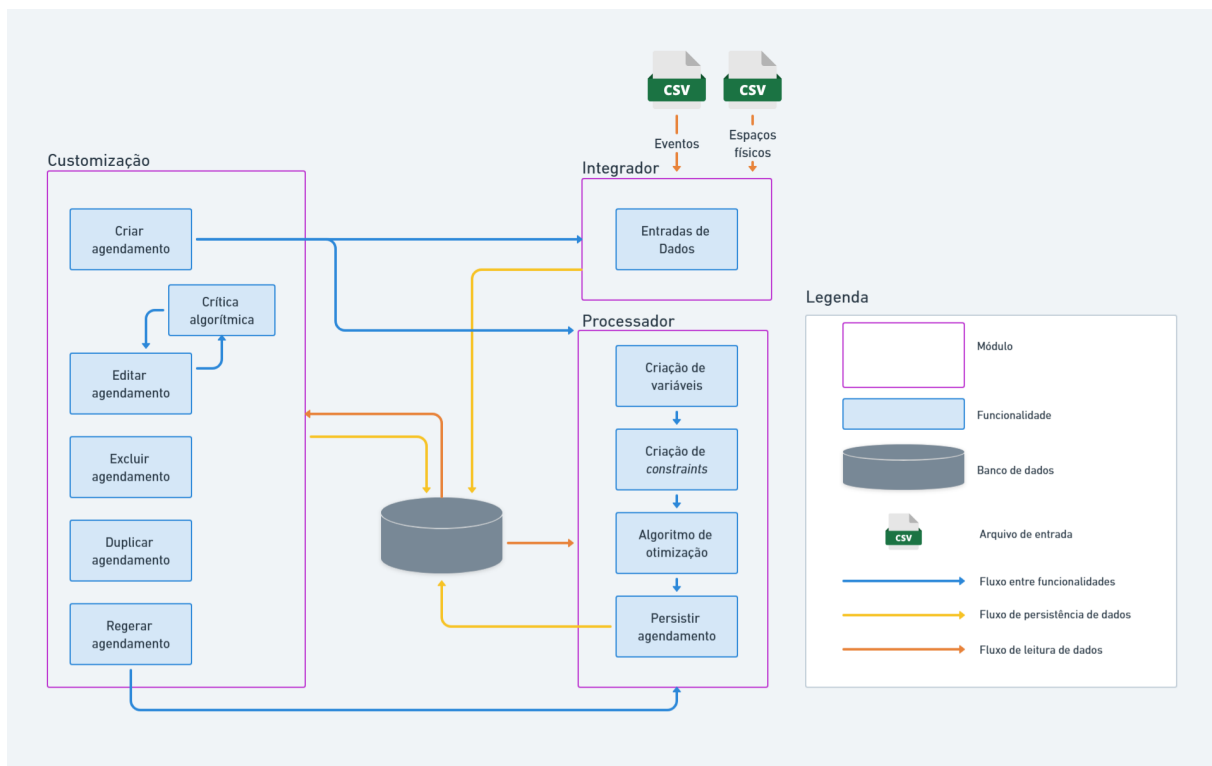


Figura 5.1: Estrutura do SARA

5.2 Características

Nesta seção, são exploradas as principais características do SARA. Entre elas, destacam-se os arquivos de entrada, que contêm as informações de eventos e espaços físicos utilizados para a criação dos agendamentos; as *constraints*, classificadas em *hard* e *soft*, onde são apresentadas as formulações das restrições implementadas; o algoritmo utilizado tanto para a criação quanto para a otimização dos agendamentos, incluindo uma demonstração dos principais parâmetros configuráveis; a crítica algorítmica, que valida em tempo real as alterações feitas nos agendamentos; os perfis de usuários, que segregam características e funções conforme a utilização do sistema; as ações disponíveis para os usuários; e as tecnologias empregadas na criação do SARA.

5.2.1 Arquivos de Entrada de dados

Os arquivos de entrada do SARA podem ser gerados manualmente ou por alguma ferramenta automatizada. O módulo Integrador permite a importação de dados por meio de arquivos no formato *Comma Separated Value* (CSV), que são comumente utilizados para armazenar informações tabulares. Este módulo inclui funções para importar informações sobre os eventos acadêmicos e os espaços físicos de forma independente das aplicações de origem, bastando para isto que os arquivos de estejam nos formatos descritos abaixo:

Arquivo de espaços físicos

O arquivo de espaços físicos contém dados sobre salas de aula e laboratórios, por exemplo. Esse arquivo deve incluir as informações específicas apresentadas na Tabela 5.1. No nosso caso, os dados para a criação deste arquivo foram extraídos do banco de dados do Sistema Unificado de Tabelas (SITAB), onde são gerenciadas as informações sobre local e projeção na UnB. Um exemplo de SQL que pode ser utilizado para extrair os dados do banco de dados do SITAB para a criação deste arquivo está disponível no Apêndice B.

Arquivo de eventos

O arquivo de eventos contém dados sobre as turmas, professores e preferências. Este arquivo deve incluir as informações específicas apresentadas na Tabela 5.2. Neste caso, os dados para a criação deste arquivos foram extraídos do banco de dados do SIGAA. Um exemplo de SQL que pode ser utilizado para extrair os dados do banco de dados do SIGAA para a criação deste arquivo está disponível no Apêndice C.

A capacidade de importar e integrar dados facilita para que o SARA possa ser uma adição ao ecossistema de gestão acadêmica das instituições. Sua arquitetura desacoplada

Campo	Descrição
LocCodigo	Identificador do espaço físico.
LocSigla	Sigla que identifica o espaço físico. Ex. A1-09/20.
LocDenominacao	Descrição do espaço físico. Ex. Sala Multiuso.
LocCapacidadePessoas	Capacidade de pessoas que o espaço suporta.
LocOInCodigoRespLocal	Código do órgão responsável pelo local.
LocOInCodigoRespPatrimonio	Código do órgão responsável pelo patrimônio.
LocProCodigo	Código da projeção do local.
LocTLoCodigo	Código do tipo de local.
LocLGeCodigo	Código da localização geográfica.

Tabela 5.1: Descrição dos campos do arquivo de espaços físicos

Campo	Descrição
unidade_nome	Nome da unidade acadêmica da turma.
curso_nome	Nome do curso.
curso_turno	Identificador do turno.
curso_id	Identificador do curso.
turma_id	Identificador da turma.
docente_nome	Nome do docente.
docente_id	Identificador do docente.
codigo_componente	Código do componente curricular. Ex. FGA0163.
turma_codigo	Código da turma. Ex. 01.
turma_capacidade_aluno	Quantidade de vagas ofertadas.
turma_horario_historico	Horário histórico em que a turma geralmente é agendada. Ex. 35T6 35N1.
id_campus_turma	Identificador do Campus onde a turma é lecionada.
turma_ano	Ano da turma.
turma_periodo	Período da turma.
turma_local	Local onde a turma foi agendada pelo processo tradicional de alocação. Utilizado apenas para referência.
componente_nome	Nome da disciplina - Uso futuro.
grau_academico_descricao	Grau acadêmico do curso.

Tabela 5.2: Descrição dos Campos dos Eventos Acadêmicos

promove flexibilidade na gestão de recursos acadêmicos, permitindo que as instituições personalizem os arquivos de importação conforme suas necessidades específicas. A abordagem baseada em arquivos CSV também reduz a complexidade e os custos associados à integração de sistemas.

5.2.2 *Constraints*

Visando garantir que os agendamentos gerados pelo SARA apresentem uma qualidade aceitável em ambientes complexos, foram implementadas globalmente as *constraints* apre-

sentadas a seguir. Essas *constraints* asseguram que o SARA possa lidar com uma variedade de situações e requisitos específicos.

Considerando que:

- $x_{i,s,p}$, é a variável de decisão, onde o valor 1 indica que o evento i foi agendado na sala s durante o horário p ; caso contrário, o valor atribuído é 0.
- $i = \{1, 2, 3, \dots, I\}$ é a sequência dos números dos eventos.
- $s = \{1, 2, 3, \dots, S\}$ é a sequência dos números dos espaços físicos disponíveis.
- $p = \{1, 2, 3, \dots, P\}$ é a sequência dos números horários que foram considerados, sendo 15 por dia durante 6 dias, totalizando 90 horários.
- N_{salas} é o número total de salas disponíveis.
- $N_{\text{horários}}$ é o número total de horários disponíveis.
- N_{eventos} é o número total de eventos.
- $N_{\text{eventos}}(i)$ é o número total de vezes que o evento i precisa ser agendado.
- $N_{\text{alunos}}(i)$ é o número de alunos inscritos no evento i .
- C_s é o número máximo de alunos que a sala s pode acomodar.

Hard constraints

HC1 - Todas as aulas são alocadas e em períodos distintos. Esta restrição garante que todas as aulas sejam programadas e que nenhuma aula deixe de ser agendada. Além disso, garante que cada aula ocorra em um período distinto, evitando sobreposições. Na nossa modelagem ela foi realizada pela combinação de duas restrições, conforme apresentadas abaixo:

$$\sum_{s=1}^{N_{\text{salas}}} \sum_{p=1}^{N_{\text{horários}}} x_{i,s,p} = N_{\text{eventos}}(i) \quad \forall i \in I$$

A primeira parte da restrição garante que o evento i seja agendado exatamente $N_{\text{eventos}}(i)$ vezes. Isso significa que todas as instâncias do evento i devem ser agendadas, cobrindo todos os horários e salas disponíveis, de modo que o número total de vezes que o evento ocorre seja igual ao necessário.

$$\sum_{s=1}^{N_{\text{salas}}} x_{i,s,p} \leq 1 \quad \forall i \in I, \forall p \in P$$

Enquanto a segunda parte garante que o evento i não seja agendado em mais de uma sala ao mesmo tempo durante qualquer horário p . Em outras palavras, um evento não pode ocorrer simultaneamente em múltiplas salas durante o mesmo período, assegurando que cada instância do evento seja única em um determinado horário.

HC2 - Duas aulas não podem acontecer na mesma sala no mesmo horário.

Para evitar conflitos de espaço, esta restrição garante que nenhuma sala seja atribuída a mais de uma aula simultaneamente. Isso assegura que cada aula tenha um espaço dedicado e evita conflitos de agendamento.

$$\sum_{i=1}^{N_{\text{eventos}}} x_{i,s,p} \leq 1 \quad \forall s \in S, \forall p \in P$$

HC3 - A capacidade da sala deve ser adequada para o número de alunos. Esta restrição verifica que a capacidade da sala é suficiente para acomodar todos os alunos inscritos na turma. Isso evita a superlotação e garante um ambiente de aprendizado confortável e seguro.

$$x_{i,s,p} \cdot N_{\text{alunos}}(i) \leq C_s \quad \forall i \in I, \forall s \in S, \forall p \in P$$

HC4 - Se o professor não estiver disponível para ministrar em um determinado horário, o evento não poderá ser agendado nele. Esta restrição garante que os eventos não sejam programados em horários nos quais os professores estejam indisponíveis. Ela respeita as preferências e limitações de horários dos professores, evitando conflitos pessoais e profissionais.

$$\forall i \in I, \forall s \in S, \forall p \in P, \quad \text{se } p \in K_k \text{ então } x_{i,s,p} = 0$$

onde K_k é o conjunto de horários que o docente não está disponível.

HC5 - Eventos ministrados pelo mesmo professor devem ser todos agendados em horários diferentes. Esta restrição assegura que um professor não tenha eventos agendados em horários sobrepostos.

$$\sum_{i \in I} \sum_{s \in S} x_{i,s,p} \cdot \delta(\text{prof}, \text{professores}(i)) \leq 1 \quad \forall \text{prof}, \forall p$$

onde $\text{professores}(i)$ é o conjunto de professores que ensinam o evento i e $\delta(\text{prof}, \text{professores}(i))$ é uma função indicadora que retorna 1 se prof estiver entre os professores do evento i , e 0 caso contrário.

Esta restrição garante que um professor não esteja agendado para mais de um evento durante o mesmo horário p . A função indicadora $\delta(\text{prof}, \text{professores}(i))$ verifica se o professor prof está associado ao evento i , e a soma ponderada assegura que o professor não seja designado a múltiplos eventos simultaneamente.

Soft constraints

SC1 - Tenta alocar o máximo de turmas respeitando o horário histórico. Esta restrição tenta preservar os horários históricos dos eventos, mantendo a consistência nos horários ao longo do tempo. Esta mesma restrição pode ser utilizada para verificar as preferências de horário dos professores, bastando para isto sobrepor a informação de horário histórico do evento.

Para cada evento i com horários históricos, definimos:

defina $\text{preferencia_satisfeita}_i \in \{0, 1\}$

Se os horários históricos forem preferidos:

$$\sum_{s=1}^{N_{\text{salas}}} \sum_{p \in P_{\text{pref}}(i)} x_{i,s,p} = \text{total_eventos_preferidos}(i) \Rightarrow \text{preferencia_satisfeita}_i = 1$$

$$\sum_{s=1}^{N_{\text{salas}}} \sum_{p \in P_{\text{pref}}(i)} x_{i,s,p} < \text{total_eventos_preferidos}(i) \Rightarrow \text{preferencia_satisfeita}_i = 0$$

Aplicamos uma penalidade se a preferência não for satisfeita:

$$\text{penalidade_nao_atender_preferencia} += 1 \times (1 - \text{preferencia_satisfeita}_i)$$

onde:

- $P_{\text{pref}}(i)$ é o conjunto de horários históricos para o evento i .
- $\text{total_eventos_preferidos}(i)$ é o número total de vezes que um evento i deve ser agendado.
- $\text{penalidade_nao_atender_preferencia}$ é o valor da penalidade aplicada para cada horário histórico que não é totalmente satisfeito. É a soma do número de períodos em que um período histórico não foi atendido.

SC2 - Tenta alocar o máximo de ocorrências do evento sempre na mesma sala.

Esta restrição procura minimizar as mudanças de sala para os eventos, permitindo que as suas ocorrências sejam na mesma sala sempre que possível. Isso aumenta o conforto e a familiaridade dos alunos e professores com o ambiente de aula.

Para cada evento i e para cada sala s , definimos:

$$\text{sala_usada}_{i,s} \in \{0, 1\}$$

$$\text{sala_usada}_{i,s} = \max(\{x_{i,s,p} : p \in P\})$$

O número total de salas usadas pelo evento i é:

$$\text{salas_usadas_por_evento}_i = \sum_{s=1}^{N_{\text{salas}}} \text{sala_usada}_{i,s}$$

onde:

- $\text{penalidade_sala_diferente}$ é o valor da penalidade aplicada para cada sala extra usada.

A penalidade por usar mais de uma sala é calculada como:

$$\text{salas_excedentes}_i = \text{salas_usadas_por_evento}_i - 1$$

A função objetivo é calculada como:

$$\text{total_penalidades} = \text{penalidade_nao_atender_preferencia} +$$

$$\text{penalidade_sala_diferente} \cdot \text{salas_excedentes}_i$$

Minimizar (total_penalidades)

Neste sentido, após a execução das *hard constraints*, que garantem a construção do agendamento, são aplicadas as *soft constraints* para realizar a otimização. Esta abordagem foi implementada em um único passo, evitando a necessidade de combinar diferentes algoritmos para a construção e a otimização do agendamento.

5.2.3 Algoritmo adotado

O CP-SAT, que é uma ferramenta de programação por restrições, foi utilizado na modelagem e resolução do problema que estamos abordando. Foi por meio dele que todas as restrições foram modeladas. A seguir, detalho como o CP-SAT foi integrado ao projeto e quais foram suas contribuições específicas.

No núcleo do SARA, o algoritmo foi configurado para otimizar o uso dos recursos da instituição, visando minimizar a não satisfação de preferências de horário ou horários históricos e alocar as ocorrências de eventos sempre no mesmo espaço físico. O CP-SAT foi configurado com várias opções para melhorar a eficiência e a performance do processo de resolução. A Tabela 5.3 sintetiza os parâmetros utilizados no sistema na fase de testes.

O CP-SAT permitiu a modelagem precisa de várias restrições essenciais para o agendamento: não sobreposição de aulas ministradas pelo mesmo professor em horários diferentes, consideração das indisponibilidades dos professores, garantindo que o número de

Parâmetro	Descrição
<code>use_phase_saving = True</code>	Ativa a técnica de <i>phase saving</i> , que ajuda a reduzir o número de conflitos ao lembrar a última atribuição de valores para as variáveis durante a busca. Isso pode acelerar a convergência para uma solução.
<code>clause_cleanup_period = 5000</code>	Define a frequência com que o solucionador deve limpar as cláusulas de restrição obsoletas ou inativas. Esse valor de 5000 indica que a limpeza ocorrerá a cada 5000 conflitos, ajudando a manter o modelo enxuto.
<code>test_feasibility_jump = True</code>	Permite que o solucionador faça <i>jumps</i> para testar rapidamente a viabilidade de soluções parciais durante a busca. Isso pode ajudar a encontrar soluções viáveis mais rapidamente ao pular partes da árvore de busca que são menos promissoras.
<code>max_time_in_seconds = 1800</code>	Define o tempo máximo de execução do solucionador em segundos. Neste caso, o solucionador tentará encontrar uma solução em até 1800 segundos (30 minutos).
<code>num_search_workers = 16</code>	Especifica o número de <i>threads</i> de busca paralela a serem utilizadas. Utilizar 16 threads pode acelerar a resolução do problema ao permitir buscas simultâneas em diferentes partes do espaço de solução.
<code>cp_model_presolve = True</code>	Ativa a pré-solução do modelo, onde o solucionador simplifica o modelo antes de iniciar a busca completa. Isso pode reduzir significativamente o tempo de solução ao eliminar variáveis e restrições redundantes.
<code>symmetry_level = 3</code>	Define o nível de tratamento de simetria a ser utilizado. Um nível mais alto (como 3) pode ajudar a eliminar simetrias no problema, reduzindo o espaço de busca e acelerando a resolução.

Tabela 5.3: Descrição dos Parâmetros do Solver CP-SAT utilizados no SARA.

alunos não excedesse a capacidade das salas disponíveis, e atendimento às preferências de horário dos professores e alunos, minimizando penalidades para horários não preferidos.

O processo de solução e otimização envolveu a criação de variáveis booleanas para cada possível alocação de aula, sala e período. A função objetivo foi definida para minimizar penalidades associadas a horários não preferidos e ao uso de múltiplas salas para a mesma turma. O CP-SAT foi capaz de encontrar soluções ótimas em muitos casos, equilibrando todas as restrições e preferências impostas.

O CP-SAT trouxe contribuições para a implementação do SARA. Primeiramente, sua eficiência na resolução de problemas complexos permitiu a consideração simultânea de múltiplas restrições e objetivos. Além disso, sua flexibilidade na modelagem de restrições possibilitou uma definição mais específica de regras institucionais.

A utilização do CP-SAT também resultou em uma otimização dos recursos disponíveis, como salas e horários, minimizando o tempo ocioso e os conflitos. Ao respeitar as preferências de horários históricos e minimizar mudanças indesejadas, o CP-SAT contribuiu para a satisfação de professores.

5.2.4 Crítica algorítmica

O SARA utiliza um algoritmo de crítica em tempo real que atua como um assistente para os usuários. Esse sistema de críticas realiza verificações automáticas e contínuas sobre as alterações realizadas no agendamento, informando imediatamente se a mudança gerou um conflito com outros eventos ou se é considerada válida. Esse *feedback* imediato permite uma análise detalhada das implicações de cada modificação, auxiliando os responsáveis pelo agendamento a tomar decisões mais informadas.

A cada movimento realizado, o algoritmo verifica automaticamente a disponibilidade de salas, horários e professores envolvidos, além de avaliar a sobreposição de recursos. Caso uma alteração introduza um conflito — como a duplicidade de um docente em diferentes eventos no mesmo horário ou a alocação de um espaço já reservado — o sistema atribui um estado de alerta ao evento. Esses estados de alerta são categorizados conforme apresentado na Tabela 5.4, facilitando a visualização dos problemas encontrados e oferecendo recomendações para solucioná-los.

A complexidade de um agendamento acadêmico, que pode envolver dezenas ou até centenas de eventos interdependentes, torna a identificação manual de conflitos uma tarefa extremamente trabalhosa e sujeita a erros. O uso de verificações algorítmicas em tempo real permite detectar possíveis incompatibilidades imediatamente, reduzindo a probabilidade de falhas na organização final. Além disso, essa funcionalidade permite simulações de alterações com validações antes de qualquer efetivação no sistema. Dessa forma, os

Estado	Descrição
Original	Representa a posição original onde o evento foi alocado na criação do agendamento. Nesta situação os eventos são representados na cor azul.
Validado	Após a alteração do evento, quando a crítica algorítmica não encontra conflitos então ele vai para o estado validado, sendo representado na cor verde.
Invalidado	Este estado ocorre quando o evento que foi arrastado e soltado causou algum tipo de conflito na célula. Neste caso ele é representado na cor vermelha.
Conflitante	Os clicar em um evento que está no estado inválido os eventos que estão conflitantes com ele são marcados com uma borda vermelha. Deste modo, é possível ver quais eventos estão em conflito.
Clicado	Ao clicar sobre um evento todas as suas ocorrências no agendamento são apresentadas na cor amarela. Assim é possível ver todas as suas ocorrências no calendário de forma simplificada.

Tabela 5.4: Estados do evento

usuários podem experimentar diferentes configurações de horários e locais sem o risco de comprometer a integridade do agendamento original.

5.2.5 Usuários

Para garantir a segurança e a integridade dos dados, foram implementados mecanismos de login e segregação de funções. Esses mecanismos asseguram que apenas usuários autorizados possam acessar e modificar informações sensíveis. A segmentação dos usuários em diferentes categorias facilita o processo de criação de agendamento, onde cada categoria tem responsabilidades que combinadas compõe os perfis descritos a seguir:

- **Comum:** Esta categoria é destinada aos indivíduos que são impactados pelo agendamento, mas não são responsáveis por sua criação. Por exemplo, um professor que deseja expressar suas preferências de horário ou informar os horários em que estará indisponível para lecionar se encaixa bem nessa categoria. Esses indivíduos podem fornecer informações que ajudam na construção de um agendamento mais eficiente e adequado às necessidades de todos os envolvidos, mas não têm a responsabilidade de definir o cronograma final.
- **Gestores:** Esta categoria é destinada aos profissionais responsáveis pela gestão dos recursos acadêmicos da instituição. Suas funções incluem o controle dos espaços disponíveis para uso, especialmente aqueles compartilhados entre as unidades acadêmicas. Esses gestores organizam a logística do agendamento, garantindo que múltiplos cursos ocorram simultaneamente, sem conflitos de horários ou recursos.

Geralmente, os gestores são responsáveis pela criação e manutenção do calendário de aulas, coordenando a utilização dos espaços institucionais pelas diferentes unidades acadêmicas. Além disso, eles representam suas respectivas unidades, gerenciando as atividades dos programas e participando ativamente de decisões importantes, como a criação de turmas a serem ofertadas, a definição do número de vagas disponíveis e a escolha das localizações onde as aulas serão realizadas.

- **Administradores:** Esta categoria é destinada aos usuários com responsabilidades administrativas abrangentes, sendo responsáveis pelas atividades de incorporação de informações específicas e gestão de usuários. Sua função pode incluir suporte na criação de agendamentos, já que possuem a capacidade de realizar todas as atividades atribuídas aos outros usuários. Além disso, a categoria de Administradores supervisiona e controla as operações, assegurando que o processo de criação de agendamento funcione corretamente e esteja em conformidade com os requisitos e padrões estabelecidos pela instituição.

5.2.6 Ações

As ações são as funcionalidades disponíveis para a manipulação dos agendamentos. Elas permitem excluir, editar, duplicar, regerar e criar um agendamento.

- **Criar Agendamento:** Esta ação invoca a importação dos arquivos de entrada e executa o algoritmo CP-SAT, que aplica as *constraints* modeladas no SARA. O resultado é persistido no banco de dados e exibido na forma de um calendário de aulas.
- **Editar:** Permite visualizar o agendamento gerado e fazer alterações nele. Com esta funcionalidade, é possível ajustar horários, locais e outros detalhes conforme necessário.
- **Duplicar:** Cria uma cópia de um agendamento. Desta forma, o SARA possibilita a realização de testes e simulações sem comprometer o agendamento original.
- **Regerar:** Exclui todas as alterações feitas no agendamento e executa o algoritmo novamente sobre este agendamento. Neste caso, o agendamento poderá retornar ao seu estado original. Dependendo da complexidade e da quantidade de possíveis soluções do agendamento, esta função poderá demorar muito tempo.
- **Excluir:** Permite remover um agendamento indesejado ou obsoleto do SARA.

5.3 Implementação

A partir do modelo proposto e das características apresentadas, todo o conceito foi implementado na forma de um sistema web. Seu desenvolvimento neste formato envolveu a integração de diferentes tecnologias, visando proporcionar uma experiência de usuário intuitiva enquanto cria agendamentos otimizados.

5.3.1 Tecnologias Utilizadas

Para a implementação do SARA, foram utilizadas as tecnologias e *frameworks* apresentados abaixo:

- **Python e Flask:** O núcleo da aplicação web foi desenvolvido em Python utilizando o framework Flask, que oferece uma estrutura leve e flexível para a criação de aplicações web. Flask foi escolhido por sua simplicidade e extensibilidade, permitindo a rápida integração de diversos componentes do sistema.
- **SQLAlchemy:** Foi utilizado para a gestão do banco de dados, proporcionando uma interface ORM (Object-Relational Mapping) eficiente. Isso facilitou a manipulação dos dados e a implementação das operações de CRUD (Create, Read, Update, Delete) necessárias para a aplicação.
- **Google OR-Tools (CP-SAT):** O solucionador CP-SAT do Google OR-Tools foi integrado ao SARA para lidar com o problema de agendamento, garantindo que todas as restrições e preferências fossem consideradas.
- **HTML, CSS e Bootstrap:** A interface do usuário foi desenvolvida utilizando HTML e CSS, com o auxílio do framework Bootstrap para garantir um design responsivo e uma boa experiência de usuário em diferentes dispositivos.
- **JavaScript:** Utilizado para adicionar interatividade à aplicação, permitindo funcionalidades dinâmicas como arrastar e soltar, além de crítica algorítmica em tempo real para a gestão de horários e eventos.

Arquitetura do Projeto

A arquitetura adotada neste projeto segue uma abordagem modular utilizando o padrão de Blueprints do Flask, que permite organizar uma aplicação web em componentes independentes e reutilizáveis. Essa organização facilita a manutenção, a escalabilidade e o desenvolvimento colaborativo, especialmente em projetos maiores.

Organização do Projeto

O projeto é estruturado em pastas e subpastas que representam diferentes funcionalidades e componentes da aplicação. A estrutura geral é dividida em diretórios principais, como `app`, que contém a lógica principal, `blueprints`, que organiza os módulos da aplicação, `model`, que lida com o acesso a dados e as definições de modelo, e `static` e `templates`, que armazenam recursos estáticos e páginas HTML.

- `sara/`
 - `app/`
 - * `blueprints/`
 - * `model/`
 - * `static/`
 - * `templates/`
 - `instance/`
 - * `agendamentos.db`
- `config.py`
- `Entidades.py`
- `integrador.py`
- `readme.md`
- `requirements.txt`
- `run.py`
- `wsgi.py`

Uso de Blueprints

O Flask Blueprint é uma maneira de dividir uma aplicação Flask em múltiplos módulos, onde cada módulo pode ter suas próprias rotas, templates e lógica de negócio. Neste projeto, há várias Blueprints configuradas, como `scheduling`, `calendar`, `help`, `main` e `user`, cada uma representando uma funcionalidade específica.

- **scheduling**: responsável por gerenciar os processos de agendamento, com rotas para configurar horários e exibir a interface de agendamento.
- **calendar**: lida com a exibição e manipulação de calendários, permitindo visualizar eventos ou compromissos.

- **help:** fornece páginas de ajuda e suporte ao usuário, com informações e instruções sobre o uso do sistema.
- **main:** contém as rotas principais da aplicação, como a página inicial (`index.html`) e a página de login (`login.html`).
- **user:** gerencia o cadastro e a administração de usuários, incluindo a criação e gerenciamento de contas.

Essa separação por Blueprints facilita o desenvolvimento de novas funcionalidades, pois cada módulo pode ser trabalhado de forma independente, com suas próprias rotas e lógica encapsulada. Além disso, torna o projeto mais organizado e legível, já que as rotas e as funcionalidades são agrupadas por domínio.

Diretório `app`

Dentro do diretório `app`, há subdiretórios como `model`, `static`, `templates` e `blueprints`, além de arquivos de configuração, como `__init__.py` e `extensions.py`. O arquivo `__init__.py` é responsável por configurar e inicializar a aplicação Flask, incluindo o registro dos Blueprints, enquanto o `extensions.py` lida com a configuração de extensões adicionais, como bibliotecas de autenticação ou banco de dados.

Diretório `model`

A camada de dados é gerida no diretório `model`, que inclui arquivos como `database.py` e `models.py`. Estes arquivos são responsáveis por definir as classes de modelo da aplicação e a lógica para interagir com o banco de dados. A separação da lógica de dados do restante do código contribui para uma arquitetura mais limpa e desacoplada.

Diretórios `static` e `templates`

O diretório `static` armazena arquivos como folhas de estilo CSS, imagens e scripts JavaScript, que são usados na interface da aplicação. O diretório `templates` contém arquivos HTML que são renderizados dinamicamente, utilizando o motor de templates Jinja2 do Flask.

Vantagens da Arquitetura com Flask Blueprints

- **Modularidade:** Cada funcionalidade da aplicação é encapsulada em seu próprio módulo, o que facilita o desenvolvimento paralelo e a manutenção do código.

- **Reutilização de Código:** Blueprints permitem que partes da aplicação sejam reutilizadas em diferentes projetos ou contextos, tornando o desenvolvimento mais eficiente.
- **Organização e Manutenção:** Separar o código em diferentes módulos e camadas torna o projeto mais fácil de entender e gerenciar, especialmente em grandes aplicações.
- **Escalabilidade:** A adição de novas funcionalidades pode ser feita sem impacto significativo na arquitetura existente, permitindo que o sistema cresça de forma sustentável.

Essa arquitetura modular é uma prática recomendada para aplicações Flask e oferece uma base sólida para o desenvolvimento de projetos web, garantindo organização, manutenibilidade e facilidade de expansão.

Capítulo 6

Resultados

Neste capítulo, apresentamos os detalhes dos experimentos realizados para avaliar o desempenho do algoritmo proposto utilizando programação por restrições (*CP-SAT*) na resolução do problema de agendamento universitário. A seção aborda o ambiente de testes, as características dos conjuntos de dados reais utilizados e as métricas de desempenho analisadas. Foram executados experimentos com onze diferentes conjuntos de dados que variam em complexidade e tamanho, permitindo uma avaliação abrangente da escalabilidade e eficiência do solucionador em encontrar soluções ótimas.

Os resultados obtidos demonstraram que o *CP-SAT* é escalável, sendo capaz de resolver instâncias de diferentes magnitudes, desde pequenos conjuntos de dados até grandes instâncias com milhões de variáveis e restrições. Em todos os casos, o solucionador foi capaz de encontrar soluções ótimas de forma eficiente, destacando seu potencial para aplicação em cenários reais de agendamento.

Além disso, são discutidos os processos de pré-resolução aplicados pelo *CP-SAT*, que têm como objetivo simplificar os modelos ao reduzir a quantidade de variáveis e restrições, tornando os problemas mais tratáveis. O comportamento do solucionador é analisado de acordo com o aumento no tamanho dos conjuntos de dados, evidenciando a escalabilidade do *CP-SAT* em termos de tempo de processamento.

Por fim, os resultados experimentais são visualizados por meio de tabelas e gráficos, que auxiliam na compreensão da evolução do tempo de execução e da complexidade dos modelos à medida que os conjuntos de dados crescem em tamanho.

6.1 Configuração do ambiente experimental

Para a execução dos testes, utilizamos uma máquina virtual rodando o sistema operacional Ubuntu, configurada inicialmente com um processador de 16 núcleos (CPU E5-2650) e 16GB de memória RAM, além de Python 3.9 como ambiente de desenvolvimento prin-

cipal. Essa configuração inicial foi projetada para suportar operações de simulação de agendamentos; no entanto, à medida que os testes avançaram, percebemos que a quantidade de memória RAM era insuficiente para os cenários de teste mais complexos.

Durante as primeiras rodadas de teste, o limite de 16GB de RAM era rapidamente atingido, levando a uma solicitação para expansões de memória até 256GB, o que permitiu que o ambiente experimental suportasse simulações de alta carga.

6.2 Conjuntos de dados

Os dados utilizados nos experimentos consistem em onze conjuntos de dados reais diferentes, resumidos na Tabela 6.1. Cada evento pode ter uma ou mais ocorrências por semana, dependendo do seu valor de crédito. Consideramos 90 períodos de tempo, o que significa que uma semana consiste em 15 períodos por dia. Os resultados experimentais mostraram que o algoritmo proposto com programação por restrições gera soluções ótimas em todos os casos. O "Tempo 1st." representa o tempo em segundos que o solucionador precisou para encontrar a primeira solução, "Tempo SO" é o tempo em segundos que o solucionador precisou para encontrar a solução ótima, e "Alocações" refere-se aos eventos e seus créditos.

Tabela 6.1: Sumário dos conjuntos de dados

DS	Eventos	Salas	Alocações	Tempo 1st.	Tempo OS
1	11	06	52	4.24	4.37
2	28	20	118	16.84	18.44
3	44	12	186	45.91	48.22
4	67	15	312	107.66	117.19
5	103	38	422	152.99	196.26
6	101	20	694	71.42	364.91
7	163	66	702	360.86	593.31
8	208	56	947	244.53	582.29
9	264	80	1396	703.66	1903.68
10	605	145	2939	1713.40	18015.2
11	1473	188	6278	7825.21	50734.9

6.3 Coleta de dados

Para a coleta de dados durante os testes, utilizamos o parâmetro *log_search_progress* do *solver CP-SAT*. Esse parâmetro controla a frequência com que o solver registra informações sobre o progresso da busca durante o processo de resolução do problema.

Ao ativar *log_search_progress*, o *solver* gera *logs* detalhados em tempo real, incluindo informações sobre o progresso da busca, como o número de nós explorados, a profundidade

de busca, o tempo decorrido, e as melhorias na solução (caso a busca seja interrompida em uma solução intermediária).

Especificamente, no contexto dos testes, a ativação deste parâmetro possibilitou a coleta de métricas para identificar gargalos e entender o impacto de diferentes configurações e restrições no tempo de execução. Com esses dados, foi possível otimizar o desempenho do sistema, ajustando parâmetros e verificando a eficiência do modelo em cenários simples e de alta complexidade.

6.4 Análise individual das instâncias

Nesta seção, apresentamos os resultados obtidos para cada conjunto de dados, abrangendo as instâncias de DS1 até DS11. A seguir, discutimos as características e os resultados específicos de cada instância.

6.4.1 DS1: Características e Análise de Desempenho

O primeiro conjunto de dados, DS1, foi resolvido utilizando o solucionador *CP-SAT*, configurado com parâmetros como *max_time_in_seconds*: 1800, *num_search_workers*: 16 e *cp_model_presolve*: *true*. Este conjunto de dados consistia em 6.094 variáveis no modelo inicial, das quais 6.083 eram variáveis booleanas e 11 variáveis inteiras com domínio [0,5], além de 5.940 restrições lineares.

Durante o processo de pré-resolução, foram aplicadas otimizações, como a detecção de redundâncias e a eliminação de restrições triviais, resultando em uma redução significativa no número de variáveis e restrições. Após a pré-resolução, o modelo foi simplificado para 4.747 variáveis e 1.947 restrições.

Ao iniciar a busca por soluções, o solucionador foi capaz de encontrar a primeira solução viável em 4,24 segundos. O tempo total para alcançar a solução ótima foi de 4,37 segundos, evidenciando o bom desempenho do *CP-SAT* ao resolver problemas de menor escala como este.

Esses resultados reforçam a capacidade do *CP-SAT* em lidar com instâncias pequenas de maneira eficiente, produzindo soluções ótimas em tempo hábil e explorando eficientemente o espaço de busca, o que é demonstrado pela convergência rápida para a solução ótima.

6.4.2 DS2: Características e Análise de Desempenho

O segundo conjunto de dados, DS2, foi resolvido utilizando o solucionador *CP-SAT*, configurado com parâmetros semelhantes ao DS1, incluindo *max_time_in_seconds*: 1800 e

num_search_workers: 16. Este conjunto de dados apresentava um problema de maior escala, com 51.576 variáveis no modelo inicial, das quais 51.548 eram booleanas e 28 eram inteiras com valores entre [0,19]. Além disso, o modelo incluía 50.400 restrições lineares.

Durante o processo de pré-resolução, o solucionador aplicou técnicas para simplificar o problema, resultando em uma redução significativa no número de variáveis e restrições. Após a pré-resolução, o número de variáveis foi reduzido para 33.961 e o número de restrições para 5.178.

A busca pela primeira solução viável foi concluída em 16,84 segundos, enquanto o tempo total para encontrar a solução ótima foi de 18,44 segundos, o que demonstra um aumento considerável no tempo de execução comparado ao DS1, devido ao aumento da complexidade do problema. A análise das soluções encontradas revelou que foram geradas 54 soluções ao longo do processo de otimização.

O solucionador empregou uma combinação de *subsolvers* e estratégias de busca paralela para explorar o espaço de soluções de forma eficiente, utilizando, por exemplo, o algoritmo *violation_ls*, responsável por encontrar e refinar as soluções viáveis.

Esses resultados mostram a escalabilidade do solucionador *CP-SAT* quando se trata de problemas maiores e mais complexos, sendo capaz de encontrar soluções ótimas em um tempo razoável, apesar da crescente complexidade dos conjuntos de dados.

6.4.3 DS3: Características e Análise de Desempenho

O terceiro conjunto de dados, DS3, foi resolvido utilizando o solucionador *CP-SAT* com a configuração padrão de *num_search_workers*: 16 e *max_time_in_seconds*: 1800. Esse conjunto de dados apresentou 48.664 variáveis no modelo inicial, das quais 48.620 eram booleanas, com valores entre [0,1], e 44 variáveis inteiras com valores de [0,11]. O modelo inicial também continha 47.520 restrições lineares.

Durante a pré-resolução, o solucionador aplicou técnicas de simplificação que resultaram na redução do modelo para 42.005 variáveis e 6.245 restrições, evidenciando a eficácia do processo de otimização da pré-resolução. Esse processo foi especialmente eficiente na detecção de redundâncias e na simplificação de variáveis e restrições lineares complexas.

O tempo necessário para encontrar a primeira solução viável foi de 45,91 segundos, e a solução ótima foi alcançada em 48,22 segundos. O solucionador explorou múltiplas estratégias de busca paralela, utilizando 16 *subsolvers*, que variaram entre métodos como *fj_long_random*, *violation_ls*, e *fj_short_default*. Isso garantiu uma busca eficiente em um espaço de soluções vasto, o que permitiu a rápida convergência para a solução ótima.

O solucionador explorou 73 soluções diferentes ao longo do processo de otimização, utilizando técnicas como busca local baseada em violações (*violation_ls*) e perturbações lineares, que se mostraram eficazes para lidar com a complexidade do problema. Além

disso, o processo de simetria também foi explorado, identificando orbitope de tamanho 461×2 , o que ajudou a reduzir a redundância e acelerar o processo de busca.

Os resultados mostram que, embora DS3 seja um problema significativamente maior em comparação aos anteriores, o *CP-SAT* conseguiu resolver o problema em um tempo razoável, demonstrando sua capacidade de escalar para problemas de maior complexidade e com um número elevado de variáveis e restrições.

6.4.4 DS4: Características e Análise de Desempenho

O quarto conjunto de dados, DS4, foi resolvido utilizando o solucionador *CP-SAT* configurado com *num_search_workers*: 16 e *max_time_in_seconds*: 1800. Esse conjunto apresentou 92.594 variáveis, sendo 92.527 variáveis booleanas no intervalo $[0,1]$ e 67 variáveis inteiras com valores entre $[0,14]$. Além disso, o modelo continha 90.450 restrições lineares.

Durante a pré-resolução, o solucionador aplicou técnicas para simplificar o modelo, resultando em uma redução significativa no número de variáveis e restrições. Após a pré-resolução, o modelo foi reduzido para 82.009 variáveis e 9.456 restrições, o que demonstra a eficácia das técnicas de simplificação aplicadas.

O tempo necessário para encontrar a primeira solução viável foi de 107,66 segundos, e a solução ótima foi alcançada em 117,19 segundos. Isso reflete o aumento na complexidade do problema em relação aos conjuntos de dados anteriores, visto que a quantidade de variáveis e restrições é consideravelmente maior.

O solucionador *CP-SAT* utilizou estratégias de busca avançadas e paralelas, incluindo *subsolvers* como *fj_long_default*, *violation_ls*, e *fj_long_perturb*. A exploração do espaço de soluções foi realizada de forma eficiente, levando em consideração as simetrias do problema e utilizando a técnica de quebra de simetria, que identificou um orbitope de tamanho 1367×2 , o que contribuiu para a redução do tempo de busca.

Ao longo da execução, o solucionador encontrou 3.151 soluções diferentes antes de convergir para a solução ótima, utilizando técnicas de violação de restrições (*violation_ls*) e perturbações na busca. O uso dessas técnicas foi essencial para lidar com a complexidade e o tamanho do conjunto de dados, permitindo uma exploração mais eficiente do espaço de soluções.

Esses resultados demonstram que o *CP-SAT* é capaz de lidar eficientemente com problemas de maior escala, mantendo a escalabilidade e encontrando soluções ótimas em tempos razoáveis, mesmo em conjuntos de dados com um número elevado de variáveis e restrições.

6.4.5 DS5: Características e Análise de Desempenho

O quinto conjunto de dados, DS5, foi resolvido utilizando o solucionador *CP-SAT* configurado com os parâmetros *max_time_in_seconds*: 1800, *num_search_workers*: 16, e *cp_model_presolve*: *true*. Esse conjunto apresentou uma grande escala de complexidade, com 360.294 variáveis no modelo inicial, das quais 360.191 eram variáveis booleanas e 103 variáveis inteiras com valores entre [0,37]. O modelo também incluía 352.260 restrições lineares, juntamente com 3.914 restrições *lin-max* e 3.420 restrições *at-most-one*.

Durante o processo de pré-resolução, o solucionador aplicou técnicas eficientes para simplificar o modelo, resultando em uma redução significativa no número de variáveis e restrições. Após a pré-resolução, o modelo foi reduzido para 140.287 variáveis e 15.649 restrições, o que evidencia a eficácia das técnicas de otimização, particularmente na eliminação de restrições redundantes e variáveis triviais.

O tempo necessário para encontrar a primeira solução viável foi de 152,99 segundos, e a solução ótima foi alcançada em 196,26 segundos. Ao longo da execução, o solucionador explorou diversas estratégias de busca paralela com *subsolvers* como *violation_ls* e *fj_long_lin_default*, resultando em uma busca eficiente e em soluções otimizadas para um problema desta magnitude.

Foram geradas 62.780 soluções diferentes ao longo do processo de otimização, com técnicas de simetria aplicadas, identificando orbitos significativos, o que ajudou a reduzir o tempo de busca. O solucionador também utilizou orbitope de tamanho 1540 x 2, que contribuiu para a exploração mais eficiente do espaço de soluções.

Os resultados mostram que, mesmo diante de um aumento significativo no número de variáveis e restrições, o *CP-SAT* manteve sua capacidade de escalabilidade, fornecendo soluções ótimas em tempos aceitáveis. A exploração do espaço de busca e as técnicas de pré-resolução foram essenciais para lidar com a complexidade desse conjunto de dados, destacando o potencial do solucionador para grandes problemas de agendamento.

6.4.6 DS6: Características e Análise de Desempenho

O sexto conjunto de dados, DS6, foi resolvido utilizando o solucionador *CP-SAT* configurado com *num_search_workers*: 16, *max_time_in_seconds*: 7200, e *cp_model_presolve*: *true*. O modelo inicial apresentou 186.042 variáveis, sendo 185.941 booleanas e 101 variáveis inteiras no intervalo [0,19]. O modelo continha 181.800 restrições lineares e diversas outras restrições, como 1.800 *at-most-one* e 2.020 *lin-max*.

Durante o processo de pré-resolução, o modelo foi simplificado significativamente, com o número de variáveis reduzido para 161.892 e o número de restrições lineares reduzido

para 14.946. A técnica de simetria aplicada identificou um orbitope de tamanho 1822×2 , o que ajudou a eliminar redundâncias e acelerar o processo de busca.

O tempo necessário para encontrar a primeira solução viável foi de 71,42 segundos, e a solução ótima foi alcançada em 364,91 segundos, o que reflete o aumento na complexidade do problema em relação aos conjuntos de dados anteriores. Durante o processo de otimização, o solucionador utilizou estratégias avançadas de busca, incluindo métodos como *violation_ls* e *fj_long_random*, para explorar o vasto espaço de soluções de maneira eficiente.

Ao longo da execução, o solucionador gerou 702 soluções antes de convergir para a solução ótima. As técnicas de simetria e violação de restrições foram fundamentais para reduzir o tempo de busca e maximizar a eficiência do solucionador.

Esses resultados mostram que, mesmo com o aumento substancial na quantidade de variáveis e restrições, o *CP-SAT* manteve sua eficiência, fornecendo soluções ótimas em tempos razoáveis e utilizando técnicas avançadas de pré-resolução e simetria para lidar com a complexidade do problema.

6.4.7 DS7: Características e Análise de Desempenho

O sétimo conjunto de dados, DS7, foi resolvido utilizando o solucionador *CP-SAT* configurado com *num_search_workers*: 16, *max_time_in_seconds*: 7200, e *cp_model_presolve*: *true*. O modelo inicial apresentou 990.062 variáveis, sendo 989.899 booleanas e 163 inteiras com valores entre $[0,65]$. O modelo inicial continha 968.220 restrições lineares, além de restrições adicionais, como 5.940 *at-most-one* e 10.758 *lin-max*.

Durante o processo de pré-resolução, o modelo foi reduzido para 779.985 variáveis e 38.543 restrições, destacando a eficácia das técnicas de pré-resolução aplicadas pelo *CP-SAT*. O uso de técnicas avançadas de detecção de simetria e eliminação de restrições redundantes desempenhou um papel importante na simplificação do modelo, o que facilitou a busca por soluções.

O tempo necessário para encontrar a primeira solução viável foi de 360,86 segundos, enquanto o tempo total para alcançar a solução ótima foi de 593,31 segundos, refletindo a complexidade crescente em relação aos conjuntos de dados anteriores. O solucionador explorou diversas estratégias de busca paralela, incluindo *subsolvers* como *fj_long_default*, *violation_ls*, e *fj_long_lin_perturb*, que permitiram uma exploração eficiente do vasto espaço de soluções.

Foram geradas 702 soluções diferentes durante o processo de otimização, com o solucionador empregando técnicas de quebra de simetria, que identificaram um orbitope de tamanho 327×2 , contribuindo para a eficiência na resolução do problema. A simetria

e o uso de perturbações em soluções anteriores ajudaram a reduzir significativamente o tempo de busca.

Esses resultados demonstram a escalabilidade do solucionador *CP-SAT*, que, mesmo diante de um aumento significativo no número de variáveis e restrições, conseguiu encontrar soluções ótimas em um tempo aceitável, utilizando técnicas avançadas de pré-resolução e busca paralela.

6.4.8 DS8: Características e Análise de Desempenho

O oitavo conjunto de dados, DS8, foi resolvido utilizando o solucionador *CP-SAT* configurado com *num_search_workers*: 16, *max_time_in_seconds*: 1800, e *cp_model_presolve*: *true*. O modelo inicial apresentou 1.072.032 variáveis, das quais 1.071.824 eram booleanas no intervalo $[0,1]$ e 208 variáveis inteiras, com valores entre $[0,55]$. Além disso, o modelo continha 1.048.320 restrições lineares, 5.040 restrições *at-most-one* e 11.648 restrições *lin-max*.

O processo de pré-resolução foi altamente eficiente, resultando em uma simplificação significativa do modelo. Após a pré-resolução, o modelo foi reduzido para 956.132 variáveis e 45.781 restrições, evidenciando o impacto das técnicas de simplificação e detecção de simetria aplicadas pelo *CP-SAT*.

O tempo necessário para encontrar a primeira solução viável foi de 244,53 segundos, e o tempo total para encontrar a solução ótima foi de 582,29 segundos. O solucionador utilizou várias estratégias de busca paralela, como os *subsolvers* *fj_long_default*, *fj_long_lin_default*, *violation_ls*, entre outros. Essas estratégias permitiram uma exploração eficiente do vasto espaço de soluções, levando a uma rápida convergência para a solução ótima.

Ao longo da execução, foram geradas 947 soluções, com o solucionador utilizando técnicas como busca local baseada em violações (*violation_ls*) para refinar as soluções viáveis e otimizar o desempenho. A simetria detectada no modelo também desempenhou um papel importante na redução do espaço de busca, facilitando o processo de otimização.

Os resultados mostram que, apesar do aumento significativo no número de variáveis e restrições em relação aos conjuntos de dados anteriores, o *CP-SAT* foi capaz de escalar eficientemente, fornecendo soluções ótimas em tempos razoáveis.

6.4.9 DS9: Características e Análise de Desempenho

O nono conjunto de dados, DS9, foi resolvido utilizando o solucionador *CP-SAT* configurado com *max_time_in_seconds*: 14400, *num_search_workers*: 16, e *cp_model_presolve*: *true*. O modelo inicial apresentou 1.943.568 variáveis, sendo 1.943.304 booleanas e 264

inteiras, com valores variando entre $[0,79]$. Além disso, o modelo continha 1.900.800 restrições lineares, 7.200 restrições *at-most-one*, e 21.120 *lin-max*.

O processo de pré-resolução foi eficiente, resultando em uma simplificação do modelo, reduzindo-o para 1.561.346 variáveis e 66.571 restrições. A aplicação de técnicas de simetria e eliminação de variáveis e restrições redundantes foi importante para essa redução, acelerando o processo de busca.

O tempo necessário para encontrar a primeira solução viável foi de 703,66 segundos, e a solução ótima foi encontrada em 1.903,68 segundos, o que reflete o aumento substancial da complexidade do problema em comparação com os conjuntos de dados anteriores. O solucionador explorou múltiplas estratégias de busca, utilizando *subsolvers* como *fj_long_lin_default* e *violation_ls*, o que permitiu uma exploração eficiente do vasto espaço de soluções.

Durante o processo de otimização, foram geradas 1.396 alocações e 4086 soluções foram exploradas antes de convergir para a solução ótima. O solucionador aplicou técnicas de simetria, identificando padrões no modelo que ajudaram a refinar e acelerar o processo de otimização.

Esses resultados demonstram a capacidade do *CP-SAT* em lidar com conjuntos de dados de grande escala, sendo capaz de encontrar soluções ótimas dentro de um tempo viável, mesmo em problemas altamente complexos e com um grande número de variáveis e restrições.

6.4.10 DS10: Características e Análise de Desempenho

O décimo conjunto de dados, DS10, foi resolvido utilizando o solucionador *CP-SAT* configurado com *max_time_in_seconds*: 28800, *num_search_workers*: 16, e *cp_model_presolve*: *true*. O modelo inicial apresentou 8.071.910 variáveis, das quais 8.071.305 eram booleanas no intervalo $[0,1]$ e 605 variáveis inteiras com valores entre $[0,144]$. Além disso, o modelo continha 8.052.895 restrições lineares, 13.050 restrições *at-most-one*, e 87.725 restrições *lin-max*.

O processo de pré-resolução foi eficiente, resultando em uma significativa simplificação do modelo, que foi reduzido para 6.649.882 variáveis e 216.160 restrições. A aplicação de técnicas de eliminação de redundâncias e redução de variáveis e restrições triviais foi fundamental para essa otimização.

O tempo necessário para encontrar a primeira solução viável foi de 1.713,40 segundos, e a solução ótima foi alcançada em 18.015,2 segundos, refletindo a alta complexidade e o tamanho do problema. O solucionador utilizou múltiplas estratégias de busca paralela, como *violation_ls* e *fj_long_lin_default*, que permitiram uma exploração eficiente do vasto espaço de soluções.

Durante o processo de otimização, foram geradas 2.939 alocações, e o solucionador explorou 50.734 soluções diferentes antes de convergir para a solução ótima. O uso de técnicas avançadas de simetria, que identificaram padrões complexos no modelo, foi essencial para lidar com a magnitude do problema.

Esses resultados demonstram a escalabilidade e a eficiência do *CP-SAT* na resolução de problemas de grande escala, mesmo com um número substancial de variáveis e restrições.

6.4.11 DS11: Características e Análise de Desempenho

O décimo primeiro e maior conjunto de dados, DS11, foi resolvido utilizando o solucionador *CP-SAT* com parâmetros configurados para lidar com sua grande escala, como *max_time_in_seconds*: 28800, *num_search_workers*: 16, e *cp_model_presolve*: *true*. O modelo inicial consistia em 25.479.954 variáveis, sendo 25.478.481 booleanas e 1.473 variáveis inteiras com valores entre [0,187]. O modelo também incluía 25.355.466 restrições lineares, além de 16.920 *at-most-one* e 276.924 *lin-max*.

Durante o processo de pré-resolução, o modelo foi reduzido significativamente para 18.899.675 variáveis e 660.171 restrições, demonstrando a eficácia das técnicas de otimização aplicadas pelo *CP-SAT*, que eliminaram restrições redundantes e variáveis triviais, acelerando a resolução do problema.

O tempo necessário para encontrar a primeira solução viável foi de 7.825,21 segundos, e a solução ótima foi alcançada em 50.734,9 segundos, refletindo o enorme tamanho e complexidade deste conjunto de dados. O solucionador utilizou múltiplas estratégias de busca, incluindo *subsolvers* como *fj_long_default*, *fj_long_lin_default*, *violation_ls*, e outros, permitindo uma exploração eficiente de um vasto espaço de soluções.

Foram geradas 6.278 alocações, e o solucionador explorou 4.086 soluções ao longo do processo de otimização antes de convergir para a solução ótima. A técnica de simetria aplicada identificou padrões no modelo, o que foi fundamental para reduzir o tempo de busca e melhorar a eficiência geral.

Esses resultados demonstram a robustez e escalabilidade do *CP-SAT* na resolução de problemas extremamente grandes e complexos, fornecendo soluções ótimas em um tempo viável, mesmo em cenários com milhões de variáveis e restrições.

6.5 Processos de pré-resolução

Ao analisar os detalhes do modelo resumidos na Tabela 6.2, para o primeiro conjunto de dados, o modelo inicial compreende 6.094 variáveis e 7.580 restrições. Após a pré-resolução, esses números são reduzidos para 4.747 variáveis e 1.947 restrições. Essa redução destaca a eficiência do processo de pré-resolução em simplificar o problema, tornando-o

Tabela 6.2: Sumário dos detalhes do modelo

DS	Variáveis IM	Variáveis PM	<i>Constraints</i> do modelo inicial	<i>Constraints</i> do modelo pré-solucionado
1	6.094	4.747	7.580	1.947
2	51.576	33.961	55.392	5.178
3	48.664	42.005	53.264	6.245
4	92.594	82.009	99.103	9.456
5	360.294	140.287	369.276	15.649
6	186.042	161.892	195.114	14.946
7	990.062	779.985	1.000.240	38.543
8	1.072.032	956.132	1.084.560	45.781
9	1.943.568	1.561.346	1.953.936	66.571
10	8.071.910	6.649.882	8.052.895	216.160
11	25.479.954	18.899.675	25.355.466	660.171

mais tratável para o solucionador. Em conjuntos de dados maiores, como o conjunto de dados 5, o modelo inicial possui 360.294 variáveis e 369.276 restrições. Após a pré-resolução, esses números são reduzidos para 140.287 variáveis e 15.649 restrições. A diminuição substancial tanto de variáveis quanto de restrições demonstra a capacidade do processo de pré-resolução de lidar com problemas mais complexos de forma eficaz. Para o maior conjunto de dados, o conjunto de dados 11, o modelo inicial contém 25.479.954 variáveis e 25.355.466 restrições. Após a pré-resolução, esses números caem para 18.899.675 variáveis e 660.171 restrições. Essa redução significativa reforça a escalabilidade das técnicas de pré-resolução utilizadas no solucionador *CP-SAT*.

Os dados sugerem que a pré-resolução pode ser uma etapa essencial no processo de otimização, especialmente para problemas de agendamento em larga escala. Ao eliminar restrições redundantes e reduzir o número de variáveis, o processo de pré-resolução melhora o desempenho do solucionador, permitindo que ele encontre soluções ótimas de maneira mais eficiente.

6.6 Análise do comportamento do solucionador

Ao analisar as características e as métricas de desempenho dos conjuntos de dados, podemos tirar conclusões sobre a eficiência e a escalabilidade do solucionador *CP-SAT* no tratamento de diferentes tamanhos de problemas.

Para o primeiro conjunto de dados, que consiste em 11 eventos, 6 salas e 52 alocações, o solucionador *CP-SAT* encontrou a primeira solução em 4,24 segundos e a solução ótima em 4,37 segundos. Em contraste, o segundo conjunto de dados com 28 eventos, 20 salas e 118 alocações exigiu 16,84 segundos para a primeira solução e 18,44 segundos para a

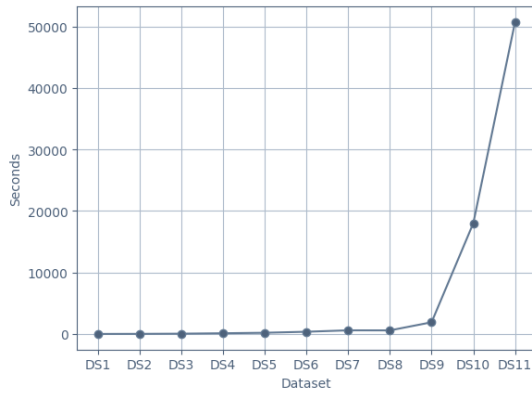


Figura 6.1: Tempo para encontrar a solução ótima.

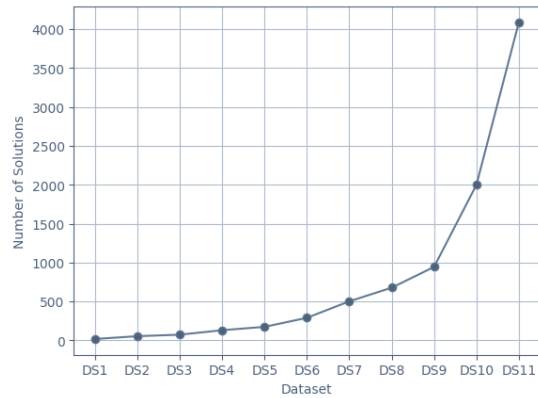


Figura 6.2: Número de soluções encontradas.

solução ótima. Embora o solucionador tenha demonstrado eficiência, o aumento no tamanho do problema levou a um aumento perceptível no tempo de execução. À medida que os conjuntos de dados se tornam maiores, o tempo necessário para encontrar soluções aumenta significativamente. O maior conjunto de dados exigiu 7.825,21 segundos para encontrar a primeira solução e 50.734,9 segundos para atingir a solução ótima, destacando o crescimento exponencial no tempo de computação com o aumento do tamanho do problema. Essa tendência é ilustrada na Figura 6.1, que mostra o tempo necessário para encontrar a solução ótima para cada conjunto de dados.

O número de soluções fornece uma visão sobre a eficiência do solucionador e a complexidade do conjunto de dados. No DS11, o solucionador encontrou 4.086 soluções antes de chegar à solução ótima, destacando o vasto espaço de busca e a capacidade do solucionador de explorar várias alternativas. Por outro lado, em conjuntos de dados menores, como o DS1, apenas 18 soluções foram encontradas antes de identificar a solução ótima. A variação no número de soluções entre os conjuntos de dados é visualizada na Figura 6.2.

Capítulo 7

Conclusão e trabalhos futuros

A pesquisa apresentada nesta dissertação investigou o problema do agendamento universitário, que representa um desafio para instituições de ensino superior, especialmente devido à necessidade de equilibrar múltiplas *constraints* e preferências específicas, durante o processo de criação de grade horária e alocação de espaços físicos para as aulas. Diante desse contexto, investigamos, propomos e implementamos um modelo para a alocação de recursos acadêmicos, utilizando técnicas de programação por *constraints* e a ferramenta *CP-SAT* do *Google OR-Tools*. O modelo SARA foi implementado na plataforma SARA, que serviu como ambiente de testes e validação da abordagem, permitindo que explorássemos o desempenho da solução em diferentes cenários e escalas de dados.

A primeira fase do trabalho consistiu em uma revisão sistemática da literatura, onde foram analisados os principais métodos e algoritmos utilizados na solução de problemas de *timetabling* acadêmico. Este levantamento possibilitou identificar abordagens relevantes, *constraints* frequentes e os principais desafios enfrentados por estudos anteriores. Com base nesses conhecimentos, delineamos um modelo que satisfaz as *constraints* essenciais – como evitar conflitos de horário e garantir alocações adequadas de salas e professores –, mas também incorpora preferências institucionais e metodológicas, oferecendo uma solução flexível e personalizável.

Na fase de implementação, o modelo foi submetido a uma série de testes com onze conjuntos de dados reais da Universidade de Brasília. Através da plataforma SARA, conseguimos executar experimentos que variaram desde casos mais simples, com um número reduzido de eventos e constraints, até cenários altamente complexos, com um volume significativo de eventos e de requisitos. Essa diversidade de testes foi fundamental para avaliar o comportamento do *CP-SAT* em condições reais de uso, permitindo uma análise profunda sobre a eficiência do solver em identificar soluções viáveis e otimizadas em contextos variados.

Um dos diferenciais da plataforma SARA é a crítica algorítmica, um módulo que

proporciona *feedback* em tempo real sobre a validade das alocações. Essa funcionalidade é especialmente útil para a visualização de conflitos e validações, pois permite que o usuário modifique os eventos no calendário enquanto o sistema verifica a consistência das alterações com as *constraints* definidas. Assim, sempre que um evento é alterado, o módulo de crítica algorítmica valida se as novas configurações geram conflitos com outros eventos, professores ou salas, e sinaliza ao usuário, facilitando a construção de soluções sem conflitos de forma mais intuitiva e interativa. Esse recurso agrega valor ao sistema, permitindo simulações rápidas e análises antes mesmo de concluir as alterações, o que melhora a experiência do usuário e minimiza o retrabalho na fase de planejamento.

Os resultados experimentais destacaram a eficácia do modelo em responder de forma precisa às necessidades de agendamento da universidade. Em conjuntos de dados menores, o *solver* encontrou soluções ótimas em poucos segundos, demonstrando que a abordagem é adequada para situações onde é necessário um processamento rápido. Em conjuntos de dados mais robustos, que envolvem um número elevado de eventos, o *CP-SAT* mostrou-se capaz de encontrar soluções ótimas também, ainda que o tempo de execução tenha aumentado, o que é esperado dada a natureza combinatória e a escala do problema.

O modelo desenvolvido através da SARA oferece também um sistema escalável, que pode ser adaptado para outras instituições de ensino superior com diferentes requisitos e características. Essa flexibilidade do modelo reforça sua aplicabilidade e sugere que ele pode ser utilizado como base para uma plataforma de agendamento de recursos acadêmicos em instituições com demandas variadas. Além disso, a capacidade de responder a *constraints* complexas e múltiplas preferências institucionais demonstra a relevância da programação por *constraints* como ferramenta para a resolução de problemas de *timetabling*, com potencial para auxiliar na modernização das práticas de gestão acadêmica e na otimização dos recursos institucionais.

Além dos ganhos operacionais, a aplicação deste modelo tem importantes implicações para as políticas acadêmicas e as práticas administrativas das instituições de ensino. O uso de um sistema como o SARA oferece agilidade no processo de alocação de recursos e um ambiente onde as necessidades de professores e alunos podem ser atendidas de forma mais integrada, contribuindo para uma experiência educacional mais satisfatória e equilibrada. Instituições que buscam modernizar suas práticas de gestão têm, neste modelo, uma ferramenta para transformar seus processos de planejamento acadêmico.

7.1 Trabalhos futuros

Como trabalho futuro, propomos explorar estratégias de decomposição que permitam dividir problemas mais complexos em subproblemas menores, facilitando a solução e re-

duzindo o tempo de execução. Além disso, técnicas de *machine learning* poderão ser integradas ao modelo, possibilitando uma personalização ainda mais precisa e uma melhor adaptação a variações no uso dos recursos e nas preferências dos usuários. Essa integração pode abrir novas possibilidades para o sistema SARA, transformando-o em uma plataforma mais robusta, capaz de prever e se adaptar a mudanças nas necessidades institucionais. Também planejamos implementar elementos adicionais do modelo SARA que ainda não foram desenvolvidos, como os módulos de *Feedback* e Avaliação, Comunicação e Notificação, Predição e Relatórios, ampliando assim o escopo do sistema. A inclusão de elementos de gamificação e de interação com os usuários poderá, por exemplo, contribuir para uma experiência mais intuitiva e engajadora, permitindo que docentes e alunos se beneficiem de um sistema de agendamento verdadeiramente adaptável e alinhado às estratégias pedagógicas mais modernas.

Referências Bibliográficas

ABDALLA, M. H. et al. Performance comparison of sequential and cooperative integer programming search methodologies in solving curriculum-based university course timetabling problems (CB-UCT). In: ALFRED, R. et al. (Ed.). *Computational Science and Technology*. Springer Singapore, 2019. v. 481, p. 145–154. ISBN 9789811326219 9789811326226. Series Title: Lecture Notes in Electrical Engineering. Disponível em: <http://link.springer.com/10.1007/978-981-13-2622-6_15>. 13

ABDULLAH, S.; BURKE, E. K.; McCollum, B. A hybrid evolutionary approach to the university course timetabling problem. In: *2007 IEEE Congress on Evolutionary Computation*. IEEE, 2007. p. 1764–1768. ISBN 978-1-4244-1339-3. Disponível em: <<https://ieeexplore.ieee.org/document/4424686/>>. 9, 16

ABRAMSON, D. Constructing school timetables using simulated annealing: sequential and parallel algorithms. *Management science*, INFORMS, v. 37, n. 1, p. 98–113, 1991. 9, 15, 19

AINI, I. F.; SAPTAWIJAYA, A.; AMINAH, S. Bringing answer set programming to the next level: A real case on modeling course timetabling. In: *2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*. IEEE, 2017. p. 471–476. ISBN 978-1-5386-3172-0. Disponível em: <<https://ieeexplore.ieee.org/document/8355076/>>. 13, 19

AKKAN, C.; GüLCü, A. A bi-criteria hybrid genetic algorithm with robustness objective for the course timetabling problem. v. 90, p. 22–32, 2018. ISSN 03050548. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S0305054817302307>>. 10, 16, 33

AKKAN, C.; GüLCü, A.; KUŞ, Z. Bi-criteria simulated annealing for the curriculum-based course timetabling problem with robustness approximation. *Journal of Scheduling*, Springer, v. 25, n. 4, p. 477–501, 2022. ISSN 1094-6136, 1099-1425. Disponível em: <<https://link.springer.com/article/10.1007/s10951-022-00722-0>>. 39

AL-BETAR, M. A.; KHADER, A. T. A harmony search algorithm for university course timetabling. v. 194, n. 1, p. 3–31, 2012. ISSN 0254-5330, 1572-9338. Disponível em: <<http://link.springer.com/10.1007/s10479-010-0769-z>>. 9, 19

AL-BETAR, M. A.; KHADER, A. T.; ZAMAN, M. University course timetabling using a hybrid harmony search metaheuristic algorithm. v. 42, n. 5, p. 664–681, 2012. ISSN 1094-6977, 1558-2442. Disponível em: <<http://ieeexplore.ieee.org/document/6135818/>>. 9

ALEXA, E.; ZMARANDA, D.; MOISI, E. V. Modeling and solving the university timetabling problem using the constraints satisfaction model: A case study. In: *2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*. IEEE, 2021. p. 1–6. ISBN 978-1-66542-534-6. Disponível em: <<https://ieeexplore.ieee.org/document/9515060/>>. 13

ASSI, M.; HALAWI, B.; HARATY, R. A. Genetic algorithm analysis using the graph coloring method for solving the university timetable problem. v. 126, p. 899–906, 2018. ISSN 18770509. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S1877050918313024>>. 10, 20, 33, 40

BABAEI, H.; KARIMPOUR, J.; HADIDI, A. A survey of approaches for university course timetabling problem. v. 86, p. 43–59, 2015. ISSN 03608352. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S0360835214003714>>. 9, 38, 40

BABAEI, H.; KARIMPOUR, J.; HADIDI, A. Applying hybrid fuzzy multi-criteria decision-making approach to find the best ranking for the soft constraint weights of lecturers in UCTP. *Int. J. Fuzzy Syst.*, v. 20, n. 1, p. 62–77, 2018. Disponível em: <<https://doi.org/10.1007/s40815-017-0296-z>>. 14, 16

BABAEI, H.; KARIMPOUR, J.; HADIDI, A. Generating an optimal timetabling for multi-departments common lecturers using hybrid fuzzy and clustering algorithms. v. 23, n. 13, p. 4735–4747, 2019. ISSN 1432-7643, 1433-7479. Disponível em: <<http://link.springer.com/10.1007/s00500-018-3126-9>>. 12, 16, 22

BAGGER, N.-C. F.; SØRENSEN, M.; STIDSEN, T. R. Dantzig–wolfe decomposition of the daily course pattern formulation for curriculum-based course timetabling. v. 272, n. 2, p. 430–446, 2019. ISSN 03772217. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S0377221718305794>>. 12

BANBARA, M. et al. $\{\varvec{\text{teaspoon}}\}$: solving the curriculum-based course timetabling problems with answer set programming. v. 275, n. 1, p. 3–37, 2019. ISSN 0254-5330, 1572-9338. Disponível em: <<http://link.springer.com/10.1007/s10479-018-2757-7>>. 11

BASHAB, A. et al. Optimization techniques in university timetabling problem: Constraints, methodologies, benchmarks, and open issues. v. 74, n. 3, p. 6461–6484, 2023. ISSN 1546-2226. Disponível em: <<https://www.techscience.com/cmc/v74n3/50939>>. 14

BURKE, E. K. et al. Automated university timetabling: The state of the art. *The Computer Journal*, Oxford University Press, v. 40, n. 9, p. 565–571, 1997. 1, 8, 20, 35, 40, 41, 48

BURKE, E. K. et al. A graph-based hyper-heuristic for educational timetabling problems. v. 176, n. 1, p. 177–192, 2007. ISSN 03772217. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S0377221705006387>>. 9, 14, 19, 35, 40

BURKE, E. K.; PETROVIC, S. Recent research directions in automated timetabling. v. 140, n. 2, p. 266–280, 2002. ISSN 03772217. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S0377221702000693>>. 8

BURKE, E. K.; PETROVIC, S.; QU, R. Case-based heuristic selection for timetabling problems. v. 9, n. 2, p. 115–132, 2006. ISSN 1094-6136, 1099-1425. Disponível em: <<http://link.springer.com/10.1007/s10951-006-6775-y>>. 9, 14, 18, 44

CARAMIA, M.; GIORDANI, S. Curriculum-based course timetabling with student flow, soft constraints, and smoothing objectives: an application to a real case study. v. 1, n. 2, p. 11, 2021. ISSN 2662-2556. Disponível em: <<https://link.springer.com/10.1007/s43069-020-0013-x>>. 13

CARTER, M. W.; LAPORTE, G. Recent developments in practical course timetabling. In: BURKE, E.; CARTER, M. (Ed.). *Practice and Theory of Automated Timetabling II*. Springer Berlin Heidelberg, 1998. v. 1408, p. 3–19. ISBN 978-3-540-64979-3 978-3-540-49803-2. Series Title: Lecture Notes in Computer Science. Disponível em: <<http://link.springer.com/10.1007/BFb0055878>>. 8

CHEN, M. C. et al. A survey of university course timetabling problem: Perspectives, trends and opportunities. v. 9, p. 106515–106529, 2021. ISSN 2169-3536. Disponível em: <<https://ieeexplore.ieee.org/document/9499056/>>. 4, 10

CHEN, M. C. et al. Investigation of heuristic orderings with a perturbation for finding feasibility in solving real- world university course timetabling problem. In: *2022 International Conference on Digital Transformation and Intelligence (ICDI)*. IEEE, 2022. p. 168–173. ISBN 9798350396980 9798350397000. Disponível em: <<https://ieeexplore.ieee.org/document/10007172/>>. 11

CHEN, T.; WU, X. A meta-heuristic approach for the constraint satisfaction problem. In: *2020 IEEE 10th International Conference on Electronics Information and Emergency Communication (ICEIEC)*. IEEE, 2021. p. 164–167. ISBN 978-1-72816-312-3 978-1-72816-313-0. Disponível em: <<https://ieeexplore.ieee.org/document/9152363/>>. 12

CHEN, X. et al. Design and application of an improved genetic algorithm to a class scheduling system. v. 16, n. 1, p. 44, 2021. ISSN 1863-0383. Disponível em: <<https://online-journals.org/index.php/i-jet/article/view/18225>>. 22

CHIARANDINI, M. et al. An effective hybrid algorithm for university course timetabling. v. 9, n. 5, p. 403–432, 2006. ISSN 1094-6136, 1099-1425. Disponível em: <<http://link.springer.com/10.1007/s10951-006-8495-8>>. 9, 15, 16, 19

DASKALAKI, S.; BIRBAS, T. Efficient solutions for a university timetabling problem through integer programming. v. 160, n. 1, p. 106–120, 2005. ISSN 03772217. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S0377221703005046>>. 9, 15, 19

DASKALAKI, S.; BIRBAS, T.; HOUSOS, E. An integer programming formulation for a case study in university timetabling. v. 153, n. 1, p. 117–135, 2004. ISSN 03772217. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S0377221703001036>>. 8, 15, 19

- DIMITSAS, A. et al. An exact based approach for the post enrollment course timetabling problem. In: *Proceedings of the 26th Pan-Hellenic Conference on Informatics*. ACM, 2022. p. 77–82. ISBN 978-1-4503-9854-1. Disponível em: <<https://dl.acm.org/doi/10.1145/3575879.3575970>>. 12
- DUNKE, F.; NICKEL, S. A matheuristic for customized multi-level multi-criteria university timetabling. 2023. ISSN 0254-5330, 1572-9338. Disponível em: <<https://link.springer.com/10.1007/s10479-023-05325-2>>. 12, 17, 33
- EKE, C. et al. A case study of model-driven engineering for automated timetabling. In: *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*. IEEE, 2019. p. 203–210. ISBN 978-1-72815-125-0. Disponível em: <<https://ieeexplore.ieee.org/document/8904643/>>. 13, 45
- GOH, S. L.; KENDALL, G.; SABAR, N. R. Monte carlo tree search in finding feasible solutions for course timetabling problem. v. 9, n. 6, p. 1936, 2019. ISSN 2460-6952, 2088-5334. Disponível em: <http://ijaseit.insightsociety.org/index.php?option=com_content&view=article&id=9&Itemid=1&article_id=10224>. 11
- GOH, S. L.; KENDALL, G.; SABAR, N. R. Simulated annealing with improved reheating and learning for the post enrolment course timetabling problem. v. 70, n. 6, p. 873–888, 2019. ISSN 0160-5682, 1476-9360. Disponível em: <<https://www.tandfonline.com/doi/full/10.1080/01605682.2018.1468862>>. 10, 15
- GOH, S. L. et al. An effective hybrid local search approach for the post enrolment course timetabling problem. v. 57, n. 4, p. 1131–1163, 2020. ISSN 0030-3887, 0975-0320. Disponível em: <<https://link.springer.com/10.1007/s12597-020-00444-x>>. 11, 16
- GOZALI, A. A. et al. Solving university course timetabling problem using localized island model genetic algorithm with dual dynamic migration policy. v. 15, n. 3, p. 389–400, 2020. ISSN 1931-4973, 1931-4981. Disponível em: <<https://onlinelibrary.wiley.com/doi/10.1002/tee.23067>>. 11
- GüLCü, A.; AKKAN, C. Robust university course timetabling problem subject to single and multiple disruptions. v. 283, n. 2, p. 630–646, 2020. ISSN 03772217. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S037722171930935X>>. 11
- HAFSA, M. et al. Solving a multiobjective professional timetabling problem using evolutionary algorithms at mandarine academy. p. itor.13276, 2023. ISSN 0969-6016, 1475-3995. Disponível em: <<https://onlinelibrary.wiley.com/doi/10.1111/itor.13276>>. 13
- HOUHAMDI, Z. et al. A multi-agent system for course timetable generation. v. 8, n. 1, 2019. 1, 12, 19, 21, 44
- JAMILI, A.; HAMID, M.; GHAROUN, H. Developing a comprehensive and multi-objective mathematical model for university course timetabling problem: A real case study. 2018. 11

JUNE, T. L. et al. Sequential constructive algorithm incorporate with fuzzy logic for solving real world course timetabling problem. In: ALFRED, R. et al. (Ed.). *Computational Science and Technology*. Springer Singapore, 2020. v. 603, p. 257–267. ISBN 9789811500572 9789811500589. Series Title: Lecture Notes in Electrical Engineering. Disponível em: <http://link.springer.com/10.1007/978-981-15-0058-9_25>. 12, 17

JUNN, K. Y. et al. A formal model of multi-agent system for university course timetabling problems. In: ALFRED, R. et al. (Ed.). *Computational Science and Technology*. Springer Singapore, 2019. v. 481, p. 215–225. ISBN 9789811326219 9789811326226. Series Title: Lecture Notes in Electrical Engineering. Disponível em: <http://link.springer.com/10.1007/978-981-13-2622-6_22>. 11

KITCHENHAM, B.; CHARTERS, S. *Guidelines for performing systematic literature reviews in software engineering*. [S.l.], 2007. Technical report. 2, 4, 5

LEMOS, A.; MONTEIRO, P. T.; LYNCE, I. Minimal perturbation in university timetabling with maximum satisfiability. In: HEBRARD, E.; MUSLIU, N. (Ed.). *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*. Springer International Publishing, 2020. v. 12296, p. 317–333. ISBN 978-3-030-58941-7 978-3-030-58942-4. Series Title: Lecture Notes in Computer Science. Disponível em: <http://link.springer.com/10.1007/978-3-030-58942-4_21>. 12

LEWIS, R. A survey of metaheuristic-based techniques for university timetabling problems. v. 30, n. 1, p. 167–190, 2007. ISSN 0171-6468, 1436-6304. Disponível em: <<http://link.springer.com/10.1007/s00291-007-0097-0>>. 1, 9, 20

LIAO, S.-H. Expert system methodologies and applications—a decade review from 1995 to 2004. *Expert systems with applications*, Elsevier, v. 28, n. 1, p. 93–103, 2005. 17

Lü, Z.; HAO, J.-K. Adaptive tabu search for course timetabling. v. 200, n. 1, p. 235–244, 2010. ISSN 03772217. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S0377221708010394>>. 9, 16, 19

MAURITSIUS, T. et al. Novel local searches for finding feasible solutions in educational timetabling problem. In: *2017 5th International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME)*. IEEE, 2017. p. 270–275. ISBN 978-1-5386-3455-4. Disponível em: <<https://ieeexplore.ieee.org/document/8537723/>>. 12

McCollum, B. et al. Setting the research agenda in automated timetabling: The second international timetabling competition. v. 22, n. 1, p. 120–130, 2010. ISSN 1091-9856, 1526-5528. Disponível em: <<https://pubsonline.informs.org/doi/10.1287/ijoc.1090.0320>>. xii, 9, 20, 40, 41, 42

Mei Ching Chen et al. A hybrid of heuristic orderings and variable neighbourhood descent for a real life university course timetabling problem. v. 6, n. 5, 2021. ISSN 2077-7973. 13

METAXIOTIS, K. et al. Decision support through knowledge management: the role of the artificial intelligence. *International Journal of Computer Applications in Technology*, v. 19, n. 2, p. 101–106, 2004. Disponível em: <<https://www.inderscienceonline.com/doi/abs/10.1504/IJCAT.2004.003641>>. 17

MUKLASON, A.; IRIANTI, R. G.; MAROM, A. Automated course timetabling optimization using tabu-variable neighborhood search based hyper-heuristic algorithm. v. 161, p. 656–664, 2019. ISSN 18770509. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S1877050919318800>>. 10

MUKLASON, A. et al. Flexible automated course timetabling system with lecturer preferences using hyper-heuristic algorithm. In: *7th International Conference on Sustainable Information Engineering and Technology 2022*. ACM, 2022. p. 258–262. ISBN 978-1-4503-9711-7. Disponível em: <<https://dl.acm.org/doi/10.1145/3568231.3568273>>. 12

NGUYEN, V. D.; NGUYEN, T. An SHO-based approach to timetable scheduling: a case study. v. 5, n. 4, p. 421–439, 2021. ISSN 2475-1839, 2475-1847. Disponível em: <<https://www.tandfonline.com/doi/full/10.1080/24751839.2021.1935644>>. 13

NUGROHO, M. A.; HERMAWAN, G. Solving university course timetabling problem using memetic algorithms and rule-based approaches. v. 407, p. 012012, 2018. ISSN 1757-899X. Disponível em: <<https://iopscience.iop.org/article/10.1088/1757-899X/407/1/012012>>. 14, 17

PILLAY, N.; ÖZCAN, E. Automated generation of constructive ordering heuristics for educational timetabling. v. 275, n. 1, p. 181–208, 2019. ISSN 0254-5330, 1572-9338. Disponível em: <<http://link.springer.com/10.1007/s10479-017-2625-x>>. 10, 14, 19

QU, R.; BURKE, E. K. Hybridizations within a graph-based hyper-heuristic framework for university timetabling problems. v. 60, n. 9, p. 1273–1285, 2009. ISSN 0160-5682, 1476-9360. Disponível em: <<https://www.tandfonline.com/doi/full/10.1057/jors.2008.102>>. 9, 14, 16, 19

RAPPOS, E. et al. A mixed-integer programming approach for solving university course timetabling problems. v. 25, n. 4, p. 391–404, 2022. ISSN 1094-6136, 1099-1425. Disponível em: <<https://link.springer.com/10.1007/s10951-021-00715-5>>. 11, 15, 19

REN, X.; LI, C. Computer intelligent course scheduling system based on deep learning. In: *2022 International Conference on Knowledge Engineering and Communication Systems (ICKES)*. IEEE, 2022. p. 1–5. ISBN 978-1-66545-637-1. Disponível em: <<https://ieeexplore.ieee.org/document/10060177/>>. 12, 17, 19

RIBEIRO, D.; TEIXEIRA, A. *Plano Orientador da Universidade de Brasília*. Brasília, DF, Brazil: Editora UnB, 1962. Documento institucional da Universidade de Brasília. 25

ROSSI-DORIA, O. et al. A comparison of the performance of different metaheuristics on the timetabling problem. In: BURKE, E.; CAUSMAECKER, P. D. (Ed.). *Practice and Theory of Automated Timetabling IV*. Springer Berlin Heidelberg, 2003. v. 2740, p. 329–351. ISBN 978-3-540-40699-0 978-3-540-45157-0. Series Title: Lecture Notes in Computer

Science. Disponível em: <http://link.springer.com/10.1007/978-3-540-45157-0_22>. 8, 40

SHUAI, C.-j. Design of automatic course arrangement system for electronic engineering teaching based on monte carlo genetic algorithm. v. 2021, p. 1–11, 2021. ISSN 1939-0122, 1939-0114. Disponível em: <<https://www.hindawi.com/journals/scn/2021/3564722/>>. 14

SIDDIQUI, A. W.; RAZA, S. A.; TARIQ, Z. M. A web-based group decision support system for academic term preparation. v. 114, p. 1–17, 2018. ISSN 01679236. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S016792361830126X>>. 1, 11, 19, 20, 35, 36, 37

SIEW, S. K. E. et al. Grouping and heuristics for a multi-stage class timetabling system. In: *2022 International Conference on Digital Transformation and Intelligence (ICDI)*. IEEE, 2022. p. 111–116. ISBN 9798350396980 9798350397000. Disponível em: <<https://ieeexplore.ieee.org/document/10007351/>>. 11, 37

SOCHA, K.; KNOWLES, J.; SAMPELS, M. A max-min ant system for the university course timetabling problem. In: SPRINGER. *Ant Algorithms: Third International Workshop, ANTS 2002 Brussels, Belgium, September 12–14, 2002 Proceedings 3*. [S.l.], 2002. p. 1–13. 8

SOCHA, K.; SAMPELS, M.; MANFRIN, M. Ant algorithms for the university course timetabling problem with regard to the state-of-the-art. In: CAGNONI, S. et al. (Ed.). *Applications of Evolutionary Computing*. Springer Berlin Heidelberg, 2003. v. 2611, p. 334–345. ISBN 978-3-540-00976-4 978-3-540-36605-8. Series Title: Lecture Notes in Computer Science. Disponível em: <http://link.springer.com/10.1007/3-540-36605-9_31>. 8

SONG, T. et al. An iterated local search algorithm for the university course timetabling problem. v. 68, p. 597–608, 2018. ISSN 15684946. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S1568494618302254>>. 10, 15, 19, 33

TAVAKOLI, M. M. et al. Proposing a novel heuristic algorithm for university course timetabling problem with the quality of courses rendered approach; a case study. v. 59, n. 5, p. 3355–3367, 2020. ISSN 11100168. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S111001682030209X>>. 11

THEPPHAKORN, T.; PONGCHAROEN, P. Performance improvement strategies on cuckoo search algorithms for solving the university course timetabling problem. v. 161, p. 113732, 2020. ISSN 09574174. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S095741742030556X>>. 11

THEPPHAKORN, T.; PONGCHAROEN, P. Modified and hybridised bi-objective firefly algorithms for university course scheduling. v. 27, n. 14, p. 9735–9772, 2023. ISSN 1432-7643, 1433-7479. Disponível em: <<https://link.springer.com/10.1007/s00500-022-07810-5>>. 12

THEPPHAKORN, T.; SOONCHAROEN, S.; PONGCHAROEN, P. Academic operating costs optimisation using hybrid MCPSO based course timetabling tool. In: CHEUNG, S. K. S. et al. (Ed.). *Blended Learning. Education in a Smart Learning Environment*. Springer International Publishing, 2020. v. 12218, p. 338–350. ISBN 978-3-030-51967-4 978-3-030-51968-1. Series Title: Lecture Notes in Computer Science. Disponível em: <http://link.springer.com/10.1007/978-3-030-51968-1_28>. 13

THEPPHAKORN, T.; SOONCHAROEN, S.; PONGCHAROEN, P. Particle swarm optimisation variants and its hybridisation ratios for generating cost-effective educational course timetables. v. 2, n. 4, p. 264, 2021. ISSN 2662-995X, 2661-8907. Disponível em: <<https://link.springer.com/10.1007/s42979-021-00652-2>>. 14, 16

UFRN, P. *Portal de cooperação*. UFRN, 2023. Acessado em: 27/09/2023. Disponível em: <<https://portalcooperacao.info.ufrn.br/>>. 29, 31

UFRN, S. *Secretaria de Tecnologia da Informação, Institucional*. UnB, 2023. Acessado em: 27/09/2023. Disponível em: <<https://docs.info.ufrn.br/doku.php?id=suporte>>. 29, 30

UNB. *Projeto Político-Pedagógico Institucional da Universidade de Brasília*. UnB, 2018. Acessado em: 25/10/2023. Disponível em: <<https://dpo.unb.br/images/PPPIVersaoAprovadaConsuni.pdf>>. 25

UNB. *História da Universidade de Brasília*. UnB, 2023. Acessado em: 25/09/2023. Disponível em: <<https://www.unb.br/a-unb/historia>>. 24

UNB. *Plano de Desenvolvimento Institucional*. UnB, 2023. Acessado em: 25/09/2023. Disponível em: <<https://planejamento.unb.br/>>. 1, 26, 27

UNB. *Regimento Interno da Prefeitura da UnB*. UnB, 2023. Acessado em: 25/10/2023. Disponível em: <https://prefeitura.unb.br/images/Ato_da_Reitoria.pdf>. 28

UNB. *Sistema de ensalamento UnB*. UnB, 2023. Acessado em: 25/09/2023. Disponível em: <<https://prefeitura.unb.br/index.php/noticias/1002-alocacao-das-salas-de-aula-de-uso-comum-para-2022-2>>. 30

UNB, F. *Faculdade de Ceilândia*. UnB, 2023. Acessado em: 25/09/2023. Disponível em: <<http://fce.unb.br/sobre-a-fce/estrutura>>. 26

UNB, F. *Faculdade do Gama*. UnB, 2023. Acessado em: 25/09/2023. Disponível em: <<https://fga.unb.br/unb-gama/sobre>>. 26

UNB, F. *Faculdade UnB de Planaltina*. UnB, 2023. Acessado em: 25/09/2023. Disponível em: <<http://fup.unb.br/apresentacao/>>. 26

UNB, P. *Prefeitura da UnB*. UnB, 2023. Acessado em: 27/09/2023. Disponível em: <<https://prefeitura.unb.br/>>. 28

UNB, P. *Sistema Integrado de Gestão da UnB*. UnB, 2023. Acessado em: 27/09/2023. Disponível em: <<https://portalsig.unb.br/>>. 29

UNB, S. *Secretaria de Tecnologia da Informação*. UnB, 2023. Acessado em: 27/09/2023. Disponível em: <<https://sti.unb.br/>>. 28

UNB, S. *Secretaria de Tecnologia da Informação, Institucional*. UnB, 2023. Acessado em: 27/09/2023. Disponível em: <<https://sti.unb.br/ins-sti>>. 28

VIANNA, D. S. et al. HYBRID VNS-TS HEURISTICS FOR UNIVERSITY COURSE TIMETABLING PROBLEM. v. 17, n. 2, p. 1–20, 2020. ISSN 2237-8960. Disponível em: <<https://bjopm.org.br/bjopm/article/view/683>>. 1, 12

WAHID, J. et al. Constructing population of initial university timetable: design and analysis. v. 15, n. 2, p. 1109, 2019. ISSN 2502-4760, 2502-4752. Disponível em: <<http://ijeecs.iaescore.com/index.php/IJEECS/article/view/19125>>. 12

WIJAYA, M. C. Two stages best first search algorithm using hard and soft constraints heuristic for course timetabling. v. 34, n. 4, p. 413–418, 2020. ISSN 0992499X, 19585748. Disponível em: <<http://www.iieta.org/journals/ria/paper/10.18280/ria.340405>>. 13

ZARANDI, M. H. F. et al. A state of the art review of intelligent scheduling. *Artificial Intelligence Review*, v. 53, n. 1, p. 501–593, 2020. ISSN 1573-7462. ID: Fazel Zarandi2020. Disponível em: <<https://doi.org/10.1007/s10462-018-9667-6>>. 10, 16, 17

ZHANG, S.; WANG, X. Intelligent course scheduling system based on case-based reasoning. v. 1920, n. 1, p. 012086, 2021. ISSN 1742-6588, 1742-6596. Disponível em: <<https://iopscience.iop.org/article/10.1088/1742-6596/1920/1/012086>>. 13, 18

ZHENG, H. et al. Course scheduling algorithm based on improved binary cuckoo search. v. 78, n. 9, p. 11895–11920, 2022. ISSN 0920-8542, 1573-0484. Disponível em: <<https://link.springer.com/10.1007/s11227-022-04341-6>>. 13

Apêndice A

Constraints Identificadas nos Artigos Pesquisados

Este apêndice apresenta as *constraints* identificadas durante a investigação realizada nesta pesquisa. Para tanto, a Tabela A.1 é organizada conforme a descrição a seguir:

- Id.: Identificador único de cada registro.
- Artigo: Referência ao artigo onde a *constraint* foi identificada. Estes artigos podem ser encontrados na Tabela 2.4.
- Constraint: Descrição da restrição em questão.
- Tipo: Classificação da *constraint* como *Hard* ou *Soft*, indicando se a restrição é rígida ou suave, respectivamente.
- UnB: Significa que a *constraint* é aderente ao contexto da UnB (Em andamento).
- Modelo: Refere-se à inclusão da *constraint* no modelo proposto. Este campo indica a prioridade para sua implementação.
- Categoria: Categoria à qual a *constraint* pertence. Foram criadas categorias para agrupamento das constraints que são semelhantes ou impactam um grupo específico de atores (Em andamento).

Esta tabela serve como um complemento ao texto principal e fornece uma visão detalhada das *constraints* identificadas na literatura sobre o problema abordado nesta pesquisa. A tabela pode ser consultada para obter informações mais específicas sobre cada *constraint*, bem como para identificar os elementos pertinentes ao contexto da UnB e o que se deseja implementar no modelo.

Tabela A.1: Constraints identificadas nos artigos pesquisados

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
1	[1]	Todos os eventos devem ser atribuídos a um determinado período e uma sala adequada.	Soft			Alocação de Recursos
2	[1]	Sem conflito entre estudantes, ou seja, nenhum estudante pode estar envolvido em dois ou mais eventos durante o mesmo período.	Hard			Conflito de Estudantes
3	[1]	Sem conflito de salas, ou seja, apenas um evento é alocado em qualquer sala em qualquer período.	Hard	x	1	Conflito de Salas
4	[3]	Todas as aulas de um curso devem ser agendadas e devem ser atribuídas a períodos distintos.	Hard	x		Alocação de Recursos
5	[3]	Duas aulas não podem ocorrer na mesma sala no mesmo período.	Hard	x		Conflito de Salas
6	[3]	Aulas de cursos no mesmo currículo ou ministradas pelo mesmo professor devem ser todas agendadas em períodos diferentes.	Hard	x	2	Conflito de Recursos
7	[3]	Se o professor do curso não estiver disponível para ministrar o curso em um determinado período, então nenhuma aula do curso pode ser agendada naquele período.	Hard	x	3	Disponibilidade de Recursos
8	[3]	Para cada aula, o número de estudantes que frequentam o curso deve ser menor ou igual ao número de assentos de todas as salas que hospedam suas aulas.	Soft			Capacidade da Sala

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
9	[3]	As aulas de cada curso devem ser distribuídas no mínimo número de dias dado.	Soft			Mínimo de dias
10	[3]	As aulas pertencentes a um currículo devem ser adjacentes entre si	Soft			Compactação de Currículo
11	[3]	Todas as aulas de um curso devem ser ministradas na mesma sala.	Soft	x	4	Consistência de Espaço
12	[5]	Nenhum estudante pode assistir a mais de uma aula ao mesmo tempo.	Hard			Conflito de Estudantes
13	[5]	Nenhum professor pode ministrar mais de uma aula ao mesmo tempo.	Hard	x		Conflito de Recursos
14	[5]	Nenhuma sala pode ser atribuída a uma aula com mais estudantes do que sua capacidade.	Hard			Otimização de Espaço
15	[5]	Algumas disciplinas são co-requisitos para outras; portanto, devem ocorrer em horários diferentes.	Hard			Conflito de Currículo
16	[5]	Para cada estudante, as disciplinas obrigatórias do currículo principal e secundário devem ser agendados em horários não sobrepostos.	Soft			Conflito de Currículo
17	[5]	Um estudante não deve ter apenas uma única aula em um dia.	Soft			Otimização de Horário
18	[5]	Cada docente deve ministrar no máximo três aulas em um dia.	Soft			Otimização de Carga Horária
19	[6]	Cada evento deve ser agendado exatamente uma vez.	Hard	x		Alocação de Eventos

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
20	[6]	Não há dois eventos com um estudante em comum agendados nos mesmos horários.	Hard			Conflito de Estudantes
21	[6]	O número de estudantes não excede a capacidade da sala.	Hard			Capacidade da Sala
22	[6]	Não há dois eventos diferentes agendados na mesma sala.	Hard	x		Conflito de Salas
23	[6]	Os estudantes não devem participar de mais de três eventos por dia.	Soft			Otimização de Horário
24	[6]	Os estudantes não devem participar de apenas um evento por dia.	Soft			Otimização de Horário
25	[6]	Os estudantes não devem participar do último evento do dia.	Soft			Otimização de Horário
26	[7]	Nenhum estudante deve participar de mais de um evento ao mesmo tempo.	Hard			Conflito de Estudantes
27	[7]	Cada evento é atribuído a uma sala com assentos suficientes para todos os estudantes participantes e todas as características necessárias.	Hard			Otimização de Espaço
28	[7]	Apenas um evento por sala em qualquer intervalo de tempo.	Hard			Conflito de Salas
29	[7]	Os eventos são atribuídos a intervalos de tempo designados.	Hard			Alocação de Eventos
30	[7]	Quando especificado, os eventos devem ser agendados na ordem correta.	Hard			Ordem de Eventos
31	[7]	Penalidade para estudantes com um evento em um dia.	Soft			Otimização de Horário
32	[7]	Penalidade para estudantes com três ou mais eventos consecutivos.	Soft			Otimização de Horário

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
33	[7]	Penalidade para estudantes com um evento no último intervalo de tempo do dia.	Soft			Otimização de Horário
34	[8]	O número de aulas especificado para cada curso deve ser agendado.	Hard			Alocação de Aulas
35	[8]	Aulas de cursos em um currículo devem ser agendadas em períodos diferentes.	Hard			Conflito de Recursos
36	[8]	Cada sala deve ser agendada apenas uma vez em um período.	Hard			Conflito de Salas
37	[8]	Cada professor deve ser agendado uma vez em um período.	Hard			Conflito de Recursos
38	[8]	Capacidade da sala não deve ser excedida	Soft			Capacidade da Sala
39	[8]	Para alguns cursos, um número mínimo de dias úteis é especificado.	Soft			Mínimo de dias
40	[8]	Aulas de cursos pertencentes ao mesmo currículo devem ser agendadas adjacentes entre si em um dia.	Soft	x	5	Compactação de Currículo
41	[8]	Todas as aulas de um curso devem ser agendadas no mesmo local.	Soft			Consistência de Espaço.
42	[9]	Os estudantes só podem frequentar um curso de cada vez.	Hard			Conflito de Estudantes
43	[9]	As salas devem ter características específicas exigidas pelos cursos.	Hard			Especificações de Sala
44	[9]	As salas devem ter capacidade suficiente para os estudantes que frequentam os cursos.	Hard			Capacidade da Sala
45	[9]	Apenas uma disciplina é permitida em cada sala de cada vez.	Hard			Conflito de Salas

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
46	[9]	Os cursos podem ter que ser agendados para um horário predefinido.	Hard			Alocação de Eventos
47	[9]	Disciplinas devem aparecer em uma sequencia específica	Hard			Ordem de eventos
48	[9]	Os estudantes devem ter mais de uma aula em um dia.	Soft			Otimização de Horário
49	[9]	Os estudantes devem ter menos de três aulas consecutivas.	Soft			Otimização de Horário
50	[9]	Os estudantes não devem ter aula no último intervalo de tempo do dia.	Soft			Otimização de Horário
51	[10]	Todas as aulas/laboratórios (elementos) necessários para cada curso devem ser agendados e atribuídos a períodos distintos.	Hard			Alocação de Aulas
52	[10]	Estudantes e professores só podem frequentar uma aula de cada vez.	Hard			Conflito de Estudantes e Professores
53	[10]	Apenas uma aula pode ocorrer em uma sala em um determinado momento.	Hard			Conflito de Espaço
54	[10]	Professores e estudantes devem estar disponíveis para que uma aula seja agendada.	Hard			Disponibilidade de Recursos
55	[10]	Todos os cursos devem ser atribuídos a salas adequadas de acordo com seus requisitos, incluindo localização do prédio, instalações da sala e tipos de sala.	Hard			Especificações de Sala
56	[10]	Todas as aulas dentro de um curso que exigem períodos consecutivos devem ser obedecidas.	Hard			Ordem de Eventos

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
57	[10]	Todos os cursos devem ser agendados no tipo apropriado de sala para evitar custos operacionais ou de aluguel desnecessários.	Soft			Especificações do Espaço
58	[10]	Os cursos ministrados pelo(s) professor(es) dado(s) devem ser atribuídos em seus dias e períodos disponíveis ou preferidos para economizar nos custos de contratação ou docência.	Soft			Disponibilidade de Recursos
59	[10]	As salas de aula devem ser agendadas em períodos consecutivos de um dia de trabalho para reduzir o número de vezes que é necessário limpar ou preparar as salas após o uso.	Soft			Ordem de Eventos
60	[11]	Todas as aulas/laboratórios (elementos) necessários para cada curso devem ser agendados e atribuídos a períodos distintos	Hard			Alocação de aulas
61	[11]	Aulas de cursos no mesmo currículo ou ministradas pelo mesmo professor devem ser todas agendadas em períodos diferentes	Hard			Conflito de Recursos
62	[11]	Se o professor do curso não estiver disponível para ministrar o curso em um determinado período, então nenhuma aula do curso pode ser agendada naquele período.	Hard			Disponibilidade de Recursos
63	[11]	Duas aulas não podem ser alocadas no mesmo espaço no mesmo período	Hard			Conflito de Salas

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
64	[11]	Para cada aula, o número de estudantes que frequentam o curso deve ser menor ou igual ao número de assentos de todas as salas que hospedam suas aulas.	Soft			Capacidade da Sala
65	[11]	As aulas de cada curso devem ser distribuídas no mínimo número de dias dado.	Soft			Mínimo de dias
66	[11]	As aulas pertencentes a um currículo devem ser adjacentes entre si	Soft			Compactação de Currículo
67	[11]	Todas as aulas de um curso devem ser ministradas na mesma sala.	Soft			Consistência de Espaço
68	[12]	Todas as aulas de cada curso devem ser agendadas e devem ser atribuídas a intervalos de tempo distintos.	Hard			Alocação de Aulas
69	[12]	Aulas de cursos no mesmo currículo ou ministradas pelo mesmo professor devem ser todas agendadas em intervalos de tempo diferentes.	Hard			Conflito de Recursos
70	[12]	Duas aulas não podem ocorrer na mesma sala no mesmo intervalo de tempo.	Hard			Conflito de Salas
71	[12]	Se o professor do curso estiver indisponível para ministrar esse curso em um determinado intervalo de tempo, então nenhuma aula do curso pode ser agendada nesse intervalo de tempo.	Hard			Disponibilidade de Recursos

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
72	[12]	Para cada aula, o número de estudantes que frequentam o curso deve ser menor ou igual ao número de assentos de todas as salas que hospedam suas aulas.	Soft			Otimização de Espaço
73	[12]	As aulas de cada curso devem ser distribuídas em um número mínimo de dias.	Soft			Mínimo de Dias
74	[12]	Aulas pertencentes a um currículo devem ser adjacentes entre si em intervalos de tempo consecutivos.	Soft			Otimização de Horário
75	[12]	Aulas pertencentes a um currículo não devem ter janelas de tempo entre elas.	Soft			Otimização de Horário
76	[12]	Todas as aulas de um curso devem ser dadas na mesma sala.	Soft			Consistência de Espaço
77	[12]	Para cada currículo, o número de aulas diárias deve estar dentro de um determinado intervalo.	Soft			Carga Horária de Estudantes
78	[12]	Os estudantes devem ter tempo para se deslocar de um prédio para outro entre duas aulas.	Soft			Otimização de Deslocamento
79	[12]	Algumas salas podem não ser adequadas para um determinado curso devido à ausência de equipamento necessário.	Soft			Adequação de Sala
80	[12]	Alguns cursos exigem que as aulas no mesmo dia sejam agrupadas.	Soft			Agrupamento de Aulas
81	[13]	Um professor ministra no máximo um curso em um intervalo de tempo.	Hard			Conflito de Professores
82	[13]	Apenas um curso pode ser agendado para um local em um intervalo de tempo.	Hard			Conflito de Espaço

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
83	[13]	A capacidade de uma sala de aula deve ser suficiente para acomodar o número de estudantes matriculados no curso.	Hard			Otimização de Espaço
84	[13]	Todos os cursos devem ser agendados.	Hard			Alocação de Aulas
85	[13]	Intervalos de tempo bloqueados são proibidos para alocação de cursos.	Hard			Restrições de Tempo
86	[13]	Um estudante só pode frequentar um curso em um intervalo de tempo.	Hard			Conflito de Estudantes
87	[14]	Todas as seções femininas/masculinas do curso podem ser atribuídas exatamente a uma sala em um prédio permitido, sessão e tipo de período específicos.	Hard			Alocação de Salas
88	[14]	No máximo uma seção feminina/-masculina seja atribuída a qualquer sala elegível de cada vez, durante qualquer sessão e tipo de período.	Hard			Conflito de Espaço
89	[14]	Apenas uma seção de curso pode ser agendada em uma sala entre os tipos de períodos 2 e 3 que se sobrepõem às segundas-feiras.	Hard			Conflito de Espaço
90	[14]	Cada seção seja atribuída a um instrutor que ensine aquele curso.	Hard			Alocação de Professores
91	[14]	Todos os instrutores que ensinam um curso de um departamento específico não recebam mais seções femininas/masculinas do que as atribuídas.	Hard			Limitação de Carga Docente

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
92	[14]	A um instrutor seja atribuído no máximo uma seção durante qualquer sessão para um tipo de período específico.	Hard			Limitação de Carga Docente
93	[14]	limita a atribuição de sessões para tipos de períodos que se sobrepõem.	Hard			Limitação de Carga Docente
94	[14]	Cada instrutor ensine um máximo de seções consecutivas durante um tipo de período específico.	Hard			Limitação de Carga Docente
95	[14]	Garante que não haja sessões de aula paralelas para o mesmo grupo de coorte durante qualquer sessão pertencente a um tipo de período específico.	Hard			Conflito de Estudantes
96	[14]	Impõe um máximo de sessões contínuas em um dia para um grupo de coorte específico.	Hard			Limitação de Carga Estudantil
97	[14]	Garante a atribuição de grupos em salas com capacidades apropriadas.	Hard			Otimização de Espaço
98	[15]	Cada curso deve ser atribuído a uma turma	Hard			Alocação de Aulas
99	[15]	Garante que o determinado curso não seja atribuído à turma que não é elegível para ele de acordo com sua capacidade e equipamento.	Hard			Adequação de Sala
100	[15]	Se determinado curso for atribuído à determinada turma, ele deve ocupar um determinado número de intervalos de tempo na semana.	Hard			Otimização de Horário

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
101	[15]	Garante que cada curso, dependendo do número de suas unidades, seja atribuído ao intervalo de tempo adequado.	Hard			Otimização de Horário
102	[15]	Verifica a disponibilidade dos instrutores em dias-intervalos de tempo e impede a atribuição de seus cursos aos dias-intervalos de tempo em que eles não estão disponíveis.	Hard			Disponibilidade de Professores
103	[15]	Garante que os instrutores não ensinem mais do que um certo limite por dia.	Hard			Limitação de Carga Docente
104	[15]	Impede a sobreposição do intervalo de tempo atribuído aos cursos de um instrutor em um dia.	Hard			Conflito de Professores
105	[15]	No máximo um curso pode ser atribuído a um intervalo de tempo em uma turma e dia.	Hard			Conflito de Espaço
106	[15]	Impede a sobreposição dos cursos oferecidos para os estudantes que ingressam na universidade no mesmo semestre.	Hard			Conflito de Estudantes
107	[15]	Garante que a distância mínima (em dias) seja estabelecida entre as sessões do curso que precisam de mais de um intervalo de tempo na semana.	Hard			Otimização de Horário
108	[15]	cada instrutor não tem permissão para ensinar mais do que um certo limite em um dia consecutivamente.	Hard			Limitação de Carga Docente

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
109	[16]	As aulas devem ser agendadas em horários diferentes tanto para professores quanto para alunos.	Hard			Conflito de Recursos
110	[16]	Uma sala só pode ser usada para uma aula no mesmo intervalo de tempo.	Hard			Utilização de Espaço
111	[16]	O número de alunos em uma aula não deve exceder a capacidade da sala.	Hard			Otimização de Espaço
112	[16]	Atribuição de sala: a capacidade e os tipos de sala devem ser apropriados para as aulas atribuídas.	Hard			Adequação de Sala
113	[16]	Preferência de período de tempo: minimizar o uso de intervalos de tempo no final da tarde (15h - 18h).	Soft			Preferência de Horário
114	[16]	Reduzir as horas diárias de aula: reservar o período da tarde para fins de consulta e atividades curriculares.	Soft			Otimização de Horário Docente
115	[16]	Utilização da sala: reduzir o desperdício de espaço da sala, atribuindo aulas em salas de capacidade fixa de tal forma que o tamanho da turma em cada sala seja, no máximo, a capacidade.	Soft			Otimização de Espaço
116	[17]	Não deve haver conflito entre professores. Nenhum professor deve ser alocado em salas diferentes ao mesmo tempo.	Hard			Conflito de Professores
117	[17]	Não deve haver conflito entre aulas. Nenhum pacote deve ser definido para eventos diferentes ao mesmo tempo.	Hard			Conflito de Aulas

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
118	[17]	Todo evento deve ser agendado em uma sala com capacidade adequada.	Hard			Otimização de Espaço
119	[17]	Professores, reitores e diretores devem ser agendados dentro de suas restrições de tempo.	Hard			Disponibilidade de Professores
120	[17]	Alguns professores devem ser agendados de acordo com suas preferências de horário.	Hard			Preferência de Horário de Professores
121	[17]	A carga de ensino de um professor deve ser definida para um máximo de LC + eventos em um dia.	Soft			Distribuição de Carga Docente
122	[17]	Um evento de aula deve ser definido com um mínimo de CC - dias de intervalo em uma semana.	Soft			Distribuição de Eventos de Aula
123	[17]	O intervalo de tempo entre dois eventos para ensino em grupo deve ser menor do que sua restrição de tempo mínima.	Soft			Coordenação de Horários em Grupo
124	[17]	O evento de ensino de um professor deve ser definido em seu horário preferido.	Soft			Preferência de Horário de Professores
125	[17]	O intervalo de tempo entre dois eventos de um professor não deve ser menor do que o intervalo de restrição de tempo mínima.	Soft			Intervalo entre Eventos de Professores
126	[17]	Os eventos de um estudante devem ser definidos para o máximo de SC + eventos em um dia.	Soft			Distribuição de Carga Estudantil
127	[17]	Minimizar o número de estudantes que foram alocados em salas ou aulas diferentes que ocorrem ao mesmo tempo.	Soft			Minimização de Conflito de Estudantes

Continua na próxima página

Tabela A.1 – Continuação da página anterior

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
128	[18]	Cada turma c deve ter uma atribuição de tempo t .	Hard			Alocação de Tempo
129	[18]	Cada turma c deve ser atribuída a uma sala r , quando aplicável.	Hard			Alocação de Sala
130	[18]	Cada estudante s deve frequentar exatamente uma turma c de cada subparte P_f da configuração de curso selecionada f para cada curso o que ele deve frequentar.	Hard			Alocação de Estudantes
131	[18]	Se uma turma c tem uma turma pai c' definida, sempre que a turma c é atribuída a um estudante s , então a turma pai c' também deve ser atribuída a esse estudante.	Hard			Hierarquia de Turmas
132	[18]	Cada estudante deve ser atribuído a uma configuração de curso f pertencente a F_0 , para cada curso o que ele frequenta.	Hard			Configuração de Curso
133	[18]	A capacidade M_c de cada turma em termos do número de estudantes deve ser satisfeita.	Hard			Capacidade de Turma
134	[18]	Uma sala não pode ser usada quando está indisponível.	Hard			Disponibilidade de Sala
135	[18]	Duas turmas não podem acontecer ao mesmo tempo na mesma sala.	Hard			Conflito de Sala
136	[18]	Para cada par de turmas de uma restrição de distribuição rígida, proibir combinações de tempo e atribuições de sala que resultem em uma restrição violada.	Hard			Restrições de Distribuição
137	[19]	Um estudante só pode frequentar um curso de cada vez.	Hard			Conflito de Estudantes

Continua na próxima página

Tabela A.1 – Continuação da página anterior

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
138	[19]	Uma sala deve cumprir os recursos exigidos por um curso.	Hard			Requisitos de Sala
139	[19]	Uma sala deve fornecer assentos suficientes para um curso.	Hard			Capacidade de Sala
140	[19]	Apenas um curso é permitido em cada intervalo de tempo e sala.	Hard			Conflito de Sala e Tempo
141	[19]	Um curso deve ser atribuído a um dos intervalos de tempo predefinidos.	Hard			Alocação de Tempo Predefinida
142	[19]	Um curso pode ser necessário para aparecer em uma determinada sequência.	Hard			Sequência de Cursos
143	[20]	Agendamento para professores de acordo com suas horas de presença	Hard			Disponibilidade de Professores
144	[20]	Um professor não pode ministrar mais de um curso durante um intervalo de tempo.	Hard			Conflito de Professores
145	[20]	Não é viável para qualquer professor ministrar mais do que as horas especificadas/dia.	Hard			Carga Horária de Professores
146	[20]	Todos os grupos previstos no programa de cada disciplina devem ser agendados no cronograma.	Hard			Cobertura Curricular
147	[20]	Não mais do que um curso deve ser atribuído a uma sala de aula em um intervalo de tempo.	Hard			Conflito de Sala
148	[20]	Os cursos são atribuídos a salas de aula que têm capacidade igual ou superior para a capacidade dos cursos de estudantes.	Hard			Capacidade de Sala
149	[20]	Os professores são atribuídos a horas específicas para tutoriais ao longo da semana.	Hard			Distribuição de Tutoria

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
150	[20]	Professores de cada universidade impõem menos custos educacionais para a universidade e estão mais acessíveis aos alunos; consequentemente, neste modelo, a prioridade é dada aos professores em vez de professores em tempo parcial.	Soft			Prioridade de Professores
151	[20]	Como uma disciplina pode ser ministrada por muitos professores, a prioridade deve ser dada ao de maior qualidade.	Soft			Qualidade de Ensino
152	[20]	Para reduzir o tempo de ida e volta da universidade gasto pelo professor, neste modelo, o tempo de presença do professor é utilizado ao máximo.	Soft			Otimização de Tempo de Professores
153	[21]	O estudante não deve frequentar mais de uma aula ao mesmo tempo.	Hard			Conflito de Estudantes
154	[21]	A capacidade de uma sala deve ser maior do que o tamanho do curso matriculado pelos estudantes.	Hard			Capacidade de Sala
155	[21]	Não deve haver mais de um curso atribuído à mesma sala ao mesmo tempo.	Hard			Conflito de Sala
156	[21]	Para cada tipo diferente de curso, deve ser satisfeito o respectivo intervalo de tempo predefinido declarado pelo departamento ASD.	Hard			Alocação de Tempo Predefinida
157	[21]	O estudante não deve frequentar mais de duas aulas consecutivas.	Soft			Consecutividade de Aulas

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
158	[21]	Os assentos vazios restantes na diferença entre a capacidade da sala e o tamanho da matrícula do curso devem ser minimizados o máximo possível.	Soft			Otimização de Espaço
159	[22]	Todas as aulas de um curso devem ser agendadas e devem ser agendadas em períodos distintos.	Hard			Agendamento de Aulas
160	[22]	Um curso pode ter períodos específicos definidos como períodos indisponíveis.	Hard			Disponibilidade de Aulas
161	[22]	Dois cursos são conflitantes se forem ministrados pelo mesmo professor ou pertencerem ao mesmo currículo.	Hard			Conflito de Cursos
162	[22]	Uma sala pode acomodar apenas uma aula em qualquer período dado.	Hard			Ocupação de Sala
163	[22]	Uma sala deve ser capaz de acomodar o maior número possível de alunos ao agendar os cursos nas salas.	Soft			Capacidade de Sala
164	[22]	As aulas de um curso devem ser todas agendadas na mesma sala durante a semana.	Soft			Estabilidade de Sala
165	[22]	Para cada curso, é preferível espalhar as aulas por um número determinado de dias.	Soft			Dias de Trabalho Mínimos
166	[22]	Se duas aulas pertencem ao mesmo dia e estão em horários consecutivos, então dizemos que os períodos são adjacentes.	Soft			Aulas Isoladas

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
167	[23]	Um estudante ou instrutor não pode estar envolvido em mais de um evento no mesmo intervalo de tempo.	Hard			Conflito de Participantes
168	[23]	Todos os eventos devem ser atribuídos a uma sala admissível dentro dos intervalos de tempo dados.	Hard			Atribuição de Sala e Tempo
169	[23]	Uma sala não deve ser agendada para mais de um evento em qualquer intervalo de tempo.	Hard			Conflito de Sala
170	[24]	Um professor não pode ensinar em dois cursos diferentes ao mesmo tempo.	Hard			Conflito de Professores
171	[24]	Um professor que tem uma posição organizacional não pode ensinar em um curso ao mesmo tempo que a agenda organizacional rotineira.	Hard			Conflito de Agenda Organizacional
172	[24]	Uma sala em cada intervalo de tempo disponível só pode ser usada para um cronograma de curso.	Hard			Conflito de Sala
173	[24]	A distância de ensino de um professor de uma aula para a próxima aula que é adjacente ou mesmo sequencial pode ser minimizada.	Soft			Minimização da Distância de Ensino
174	[24]	Um intervalo de tempo que não é priorizado pode ser minimizado.	Soft			Minimização de Intervalos de Tempo Não Priorizados

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
175	[25]	Um professor ensina no máximo um curso em um intervalo de tempo.	Hard			Conflito de Professores
176	[25]	Um grupo aprende no máximo um curso em um intervalo de tempo.	Hard			Conflito de Grupos
177	[25]	Uma tarefa deve ser agendada para no máximo um intervalo de tempo por dia.	Hard			Agendamento de Tarefas
178	[25]	Uma tarefa não é agendada para dois dias consecutivos.	Hard			Agendamento Consecutivo de Tarefas
179	[25]	As horas de crédito organizadas devem ser o mais próximas possível das horas de crédito necessárias.	Hard			Otimização de Horas de Crédito
180	[25]	Uma tarefa não deve ser agendada no 3 ^o ou 4 ^o intervalo de tempo de quinta-feira.	Hard			Restrições de Tempo Específicas
181	[25]	Uma tarefa deve evitar ser organizada no último intervalo de tempo, tanto quanto possível.	Soft			Otimização de Último Intervalo de Tempo
182	[25]	Um professor deve fornecer aulas de forma tão descontínua quanto possível.	Soft			Distribuição Descontínua de Aulas
183	[25]	Um grupo deve assistir às aulas todos os dias úteis da 1 ^a à 16 ^a semana, se possível.	Soft			Frequência de Aulas de Grupo
184	[26]	O cronograma é válido para um semestre.	Hard			Validade do Cronograma
185	[26]	O número de salas de aula é fixo.	Hard			Número Fixo de Salas

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
186	[26]	As salas de aula têm assentos limitados.	Hard			Capacidade Limitada de Sala
187	[26]	Os horários são definidos pela universidade.	Hard			Definição de Horários
188	[26]	Uma sala de aula é atribuída a apenas uma aula de cada vez.	Hard			Atribuição Única de Sala
189	[26]	Recursos limitados durante um horário específico.	Hard			Limitação de Recursos
190	[26]	Estudantes e instrutores podem estar em uma sala de aula em um horário específico.	Hard			Ocupação de Sala
191	[26]	O instrutor pode preferir alguns horários.	Soft			Preferência de Horário do Instrutor
192	[26]	O instrutor pode preferir um dia de trabalho específico.	Soft			Preferência de Dia de Trabalho do Instrutor
193	[26]	Evitar lacunas entre as aulas.	Soft			Evitar Lacunas
194	[26]	Evitar agendar cursos no último horário.	Soft			Evitar Último Horário
195	[26]	Um curso pode precisar ser agendado antes/depois de outro curso.	Soft			Sequenciamento de Cursos
196	[26]	Distribuir a carga de forma uniforme ao longo da semana.	Soft			Distribuição Uniforme da Carga
197	[26]	O instrutor pode preferir uma sala de aula específica.	Soft			Preferência de Sala do Instrutor
198	[26]	Alguns alunos podem ter um dia de folga durante a semana.	Soft			Dia de Folga para Alunos

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
199	[26]	Os instrutores podem preferir dar mais de uma aula por dia.	Soft			Preferência de Múltiplas Aulas
200	[27]	Todas as aulas devem ser agendadas para um período e sala diferentes.	Hard			Agendamento de Aulas
201	[27]	Duas aulas não podem ser atribuídas à mesma sala e ao mesmo período.	Hard			Ocupação de Sala
202	[27]	Aulas no mesmo currículo ou ministradas pelo mesmo professor devem ser atribuídas a períodos separados.	Hard			Conflitos de Aulas
203	[27]	Se o professor de um curso não estiver disponível em um determinado período, então nenhuma aula do curso pode ser alocada para aquele período.	Hard			Disponibilidade do Professor
204	[27]	A quantidade de alunos para cada aula deve ser menor ou equivalente à capacidade das salas.	Soft			Capacidade da Sala
205	[27]	As aulas de cada curso devem abranger um determinado número de dias.	Soft			Dias de Trabalho Mínimos
206	[27]	Aulas de cursos do mesmo currículo devem estar em períodos contíguos.	Soft			Compactação do Currículo
207	[27]	Todas as aulas de um curso específico devem ser alocadas na mesma sala.	Soft			Estabilidade da Sala
208	[28]	Classes com mais de 2 horas podem ser alocadas em dias distintos, mas devem ser alocadas no mesmo horário durante a semana.	Soft			Horário Alocado para Aulas

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
209	[28]	As aulas não devem ser oferecidas em dias consecutivos durante a semana.	Soft			Dia Alocado para Aulas
210	[28]	Refere-se ao número máximo de horas que uma aula pode ser atribuída em um dia.	Soft			Máximo de Horas por Dia para Aula
211	[28]	As aulas, se atribuídas em um dia, devem ter um número mínimo de horas.	Soft			Mínimo de Horas por Dia para Aula
212	[28]	As aulas devem ser alocadas de acordo com os turnos disponíveis para o respectivo curso universitário.	Soft			Turno Disponível para Aula
213	[28]	Existem algumas aulas predefinidas que são agendadas em horários dados.	Hard			Horários Fixos de Aulas
214	[28]	O intervalo do almoço é das 13h às 14h.	Hard			Intervalo de Almoço
215	[28]	O horário de início das aulas é às 7h e o horário de término é às 22h.	Hard			Limites de Horários de Aulas
216	[28]	O número de salas de aula disponíveis deve ser considerado.	Hard			Salas de Aula Disponíveis
217	[28]	Aulas pertencentes a um grupo de semestre em um programa de curso não podem ser alocadas no mesmo dia e horário.	Hard			Colisão de Horários de Aulas
218	[28]	Aulas oferecidas em dias distintos devem ser oferecidas no mesmo turno.	Soft			Turno da Aula
219	[28]	Aulas com mais de 2h em um dia devem ser oferecidas continuamente.	Soft			Janelas de Tempo para Aula

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
220	[28]	As aulas devem ser alocadas considerando os horários que um professor se declara disponível.	Soft			Indisponibilidade do Professor
221	[28]	Duas ou mais aulas não podem ser atribuídas a um professor no mesmo dia e horário.	Hard			Colisão de Horários do Professor
222	[28]	Refere-se ao número máximo de dias em uma semana em que um professor tem aulas atribuídas.	Soft			Máximo de Dias para o Professor
223	[28]	Visa maximizar o número de aulas continuamente alocadas para um professor em um dia.	Soft			Janelas de Tempo do Professor
224	[28]	Um professor não pode assistir a duas aulas ao mesmo tempo.	Hard			Colisão de Aulas do Professor
225	[28]	Refere-se ao número máximo de horas que um professor ensina em um dia.	Soft			Máximo de Horas por Dia para o Professor
226	[28]	Visa maximizar o número de aulas que cada aluno deseja frequentar.	Soft			Colisão de Aulas entre Alunos
227	[29]	Cada evento deve ser agendado em um horário e uma sala.	Hard			Agendamento de Evento
228	[29]	Eventos com alunos em comum devem ser colocados em horários diferentes.	Hard			Conflito de Horários entre Eventos
229	[29]	Uma sala pode hospedar no máximo um evento em cada horário.	Hard			Ocupação da Sala
230	[29]	A capacidade e os requisitos de recursos de cada evento devem ser atendidos pela sala que eventualmente o hospedará.	Hard			Requisitos de Sala

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
231	[29]	Certos requisitos de horário podem se aplicar a eventos específicos.	Hard			Requisitos de Horário para Eventos
232	[29]	Pode haver relações de precedência entre eventos.	Hard			Precedência entre Eventos
233	[29]	Um aluno assiste a um evento no último horário do dia.	Soft			Último Horário do Dia para Aluno
234	[29]	Um aluno assiste a três (ou mais) eventos consecutivos no mesmo dia.	Soft			Eventos Consecutivos para Aluno
235	[29]	Um aluno assiste a apenas um evento em um dia.	Soft			Único Evento em um Dia para Aluno
236	[30]	As aulas devem começar ao mesmo tempo.	Hard			Início Simultâneo de Aulas
237	[30]	As aulas devem ser ministradas em horários diferentes (ou iguais).	Hard			Horários Diferentes (ou Iguais)
238	[30]	Não deve haver mais de V intervalos de tempo entre o início da primeira aula e o fim da última aula em qualquer dia.	Hard			Jornada de Trabalho
239	[30]	As aulas devem ser ministradas em dias diferentes (ou nos mesmos dias).	Hard			Dias Diferentes (ou Iguais)
240	[30]	As aulas devem ser ministradas em semanas diferentes (ou nas mesmas semanas).	Hard			Semanas Diferentes (ou Iguais)
241	[30]	As aulas devem ser ministradas em salas diferentes (ou na mesma sala).	Hard			Salas Diferentes (ou Iguais)

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
242	[30]	As aulas não podem se sobrepor em tempo, dias e semanas.	Hard			Mesmos Participantes
243	[30]	As aulas devem (ou não) se sobrepor em tempo, dia e semana.	Hard			Sobreposição (ou Não Sobreposição)
244	[30]	O primeiro encontro de uma aula em uma semana deve ser antes do primeiro encontro de outra aula.	Hard			Precedência
245	[30]	As aulas que são ministradas no mesmo dia e no mesmo conjunto de semanas devem estar pelo menos V intervalos de tempo distantes.	Hard			Intervalo Mínimo
246	[30]	As aulas não podem ser ministradas em mais de V dias diferentes.	Hard			Máximo de Dias
247	[30]	As aulas devem (ou deveriam, se a restrição for suave) ser distribuídas ao longo dos dias de forma que não haja mais do que um determinado número de intervalos de tempo ocupados V em cada dia.	Soft			Carga Diária Máxima
248	[30]	Há no máximo V_1 intervalos ao longo de um dia entre um conjunto de aulas nesta restrição.	Soft			Máximo de Intervalos
249	[30]	Há no máximo V_1 intervalos consecutivos ao longo de um dia entre um conjunto de aulas nesta restrição.	Soft			Bloco Máximo
250	[31]	Um professor comum não pode lecionar mais de 6 horas por dia.	Hard			Limite de Horas de Professor

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
251	[31]	Um estudante ou um grupo de estudantes não pode estar em mais de uma sala de aula no mesmo horário, simultaneamente.	Hard			Conflito de Horário de Estudantes
252	[31]	Um professor comum não pode estar em mais de um departamento no mesmo horário em um dia, simultaneamente.	Hard			Restrição de Departamento para Professores
253	[31]	Dois professores comuns não podem estar na mesma sala de aula ao mesmo tempo.	Hard			Conflito de Sala de Professores
254	[31]	Quando um curso é alocado para uma turma, ele deve fornecer as instalações, recursos e capacidade necessários para esse curso.	Hard			Requisitos de Sala de Aula
255	[31]	Um professor comum não pode lecionar para mais de um grupo de estudantes dentro de um departamento ou vários departamentos, simultaneamente.	Hard			Restrição de Grupo de Estudantes para Professores
256	[31]	O número de cursos alocados para cada professor comum e o número máximo de unidades correspondentes a cada curso devem ser determinados por professor comum.	Hard			Limite de Cursos e Unidades por Professor
257	[32]	Todos os eventos associados a palestras/laboratórios devem ser atribuídos a períodos distintos.	Hard			Atribuição de Períodos para Eventos
258	[32]	Todos os professores e alunos só podem assistir a uma palestra de cada vez.	Hard			Conflito de Horário de Professores e Alunos

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
259	[32]	Apenas uma palestra pode ser atribuída em uma sala de aula em um determinado período.	Hard			Atribuição de Sala de Aula
260	[32]	Todos os alunos e professores devem ser agendados em seus períodos disponíveis.	Hard			Disponibilidade de Professores e Alunos
261	[32]	Todos os cursos devem ser atribuídos às salas de aula de acordo com seus requisitos dados, incluindo localização do prédio, instalações da sala e tipos de sala.	Hard			Requisitos de Sala de Aula
262	[32]	Todas as palestras dentro de um curso que requerem períodos consecutivos devem ser obedecidas.	Hard			Períodos Consecutivos para Cursos
263	[32]	Todos os cursos devem ser agendados no tipo de sala apropriado para evitar custos operacionais ou de aluguel desnecessários associados ao tipo de sala.	Soft			Otimização de Custos de Sala
264	[32]	Um curso ministrado pelo(s) professor(es) dado(s) deve ser atribuído aos seus períodos disponíveis para economizar os custos de palestras ou contratação (por hora).	Soft			Otimização de Custos de Professores
265	[32]	A sala de aula deve ser agendada em horários consecutivos para economizar custo de configuração/limpeza.	Soft			Otimização de Custos de Configuração
266	[32]	O tamanho da sala atribuída para qualquer curso deve ter assentos suficientes ou extras para acomodar os alunos matriculados no curso.	Soft			Otimização de Espaço

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
267	[34]	Cada professor só pode ter uma aula em uma sala de aula de cada vez.	Hard			Conflito de Horário de Professores
268	[34]	Os alunos só podem ouvir uma aula em uma sala de aula de cada vez.	Hard			Conflito de Horário de Alunos
269	[34]	Apenas uma aula é permitida em cada sala de aula de cada vez.	Hard			Atribuição de Sala de Aula
270	[34]	Cada aula só pode ter um curso de cada vez, e a capacidade da sala de aula de cada curso é maior do que o número de alunos na aula.	Hard			Capacidade e Atribuição de Sala de Aula
271	[34]	O arranjo curricular da aula é distribuído de forma uniforme.	Soft			Distribuição Uniforme de Currículo
272	[34]	O tempo dos cursos contínuos dos professores deve ser razoável.	Soft			Razoabilidade do Tempo de Curso
273	[34]	As aulas de diferentes cursos estão próximas entre si no andar da sala de aula.	Soft			Proximidade de Aulas no Andar
274	[34]	O horário das aulas do mesmo curso em diferentes turmas é semelhante.	Soft			Consistência de Horário entre Turmas
275	[34]	O número de assentos na sala de aula não deve ser menor do que o número de alunos na aula.	Soft			Otimização de Espaço
276	[35]	Nenhum aluno ou professor é designado para assistir a mais de um curso simultaneamente.	Hard			Conflito de Horário de Alunos e Professores

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
277	[35]	Um curso deve ser atribuído a uma sala que possa acomodar todos os alunos desse curso específico.	Hard			Capacidade da Sala de Aula
278	[35]	Não é possível atribuir mais de um curso a uma sala em cada intervalo de tempo.	Hard			Atribuição de Sala de Aula
279	[35]	Existem certos cursos que devem ser atribuídos a intervalos de tempo específicos.	Hard			Atribuição de Horário Específico
280	[35]	Alunos e professores não devem assistir a mais de dois cursos consecutivos.	Soft			Limitação de Cursos Consecutivos
281	[35]	Alunos e professores não devem assistir a cursos no último intervalo de tempo do dia.	Soft			Limitação de Horário Final
282	[35]	Os alunos não devem assistir a apenas um único curso em um dia.	Soft			Diversificação de Cursos Diários
283	[35]	O espaço extra da sala deve ser reduzido.	Soft			Otimização de Espaço
284	[36]	Todas as aulas/laboratórios necessários para cada curso devem ser agendados e atribuídos a períodos distintos.	Hard			Agendamento de Aulas e Laboratórios
285	[36]	Estudantes e professores só podem assistir a uma aula de cada vez.	Hard			Conflito de Horário de Alunos e Professores
286	[36]	Apenas uma aula pode ocorrer em uma sala em um determinado período.	Hard			Atribuição de Sala de Aula

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
287	[36]	Professores e alunos devem estar disponíveis para que uma aula seja agendada.	Hard			Disponibilidade de Alunos e Professores
288	[36]	Todos os cursos devem ser atribuídos a salas adequadas de acordo com seus requisitos, incluindo localização do prédio, instalações da sala e tipos de sala.	Hard			Requisitos de Sala de Aula
289	[36]	Todas as aulas dentro de um curso que requerem períodos consecutivos devem ser obedecidas.	Hard			Períodos Consecutivos de Aula
290	[36]	Todos os cursos devem ser agendados nos tipos apropriados de sala para evitar custos operacionais ou de aluguel desnecessários.	Soft			Otimização de Custos de Sala
291	[36]	Os cursos ministrados pelos professores dados devem ser atribuídos em seus dias e períodos disponíveis ou preferidos para economizar nos custos de palestras ou contratações.	Soft			Otimização de Custos de Professores
292	[36]	As salas de aula devem ser agendadas em períodos consecutivos de trabalho de um dia para reduzir o número de vezes para limpar ou configurar após o uso das salas.	Soft			Otimização de Configuração e Limpeza
293	[38]	Cada curso deve ser absolutamente colocado em um horário.	Hard			Atribuição de Horário para Cursos
294	[38]	Professores, turmas e salas de aula não podem ser atribuídos a períodos em que não estão disponíveis.	Hard			Disponibilidade de Recursos

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
295	[38]	Todos os cursos devem ser agendados.	Hard			Agendamento de Cursos
296	[38]	A pré-atribuição de qualquer curso deve ser respeitada.	Hard			Respeito à Pré-Atribuição
297	[38]	Todas as salas de aula agendadas são grandes o suficiente para acomodar os alunos.	Hard			Capacidade da Sala de Aula
298	[38]	Alguns cursos requerem salas de aula particulares (por exemplo, cursos práticos devem ser atribuídos em laboratórios).	Hard			Requisitos Específicos de Sala
299	[38]	O número de dias em que um professor foi atribuído deve ser o menor possível.	Soft			Otimização de Dias de Trabalho do Professor
300	[38]	Cursos não devem ser atribuídos em horários ocupados de professores que ensinam esses cursos.	Soft			Evitar Horários Ocupados para Professores
301	[38]	Cursos não devem ser atribuídos em horários ocupados de salas.	Soft			Otimização de Uso de Sala
302	[38]	A distância que um professor caminha em um dia deve ser minimizada.	Soft			Minimização de Deslocamento do Professor
303	[38]	Professores não devem ensinar dois cursos teóricos consecutivos por dia.	Soft			Evitar Cursos Teóricos Consecutivos
304	[38]	Horários atribuídos no fim de semana devem ser evitados.	Soft			Evitar Fins de Semana

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
305	[38]	O número de horários que um professor é atribuído em um dia deve ser menor que 4.	Soft			Limitação de Horários Diários para Professores
306	[38]	O tempo ocioso entre cursos consecutivos no mesmo dia para um professor ou uma turma deve ser evitado.	Soft			Minimização de Tempo Ocioso
307	[38]	O número de alunos de cada curso é superior a 80% da capacidade da sala de aula atribuída.	Soft			Otimização de Capacidade de Sala
308	[40]	Duas aulas não podem ocorrer na mesma sala no mesmo período.	Hard			Ocupação da Sala
309	[40]	Se o professor do curso não estiver disponível para lecionar esse curso em um determinado período, então nenhuma aula do curso pode ser agendada nesse período.	Hard			Disponibilidade do Professor
310	[40]	Todas as aulas de um curso devem ser ministradas na mesma sala.	Soft			Estabilidade da Sala
311	[40]	Todas as aulas de um curso devem ser agendadas e devem ser atribuídas a períodos distintos.	Hard			Agendamento de Aulas
312	[40]	As aulas de cada curso devem ser distribuídas no mínimo número de dias.	Soft			Dias Mínimos de Trabalho
313	[40]	Aulas de cursos no mesmo currículo ou ministradas pelo mesmo professor devem ser todas agendadas em períodos diferentes.	Hard			Conflitos de Agendamento

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
314	[40]	Aulas pertencentes a um currículo devem ser adjacentes entre si (ou seja, em períodos consecutivos).	Soft			Compactação do Currículo
315	[42]	Há um pedido em que o professor não pode lecionar em certas horas e dias.	Hard			Restrições de Horário do Professor
316	[42]	Os horários dos alunos e dos professores são feitos de tal forma que não haja muito tempo livre.	Soft			Otimização de Tempo Livre
317	[43]	Eventos só podem ser realizados em salas que possuam o mesmo tipo que o tipo do evento.	Hard			Tipo de Sala
318	[43]	Eventos devem ocorrer em salas com a mesma localização que a localização do evento.	Hard			Localização da Sala
319	[43]	Apenas salas com capacidade suficiente podem hospedar o treinamento.	Hard			Capacidade da Sala
320	[43]	Os tipos de dispositivos devem corresponder aos indicados pelo evento.	Hard			Tipo de Dispositivo
321	[43]	Recursos indisponíveis (treinadores, salas e dispositivos) não podem ser incluídos no cronograma.	Hard			Recursos Indisponíveis
322	[43]	Um treinador não pode ser selecionado para treinamento que exija habilidades individuais ou treinadores qualificados.	Hard			Habilidades do Treinador
323	[43]	Cada treinamento deve ser planejado dentro do prazo especificado pelo planejador.	Hard			Prazo do Treinamento
324	[43]	Dias impostos pela equipe organizacional de um evento devem ser seguidos.	Hard			Dias Especificados

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
325	[43]	Eventos devem começar em um de seus dias de início (se especificado).	Hard			Dia de Início
326	[43]	Nenhum recurso (treinador, salas e dispositivos) pode ser colocado em eventos sobrepostos mais do que seu fator de sobrecarga especificado.	Hard			Fator de Sobrecarga
327	[43]	Horários de início/término são indicados para cada evento e devem ser seguidos.	Hard			Horários de Início/Término
328	[43]	Sequências são programadas inteiramente (em uma ordem específica) ou não são programadas.	Hard			Sequência de Eventos
329	[43]	Apenas recursos que compartilham o mesmo perímetro que o evento podem ser selecionados.	Hard			Perímetro de Recursos
330	[43]	O número de sessões programadas para o evento não pode exceder o número de sessões especificado.	Hard			Número de Sessões
331	[43]	Um treinamento pode indicar uma configuração específica de sala.	Soft			Configuração de Sala
332	[43]	Um treinamento pode indicar um tipo específico de treinador (interno, externo, associado, etc.).	Soft			Tipo de Treinador
333	[44]	Palestras ministradas pelo mesmo palestrante não podem ser realizadas no mesmo horário.	Hard			Conflito de Horário do Palestrante
334	[44]	Apenas uma palestra pode ser atribuída a um local em um horário específico.	Hard			Conflito de Horário do Local

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
335	[44]	Uma sala atribuída a uma palestra deve ser grande o suficiente para acomodar o número de estudantes.	Hard			Capacidade da Sala
336	[44]	Palestras para todos os eventos devem ser agendadas.	Hard			Agendamento de Eventos
337	[44]	Horários bloqueados para palestras devem ser levados em consideração.	Hard			Horários Bloqueados
338	[44]	Um estudante só pode assistir a uma palestra em um horário específico.	Hard			Conflito de Horário do Estudante
339	[44]	Eventos com 4 horas de palestra são distribuídos uniformemente ao longo dos dias úteis mínimos.	Soft			Distribuição de Horas de Palestra
340	[45]	Cada palestra de cada curso deve ser agendada e atribuída a diferentes horários.	Hard			Agendamento de Palestras
341	[45]	Palestras de cursos ministrados pelo mesmo professor ou pertencentes ao mesmo currículo devem ser agendadas em horários distintos.	Hard			Conflito de Horário do Professor e Currículo
342	[45]	Duas palestras não podem ser agendadas no mesmo horário e na mesma sala.	Hard			Conflito de Sala e Horário
343	[45]	Se um professor não estiver disponível para ensinar em um determinado horário (devido às suas funções, cf. especificações (3) e (4)), então nenhuma palestra é agendada naquele horário.	Hard			Disponibilidade do Professor

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
344	[45]	Palestras de cada curso em um dia devem ser agrupadas satisfazendo a especificação (2) e uma palestra agrupada (palestras consecutivas) deve ocorrer na mesma sala.	Hard			Agrupamento de Palestras e Sala
345	[45]	Cada palestra de um curso deve ser realizada na mesma sala. Violar esta restrição resulta em pontos de penalidade que refletem o número de salas distintas atribuídas.	Soft			Consistência da Sala
346	[45]	Deve haver tempo suficiente para os alunos com o mesmo currículo se deslocarem entre os prédios (cf. especificação (5)), quando se deslocam de uma palestra para outra palestra adjacente em um dia. Cada violação desta restrição resulta em 1 ponto de penalidade.	Soft			Tempo de Deslocamento entre Prédios
347	[46]	Os professores só podem dar uma aula no mesmo período de aula.	Hard			Restrições do Professor
348	[46]	Uma turma só pode ter um curso no mesmo período de aula.	Hard			Restrições da Turma
349	[46]	Apenas um curso pode ser ministrado na mesma sala de aula dentro do mesmo período de aula.	Hard			Restrições da Sala de Aula

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
350	[46]	O tempo de agendamento do mesmo elemento de classe, M_i , deve ser igual ao tempo de agendamento da tarefa de ensino. O número de horas de aula para cada elemento de classe, M_i , é registrado como X_i em uma semana, e um arranjo de classe ocupa duas horas de aula.	Hard			Restrição de Horas de Aula
351	[46]	No agendamento de cursos, o número de alunos no elemento de classe, M_i , deve ser menor que a capacidade de assentos na sala de aula. $num \leq N_j$ é o número de assentos na sala de aula e $snum \leq M_i$ é o número de alunos no elemento de classe, M_i .	Hard			Restrição de Número
352	[46]	Ao atender aos requisitos do ensino regular em sala de aula, o peso, W_i , de M_i para a sala de aula deve ser maior que o peso correspondente da categoria da sala de aula, e o peso correspondente da categoria da sala de aula é obtido pelo mapeamento $w(N_j)$.	Hard			Restrição de Peso da Sala de Aula
353	[46]	Para organizar os cursos dentro do período de tempo apropriado, de acordo com a experiência e pesquisa de ensino de longo prazo, 25 períodos de tempo na coleção de períodos de aula são classificados, e o coeficiente de efeito de aprendizado também é definido.	Soft			Restrições de Tempo de Agendamento

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
354	[46]	Para melhorar o efeito do ensino e garantir que os alunos tenham tempo suficiente para pré-visualizar e revisar o conteúdo do ensino, o período de tempo do mesmo curso deve ser disseminado o máximo possível.	Soft			Intervalo de Agendamento de Cursos
355	[46]	Para atender à demanda de ajuste de classe temporário, o número médio de assentos no par sala de aula-tempo disponível deve ser maximizado após o término do agendamento do curso.	Soft			Restrições de Recursos da Sala de Aula
356	[48]	Apenas um curso pode ser organizado na mesma turma ao mesmo tempo.	Hard			Restrições de Sala de Aula
357	[48]	O mesmo professor só pode organizar um curso ao mesmo tempo.	Hard			Restrições de Professor
358	[48]	Apenas um curso pode ser organizado na mesma sala de aula ao mesmo tempo.	Hard			Restrições de Sala de Aula
359	[48]	A capacidade da sala de aula deve ser maior do que o número de alunos na turma.	Hard			Restrições de Sala de Aula
360	[48]	O horário de cada curso deve atender aos requisitos dos professores tanto quanto possível.	Soft			Restrições de Professor
361	[48]	O horário de cada curso deve atender aos requisitos dos professores tanto quanto possível.	Soft			Restrições de Professor
362	[48]	A relação entre a utilização da sala de aula e a localização da turma é usada de forma racional.	Soft			Restrições Gerais

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
363	[50]	Conflito entre horário de aula e disponibilidade de recursos.	Hard			Disponibilidade de Recursos
364	[50]	Conflito de professor, quando um professor ensina no mesmo horário no mesmo dia.	Hard			Restrições de Professor
365	[50]	Conflito com a oração de sexta-feira.	Hard			Restrições Culturais
366	[50]	Conflito de disponibilidade de professor especial.	Hard			Restrições de Professor
367	[51]	Todos os cursos fornecidos devem ser programados, e todas as aulas ou laboratórios para cada curso devem ser atribuídos a períodos diferentes.	Hard			Restrições de Curso
368	[51]	Professores e alunos só podem participar de uma aula ou evento de cada vez.	Hard			Restrições de Professor/A-luno
369	[51]	Cada horário para as salas de aula dadas não pode atribuir uma aula ou evento mais de uma vez.	Hard			Restrições de Sala de Aula
370	[51]	Todos os cursos exigidos por professores e alunos não devem ser programados em seus períodos indisponíveis.	Hard			Restrições de Disponibilidade
371	[51]	Cada curso deve ser programado nas salas de aula específicas de acordo com seus requisitos dados (por exemplo, tipos de sala de aula, instalações da sala e localização do prédio).	Hard			Restrições de Sala de Aula
372	[51]	Todos os eventos ou aulas para um curso que requerem horários consecutivos devem ser atendidos.	Hard			Restrições de Sequenciamento

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
373	[51]	Todos os cursos devem ser programados nos tipos adequados de sala de aula para evitar custos desnecessários de aluguel ou operação.	Soft			Restrições Financeiras
374	[51]	Cada curso ministrado pelos professores dados deve ser atribuído de acordo com seus dias e períodos de trabalho preferidos para economizar nos custos de contratação ou emprego.	Soft			Restrições Financeiras
375	[51]	Cada sala de aula deve ser usada em períodos consecutivos de um dia para reduzir o número de vezes para preparar ou limpar aquela sala de aula após a usabilidade.	Soft			Restrições Operacionais
376	[52]	Um professor não pode ensinar mais de 6 horas por dia.	Hard			Restrições de Professor
377	[52]	Um professor não pode estar em mais de um departamento no mesmo horário.	Hard			Restrições de Professor
378	[52]	Uma sala de aula é atribuída a um professor em um horário.	Hard			Restrições de Sala de Aula
379	[52]	O professor tem a oportunidade de solicitar horários específicos para seus cursos.	Soft			Preferências do Professor
380	[52]	Um professor pode solicitar uma sala de aula especial para um determinado curso.	Soft			Preferências do Professor
381	[52]	Os cursos devem ser programados de forma que os horários vagos tanto do professor quanto do aluno sejam minimizados.	Soft			Otimização de Horário

Continua na próxima página

Tabela A.1 – Continuação da página anterior

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
382	[55]	Nenhum aluno assiste a mais de um evento ao mesmo tempo.	Hard			Restrições de Aluno
383	[55]	A sala é grande o suficiente para todos os alunos presentes e atende a todas as características exigidas pelo evento.	Hard			Restrições de Sala de Aula
384	[55]	Apenas um evento está em cada sala em qualquer horário.	Hard			Restrições de Sala de Aula
385	[55]	Um aluno tem uma aula no último horário do dia.	Soft			Preferências do Aluno
386	[55]	Um aluno tem mais de duas aulas seguidas.	Soft			Preferências do Aluno
387	[55]	Um aluno tem exatamente uma aula em um dia.	Soft			Preferências do Aluno
388	[57]	Nenhum aluno assiste a mais de um evento ao mesmo tempo.	Hard			Restrições de Aluno
389	[57]	A sala é grande o suficiente para todos os alunos presentes e atende a todas as características exigidas pelo evento.	Hard			Restrições de Sala de Aula
390	[57]	Apenas um evento está em cada sala em qualquer horário.	Hard			Restrições de Sala de Aula
391	[57]	Um aluno tem uma aula no último horário do dia.	Soft			Preferências do Aluno
392	[57]	Um aluno tem mais de duas aulas seguidas.	Soft			Preferências do Aluno
393	[57]	Um aluno tem uma única aula em um dia.	Soft			Preferências do Aluno
394	[55]	Nenhum aluno assiste a mais de um evento ao mesmo tempo.	Hard			Restrições de Aluno
395	[58]	A sala é grande o suficiente para todos os alunos presentes e atende a todas as características exigidas pelo evento.	Hard			Restrições de Sala de Aula

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
396	[58]	Apenas um evento está em cada sala em qualquer horário.	Hard			Restrições de Sala de Aula
397	[58]	Um aluno tem uma aula no último horário do dia.	Soft			Preferências do Aluno
398	[58]	Um aluno tem mais de duas aulas seguidas.	Soft			Preferências do Aluno
399	[58]	Um aluno tem exatamente uma aula em um dia.	Soft			Preferências do Aluno
400	[59]	Cada membro do corpo docente deve ser designado no máximo um curso, um grupo de alunos e uma sala de aula de cada vez.	Hard			Restrições de Professor
401	[59]	Para cada grupo de alunos, no máximo um curso, uma pessoa docente e uma sala de aula devem ser designados para cada período de ensino.	Hard			Restrições de Aluno
402	[59]	Cada sala de aula pode ser designada para no máximo um curso, um professor e um grupo de alunos de cada vez.	Hard			Restrições de Sala de Aula
403	[59]	Todos os cursos no currículo de cada ano estudantil devem estar no cronograma e no número certo de períodos de ensino.	Hard			Restrições de Curso
404	[59]	Cada curso deve ser programado para tantos períodos de ensino quanto o currículo de cada grupo de alunos exige.	Hard			Restrições de Curso
405	[59]	Cada pessoa no corpo docente deve ser designada para tantos períodos de ensino quanto sua carga semanal de ensino exige.	Hard			Restrições de Professor

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
406	[59]	Um curso que requer uma sessão de períodos consecutivos deve ser designado exatamente esses períodos em um determinado dia.	Hard			Restrições de Curso
407	[59]	Se períodos de um determinado dia foram designados para um curso, eles também devem ser consecutivos.	Hard			Restrições de Curso
408	[59]	As sessões de períodos devem ser tantas quanto necessário durante a semana e para as partes não repetitivas dos cursos deve haver no máximo uma sessão por dia.	Hard			Restrições de Curso
409	[59]	O curso ensinado pelo professor a um grupo de alunos deve ser designado para um determinado período em um determinado dia.	Hard			Restrições de Professor
410	[60]	Cada grupo de alunos frequenta uma lista de cursos, alguns obrigatórios e outros eletivos.	Hard			Restrições de Aluno
411	[60]	Colisões não são permitidas entre cursos destinados ao mesmo grupo de alunos.	Hard			Restrições de Curso
412	[60]	Conflitos não são permitidos no cronograma de qualquer membro individual do corpo docente.	Hard			Restrições de Professor
413	[60]	Dois cursos diferentes nunca devem ser programados para o mesmo par de período e sala de aula.	Hard			Restrições de Sala de Aula
414	[60]	Certos cursos podem ser frequentados por mais de um grupo de alunos.	Hard			Restrições de Curso

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
415	[60]	Todos os cronogramas devem ser completos, o que significa que todos os cursos para todos os grupos de alunos devem ser programados, juntamente com todos os seus requisitos para recitações e trabalhos de laboratório.	Hard			Restrições de Curso
416	[60]	Sessões de vários períodos de qualquer tamanho são permitidas, solicitando assim períodos consecutivos para uma determinada sessão.	Hard			Restrições de Curso
417	[60]	Certos cursos, como cursos que incluem uma parte com trabalho de laboratório, exigem sessões repetitivas, que são encontros de um ou vários períodos de um professor e parte de um grupo de alunos.	Hard			Restrições de Curso
418	[60]	Dado um curso ministrado a um grupo específico de alunos, apenas as salas de aula que têm capacidade para esse grupo e satisfazem os requisitos para o curso são consideradas.	Hard			Restrições de Sala de Aula
419	[60]	As preferências dos professores devem ser atendidas tanto quanto possível.	Soft			Restrições de Professor
420	[60]	Os cronogramas dos alunos devem ser o mais compactos possível.	Soft			Restrições de Aluno
421	[60]	Deve haver intervalos para almoço para todos os alunos, se possível.	Soft			Restrições de Aluno

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
422	[60]	As mudanças de sala de aula devem ser minimizadas tanto quanto possível.	Soft			Restrições de Sala de Aula
423	[61]	Apenas um evento é atribuído a cada sala em qualquer intervalo de tempo.	Hard			Restrições de Sala de Aula
424	[61]	A sala é grande o suficiente para acomodar todos os alunos presentes e satisfaz todas as características exigidas pelo evento.	Hard			Restrições de Sala de Aula
425	[61]	Nenhum aluno frequenta mais de um evento ao mesmo tempo.	Hard			Restrições de Aluno
426	[61]	Um aluno não deve ter uma aula no último intervalo de tempo do dia.	Soft			Restrições de Aluno
427	[61]	Um aluno não deve ter mais de duas aulas seguidas.	Soft			Restrições de Aluno
428	[61]	Um aluno não deve ter uma única aula em um dia.	Soft			Restrições de Aluno
429	[63]	Nenhum aluno pode ser agendado para mais de um evento ao mesmo tempo.	Hard			Restrições de Aluno
430	[63]	A sala atende a todas as características exigidas pelo evento.	Hard			Restrições de Sala de Aula
431	[63]	A capacidade da sala é respeitada.	Hard			Restrições de Sala de Aula
432	[63]	Não é permitido mais de um evento por sala e por intervalo de tempo.	Hard			Restrições de Sala de Aula
433	[63]	Um aluno tem uma aula no último intervalo de tempo do dia.	Soft			Restrições de Aluno
434	[63]	Um aluno tem mais de duas aulas seguidas.	Soft			Restrições de Aluno

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
435	[63]	Um aluno tem apenas uma aula em um dia.	Soft			Restrições de Aluno
436	[64]	Nenhum aluno pode ser designado para mais de um curso ao mesmo tempo.	Hard			Restrições de Aluno
437	[64]	A sala deve satisfazer as características exigidas pelo curso.	Hard			Restrições de Sala de Aula
438	[64]	O número de alunos que frequentam o curso deve ser menor ou igual à capacidade da sala.	Hard			Restrições de Sala de Aula
439	[64]	Não é permitido mais de um curso por intervalo de tempo em cada sala.	Hard			Restrições de Sala de Aula
440	[66]	Nenhum aluno pode ser agendado para mais de um evento ao mesmo tempo.	Hard			Restrições de Aluno
441	[66]	Um aluno tem uma aula no último horário do dia.	Soft			Restrições de Aluno
442	[66]	A sala atende a todas as características exigidas pelo evento.	Hard			Restrições de Sala de Aula
443	[66]	Um aluno tem mais de duas aulas seguidas.	Soft			Restrições de Aluno
444	[66]	Não é permitido mais de um evento por sala e por intervalo de tempo.	Hard			Restrições de Sala de Aula
445	[66]	Um aluno tem apenas uma aula por dia.	Soft			Restrições de Aluno
446	[66]	A capacidade da sala é respeitada.	Hard			Restrições de Sala de Aula
447	[67]	Cada aula de um curso deve ser agendada em um período e sala distintos.	Hard			Restrições de Aula

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
448	[67]	Duas aulas não podem ser atribuídas no mesmo período e na mesma sala.	Hard			Restrições de Sala de Aula
449	[67]	Aulas de cursos no mesmo currículo ou ministradas pelo mesmo professor não podem ser agendadas no mesmo período.	Hard			Restrições de Professor e Aluno
450	[67]	Se o professor de um curso não estiver disponível em um determinado período, então nenhuma aula do curso pode ser atribuída àquele período.	Hard			Restrições de Professor
451	[67]	Para cada aula, o número de alunos frequentando o curso não deve ser maior do que a capacidade da sala que hospeda a aula.	Soft			Restrições de Sala de Aula
452	[67]	Todas as aulas de um curso devem ser agendadas na mesma sala. Se isso for impossível, o número de salas ocupadas deve ser o menor possível.	Soft			Restrições de Sala de Aula
453	[67]	As aulas de um curso devem ser distribuídas no mínimo número de dias dado.	Soft			Restrições de Aula
454	[67]	Para um currículo dado, uma violação é contada se houver uma aula não adjacente a qualquer outra aula pertencente ao mesmo currículo no mesmo dia.	Soft			Restrições de Aluno
455	[69]	Os alunos não devem ser agendados em dois cursos ao mesmo tempo.	Hard			Restrições de Aluno

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
456	[69]	O tamanho e as características da sala devem ser adequados para os cursos atribuídos.	Hard			Restrições de Sala de Aula
457	[69]	As salas não devem ser agendadas para dois cursos ao mesmo tempo.	Hard			Restrições de Sala de Aula
458	[69]	Um aluno não deve ter uma aula no último horário do dia.	Soft			Restrições de Aluno
459	[69]	Um aluno não deve ter mais de duas aulas consecutivas.	Soft			Restrições de Aluno
460	[69]	Um aluno não deve ter apenas uma aula em um dia.	Soft			Restrições de Aluno
461	[70]	Os alunos não devem ser agendados em dois cursos ao mesmo tempo.	Hard			Restrições de Aluno
462	[70]	O tamanho e as características da sala devem ser adequados para os cursos atribuídos.	Hard			Restrições de Sala de Aula
463	[70]	As salas não devem ser agendadas para dois cursos ao mesmo tempo.	Hard			Restrições de Sala de Aula
464	[70]	Um aluno não deve ter uma aula no último horário do dia.	Soft			Restrições de Aluno
465	[70]	Um aluno não deve ter mais de duas aulas consecutivas.	Soft			Restrições de Aluno
466	[70]	Um aluno não deve ter apenas uma aula em um dia.	Soft			Restrições de Aluno
467	[72]	As aulas são assumidas como não conflitantes com qualquer outra aula.	Hard			Restrições de Aula
468	[72]	Uma aula pode ser agendada para mais de um período consecutivo.	Hard			Restrições de Aula
469	[72]	Pode ser necessário expressar uma preferência para que um requisito apareça em um período específico ou intervalo de períodos.	Soft			Restrições de Aula

Continua na próxima página

Tabela A.1 – *Continuação da página anterior*

Id.	Artigo	Constraint	Tipo	UnB	Modelo	Categoria
470	[72]	Muitas escolas desejam expressar uma preferência para que um professor não seja agendado mais do que um certo número de vezes em um determinado intervalo de períodos.	Soft			Restrições de Professor

Apêndice B

SQL para extração de dados de espaços físicos

```
1 SELECT LocCodigo,  
2     LocSigla,  
3     LocDenominacao,  
4     LocCapacidadePessoas,  
5     Loc0InCodigoRespLocal,  
6     Loc0InCodigoRespPatrimonio,  
7     LocProCodigo,  
8     LocTLoCodigo,  
9     LocLGeCodigo  
10 FROM BDTabelaApoio.dbo.TB_Local t1  
11 WHERE LocSigla IN ('BSA_N_A1_39/41')  
12 ORDER BY LocCapacidadePessoas DESC;
```


Apêndice C

SQL para extração de dados de eventos

```
1 SELECT
2     unidade.nome,
3     curso.nome AS curso_nome,
4     curso.id_turno AS curso_turno,
5     curso.id_curso,
6     turma.id_turma,
7     pessoa.nome,
8     pessoa.id_pessoa,
9     COMPONENTE_CURRICULAR.codigo AS codigo_componente,
10    turma.codigo,
11    turma.capacidade_aluno,
12    turma.descricao_horario AS preferencia_horario,
13    turma.id_campus_ies AS id_campus_turma,
14    turma.ano,
15    turma.periodo,
16    turma.local
17 FROM
18     ensino.turma AS turma
19     JOIN curso AS curso ON turma.id_curso = curso.id_curso
20     JOIN comum.unidade AS unidade ON unidade.id_unidade = curso.id_unidade
21     JOIN ensino.docente_turma AS docente_turma ON docente_turma.id_turma =
22         turma.id_turma
23     JOIN rh.servidor AS servidor ON servidor.id_servidor = docente_turma.
24         id_docente
25     JOIN comum.pessoa AS pessoa ON pessoa.id_pessoa = servidor.id_pessoa
```

```

24 JOIN GRADUACAO.RESERVA_CURSO AS RESERVA_CURSO ON RESERVA_CURSO.ID_TURMA =
      turma.id_turma
25 JOIN ENSINO.COMPONENTE_CURRICULAR AS COMPONENTE_CURRICULAR ON
      COMPONENTE_CURRICULAR.ID_DISCIPLINA = turma.ID_DISCIPLINA
26 WHERE
27 curso.nome IN ('CIENCIAS_ECONOMICAS')
28 AND curso.id_modalidade_educacao = 1
29 AND turma.AGRUPADORA = FALSEVALUE()
30 AND turma.ID_SITUACAO_TURMA NOT IN (4)
31 AND turma.ANO = 2023
32 AND turma.PERIODO = 2
33 AND COMPONENTE_CURRICULAR.NIVEL = 'G'
34 AND (
35     turma.ID_CURSO = curso.id_curso
36     OR RESERVA_CURSO.ID_MATRIZ_CURRICULAR IN (
37         SELECT MATRIZCURR18_.ID_MATRIZ_CURRICULAR
38         FROM GRADUACAO.MATRIZ_CURRICULAR AS MATRIZCURR18_
39         WHERE MATRIZCURR18_.ID_CURSO = curso.id_curso
40     )
41     OR RESERVA_CURSO.ID_CURSO = curso.id_curso
42 )
43 ORDER BY
44     turma.capacidade_aluno DESC,
45     codigo_componente,
46     turma.codigo;

```

Apêndice D

Guia Visual do SARA

Este apêndice apresenta um guia visual do Sistema de Agendamento e Recursos Acadêmicos (SARA), com o objetivo de facilitar o entendimento das principais telas e funcionalidades do sistema. Cada seção detalha os elementos presentes em cada tela, explicando como utilizá-los para realizar as operações propostas.

D.1 Organização do Guia Visual

Este guia está estruturado de forma a apresentar as principais telas do SARA. Cada subseção aborda os elementos visuais e funcionais de uma página ou módulo específico do sistema, destacando suas funcionalidades principais. A organização segue a seguinte ordem:

- **Página pública:** descrição inicial do sistema acessível sem autenticação.
- **Tela de login e autenticação:** interface para entrada no sistema.
- **Tela de agendamentos (dashboard):** visão geral e gerenciamento de horários.
- **Tela de usuários:** gestão dos perfis de usuários.
- **Página de ajuda:** seção dedicada a fornecer suporte e orientações dentro do sistema.

D.1.1 Página Pública

A página pública do Sistema de Agendamento e Recursos Acadêmicos (SARA) apresenta uma visão geral do sistema, suas tecnologias e principais características. A Figura D.1 exibe a interface inicial acessível a todos os usuários.

Bem-vindo(a) ao SARA - Sistema de Agendamento de Recursos Acadêmicos

O SARA é uma plataforma desenvolvida para auxiliar na criação de grades horárias, otimizando a alocação de eventos acadêmicos em instituições de ensino superior.

Tecnologia

O SARA utiliza inteligência artificial e algoritmos de otimização por restrições, como o CP-SAT, para gerar agendamentos acadêmicos otimizados. No núcleo do sistema, o algoritmo foi configurado para maximizar o uso eficiente dos recursos institucionais, minimizando a insatisfação com preferências de horários e respeitando padrões históricos. Além disso, busca-se garantir que eventos recorrentes sejam sempre alocados no mesmo espaço físico, promovendo consistência e organização.

Sobre a abordagem

O SARA adota uma abordagem para resolver problemas de agendamento acadêmico, combinando flexibilidade, robustez e integração. Essa metodologia permite atender às demandas específicas das instituições educacionais em diferentes cenários.

- **Flexibilidade:** Permite criar múltiplos agendamentos para decompor problemas complexos, possibilitando personalizações que atendem às necessidades específicas de diferentes setores das instituições.
- **Robustez:** Capaz de encontrar soluções ótimas mesmo sob restrições rigorosas e conflitantes, garantindo que os critérios sejam atendidos de forma confiável e otimizada.
- **Integração:** Integrável a sistemas existentes, oferecendo uma solução versátil e adaptável para diferentes ambientes acadêmicos.

Características:

- Gestão dinâmica de espaços e horários com funcionalidade de arrastar e soltar.
- Crítica algorítmica em tempo real, prevenindo conflitos decorrentes de alterações no agendamento.
- Mecanismos de login e controle de permissões para segregação e segurança de dados.
- Filtros permitindo visões diferentes por curso, professor e sala.
- Respeito aos horários históricos, contribuindo para evitar mudanças na rotina de professores.
- Agendamento das aulas preferencialmente na mesma sala, visando a satisfação de alunos e professores.

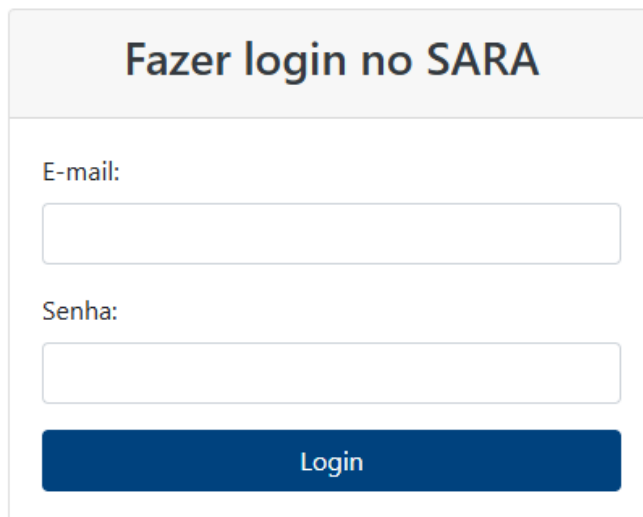
Figura D.1: Página pública do SARA

D.1.2 Tela de Login e Autenticação

A tela de login é a interface de acesso ao SARA, onde os usuários devem autenticar suas credenciais para acessar as funcionalidades do sistema. A Figura D.2 apresenta a tela de login do SARA.

Elementos da Tela

- **Campos de autenticação:**
 - **E-mail:** Campo para inserção do e-mail do usuário.
 - **Senha:** Campo para inserção da senha de acesso.
- **Botão "Login":** Inicia o processo de autenticação. O sistema valida as credenciais fornecidas antes de permitir o acesso.
- **Instrução adicional:** Caso o usuário não possua credenciais, é exibida uma mensagem para solicitar a criação do login, com um e-mail de contato para suporte (ebermoreira@unb.br).



Fazer login no SARA

E-mail:

Senha:

Login

Para solicitar a criação de um login entre em contato pelo email ebermoreira@unb.br

Figura D.2: Tela de Login do SARA

Funcionamento

Após preencher os campos de e-mail e senha, o usuário deve clicar no botão **Login**. Caso as credenciais sejam inválidas, o sistema exibirá uma mensagem de erro solicitando a correção. A tela também permite suporte para recuperação de senha através do e-mail informado no rodapé.

D.1.3 Tela de Agendamentos

Esta tela exibe uma lista de agendamentos existentes, permitindo a gestão dos horários e turmas. A Figura D.3 apresenta esta interface.

Elementos da Tela

- **Tabela de Agendamentos:** Exibe informações organizadas em colunas:
 - **Identificador:** Número único que identifica cada agendamento.
 - **Período Letivo:** Indica o período acadêmico ao qual o agendamento pertence (por exemplo, 2024.1, 2024.2).

Agendamentos			+ Criar Agendamento			
Identificador	Período Letivo	Descrição	Ações			
22	2024.1	Agendamento das turmas do Gama.	Editar	Excluir	Duplicar	Regerar
23	2024.1	Agendamento das turmas do curso de Ciências Econômicas.	Editar	Excluir	Duplicar	Regerar
24	2024.1	Agendamento das turmas do curso de Matemática.	Editar	Excluir	Duplicar	Regerar

Figura D.3: Tela de Agendamentos do SARA

- **Descrição:** Descreve o agendamento, como o curso ou grupo de turmas relacionado.
- **Ações:** Para cada agendamento, os seguintes botões estão disponíveis:
 - **Editar:** Permite modificar os detalhes do agendamento.
 - **Excluir:** Remove o agendamento do sistema.
 - **Duplicar:** Cria uma cópia do agendamento selecionado.
 - **Regerar:** Reinicia o agendamento, reaplicando as restrições definidas para gerar uma solução atualizada, descartando todas as alterações realizadas.
- **Botão "Criar Agendamento":** Localizado no canto superior direito, este botão permite ao usuário iniciar o processo de criação de um novo agendamento.

Funcionamento

Ao acessar esta tela, o usuário visualiza todos os agendamentos cadastrados disponíveis para o seu login. Ele pode realizar ações utilizando os botões disponíveis na coluna "Ações". O botão "Criar Agendamento" permite iniciar o processo para um novo agendamento

Tela de Edição de Agendamentos

A tela de edição de agendamentos permite visualizar e modificar os detalhes de um agendamento existente. Esta tela é composta por duas seções principais: a parte superior, que apresenta informações gerais do agendamento e filtros para refinamento dos dados, e a parte inferior, que exibe o quadro de horários. As Figuras D.4 e D.5 ilustram essas seções.

Parte Superior: Informações do Agendamento e Filtros

A imagem mostra a interface de usuário para a edição de agendamentos, dividida em duas seções principais: 'Agendamento' e 'Filtros'.

Seção Agendamento: Possui um cabeçalho com o título 'Agendamento' e um ícone de caneta azul no canto superior direito. Abaixo, há um campo de texto com o identificador '36', o período letivo '2024.2' e a descrição 'Agendamento das turmas do curso de Ciências Econômicas'.

Seção Filtros: Possui um cabeçalho com o título 'Filtros'. Abaixo, há três menus suspensos: 'Escolha uma sala:' com a opção 'Todas as Salas', 'Escolha um docente:' com a opção 'Todos os docentes' e 'Escolha um curso: (Máx. 5)' com a opção 'Todos os Cursos' e 'CIÊNCIAS ECONÔMICAS - Bacharelado - Diurno'. Na base da seção, há dois botões: 'Aplicar Filtros' (em azul) e 'Limpar Filtros' (em cinza).

Figura D.4: Seção superior da tela de edição de agendamentos.

Informações do Agendamento: Esta parte apresenta informações gerais do agendamento, permitindo alterá-las.

- **Identificador:** Número único que identifica o agendamento no sistema.
- **Período letivo:** Indica o período acadêmico ao qual o agendamento pertence.
- **Descrição:** Exibe uma descrição do agendamento, facilitando a identificação e distinção entre diferentes agendamentos mantidos no sistema.
- **Botão de edição:** Localizado no canto superior direito, permite editar os detalhes gerais do agendamento. Este botão se apresenta na forma de um ícone com uma caneta.

Filtros: Esta seção permite ao usuário aplicar filtros para refinar os dados exibidos, facilitando a visualização e o gerenciamento de informações, como salas, docentes ou cursos selecionados.

- **Escolha uma sala:** Menu suspenso para filtrar as informações por salas.
- **Escolha um docente:** Menu suspenso para filtrar os horários por docente.
- **Escolha um curso:** Lista com múltiplas opções que permite selecionar até cinco cursos simultaneamente.
- **Botões de ação:**

- **Aplicar Filtros:** Atualiza os dados exibidos no quadro de horários com base nos filtros selecionados.
- **Limpar Filtros:** Remove todos os filtros aplicados e exibe os dados completos.

Parte Inferior: Quadro de Horários

Salvar alterações Desfazer alterações ← Voltar

Horário	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
M1 08:00 - 08:55	POL0011(100) ICC ANF.18(135) Ⓢ	ECO0050(75) BSA N A1 39/41(88) Ⓢ	POL0011(100) ICC ANF.18(135) Ⓢ	ECO0050(75) BSA N A1 39/41(88) Ⓢ		
	ECO0023(75) BSA N AT 29/41(88) Ⓢ	ECO0187(72) ICC ANF.18(135) Ⓢ	ECO0023(75) BSA N AT 29/41(88) Ⓢ	ECO0187(72) ICC ANF.18(135) Ⓢ		
	ECO0185(75) BSA N A1 39/41(88) Ⓢ	ECO0025(60) BSA N A1 49/41(88) Ⓢ	ECO0185(75) BSA N A1 39/41(88) Ⓢ	ECO0025(60) BSA N A1 49/41(88) Ⓢ		
	EST0023(70) BSA N A1 09/41(77) Ⓢ	FDD0004(60) BSA N A1 63/11(70) Ⓢ	EST0023(70) BSA N A1 09/41(77) Ⓢ	FDD0004(60) BSA N A1 63/11(70) Ⓢ		
	ECO0134(50) PAT AT 125(50) Ⓢ	ECO0040(50) PAT AT 124(50) Ⓢ	ECO0134(50) PAT AT 125(50) Ⓢ	ECO0040(50) PAT AT 124(50) Ⓢ		
	ECO0054(50) ICC ANF.15(120) Ⓢ		ECO0054(50) ICC ANF.15(120) Ⓢ			
	ECO0079(50) PAT AT 100(50) Ⓢ		ECO0079(50) PAT AT 100(50) Ⓢ			

Figura D.5: Seção inferior da tela de edição de agendamentos: quadro de horários.

O quadro de horários exibe as alocações de eventos organizados por dias da semana e horários, com as seguintes características:

- **Colunas:** Representam os dias da semana, de segunda-feira a sábado.
- **Linhas:** Indicam os horários disponíveis no período acadêmico.
- **Blocos de aulas:** Cada bloco representa uma aula, com informações como código da disciplina, local da aula e número de vagas alocadas e disponíveis.
- **Botões de ação:**
 - **Salvar alterações:** Salva todas as modificações realizadas no quadro de horários.

- **Desfazer alterações:** Restaura o estado anterior do agendamento, descartando as mudanças feitas.
- **Voltar:** Retorna à tela anterior sem salvar as alterações.

Crítica algorítmica O quadro de horários exibido permite que alterações sejam feitas por meio de arrastar e soltar. A cada movimento realizado, o algoritmo verifica automaticamente a disponibilidade de salas, horários e professores envolvidos, além de avaliar a sobreposição de recursos. Mais detalhes podem ser vistos na Seção 5.2.4.

Eventos

No SARA, os eventos que compõem os agendamentos são exibidos na forma representada na Figura D.6. Cada evento apresenta as seguintes informações:

- **Identificação da turma e quantidade de alunos:** Representado no formato POL0011(100), onde POL0011 é a identificação da turma e 100 é o número de alunos.
- **Espaço físico e capacidade de alunos:** Representado no formato ICC ANF.18(135), onde ICC ANF.18 é a identificação do espaço físico e 135 é a sua capacidade.

Informações adicionais, como o nome do docente, o horário histórico ou preferencial e o código da turma, podem ser visualizadas ao passar o mouse sobre o ícone de informações exibido no evento.

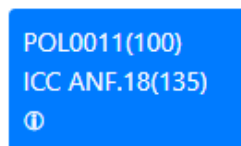


Figura D.6: Representação de evento no SARA.

Os eventos do agendamento podem assumir cores diferentes nas seguintes situações:

Cor	Descrição
POL0011(100) ICC ANF.18(135) ⓘ	Posição original do evento.
POL0011(100) ICC ANF.18(135) ⓘ	Após a crítica algorítmica da movimentação, o evento ocupa uma posição válida.
POL0011(100) ICC ANF.18(135) ⓘ	Após a crítica algorítmica da movimentação, o evento está em conflito.
POL0011(100) ICC ANF.18(135) ⓘ	A borda vermelha identifica o evento conflitante.
POL0011(100) ICC ANF.18(135) ⓘ	Ao clicar em um evento, todas as suas ocorrências são sinalizadas.

Tabela D.1: Legenda de cores dos eventos de agendamento

D.1.4 Tela Usuários

A tela de usuários permite ao administrador visualizar, adicionar, editar e excluir usuários no sistema. Essa funcionalidade está disponível exclusivamente para usuários com perfil de **Administrador**. A Figura D.7 apresenta a interface da tela. Mais informações sobre os perfis de usuários no SARA podem ser vistas na seção 5.2.5.

Usuários

[Adicionar Novo Usuário](#)

Nome	E-mail	Perfil	Ações
Éber Moreira	ebernj@hotmail.com	Administrador	Editar Excluir
gestor1@unb.br	gestor1@unb.br	Gestor	Editar Excluir
gestor2@unb.br	gestor2@unb.br	Gestor	Editar Excluir
prof1	prof1@unb.br	Comum	Editar Excluir
prof2	prof2@unb.br	Comum	Editar Excluir

Figura D.7: Tela de Usuários

Elementos da Tela

- **Botão "Adicionar Novo Usuário":** Localizado no canto superior direito, permite criar um novo usuário no sistema.
- **Tabela de Usuários:** Apresenta informações dos usuários cadastrados no sistema, organizadas em colunas:
 - **Nome:** Nome do usuário.
 - **E-mail:** Endereço de e-mail utilizado para autenticação.
 - **Perfis:** Perfil do usuário no sistema, podendo ser "Administrador", "Gestor" ou "Comum".
 - **Ações:** Contém botões para realizar as seguintes operações:
 - * **Editar:** Abre a tela de edição para alterar as informações do usuário.
 - * **Excluir:** Remove o usuário do sistema após confirmação.

D.1.5 Página de Ajuda

A página de ajuda do SARA oferece uma visão sobre os principais recursos e funcionalidades do sistema, além de fornecer orientações e exemplos para os usuários. Nesta página, estão disponíveis as seguintes informações:

Contato e Suporte

Informações de contato, incluindo o e-mail e telefone, são fornecidas para que os usuários possam solicitar suporte técnico ou enviar feedback sobre o sistema.

Instruções para Criação de Agendamentos

A página apresenta um guia passo a passo para criar novos agendamentos, desde o login no sistema até o envio dos arquivos de entrada. Também são destacadas informações sobre o tempo máximo de execução do algoritmo (30 minutos) e o comportamento do sistema caso uma solução ótima não seja encontrada.

Constraints Implementadas

A página apresenta as regras (*constraints*) que orientam o funcionamento do algoritmo de agendamento, divididas em:

- **Hard Constraints (HC):** Regras obrigatórias, como a alocação distinta de aulas em períodos e salas.
- **Soft Constraints (SC):** Regras desejáveis, como alocar turmas no horário histórico ou manter aulas de uma turma na mesma sala.

Formato dos Arquivos de Entrada

A página explica os formatos exigidos para os arquivos de entrada no sistema (.csv) e os campos necessários para cada tipo:

- **Espaços Físicos:** Contém informações sobre identificadores, capacidade, localização e descrição dos espaços disponíveis.
- **Eventos Acadêmicos:** Inclui dados como nome do curso, docente, identificação da turma, capacidade de alunos e horários históricos.

Exemplos de Arquivos

A página também disponibiliza exemplos de arquivos que podem ser utilizados para gerar testes ou como referência para a criação de novos casos. Entre os exemplos estão:

- Espaços Físicos FGA
- Eventos FGA
- Espaços Físicos Ciências Econômicas
- Eventos Ciências Econômicas

Glossário

Agendamento Processo de alocação de recursos, como salas, professores e horários, de maneira a atender restrições e preferências específicas.

Alocação de Recursos Processo de designar recursos como salas, professores e horários para eventos, atendendo restrições e maximizando a eficiência.

Busca Local (Local Search) Método de otimização que explora soluções próximas para melhorar a solução atual em problemas combinatórios.

CP-SAT Solver Otimizador baseado em programação por restrições (Constraint Programming) utilizado para resolver problemas combinatórios, como agendamentos complexos.

Espaço Físico Local onde eventos acadêmicos são realizados, como salas de aula, laboratórios ou auditórios, sujeitos a restrições de capacidade e disponibilidade.

Evento Entidade que representa uma atividade acadêmica, como uma aula, palestra ou reunião, que precisa ser agendada em um espaço físico e horário específicos.

Feedback Mecanismo de coleta de informações dos usuários sobre a qualidade e eficácia do agendamento, permitindo ajustes e melhorias contínuas.

Hard Constraints Restrições que devem ser obrigatoriamente atendidas em uma solução, como a disponibilidade de professores e salas.

Heurística Técnica de solução de problemas baseada em métodos empíricos, que busca encontrar boas soluções rapidamente sem garantir a solução ótima.

Horário Intervalo de tempo alocado para a realização de eventos acadêmicos, sujeito a restrições de sobreposição e disponibilidade de recursos.

Inteligência Artificial (IA) Campo da ciência da computação focado no desenvolvimento de sistemas capazes de executar tarefas que normalmente requerem inteligência humana, como aprendizado e tomada de decisão.

Metaheurística Estratégia de alto nível para resolver problemas de otimização combinatoria, baseada em abordagens como Simulated Annealing, Algoritmos Genéticos e Enxame de Partículas.

Modelagem Matemática Representação de problemas reais em termos matemáticos, incluindo variáveis, restrições e funções objetivo para análise e otimização.

Predição Uso de dados históricos e algoritmos para prever demandas futuras de agendamento, como disponibilidade de salas ou aumento de inscrições em disciplinas.

Presolver Técnica utilizada para simplificar e reduzir problemas de otimização antes de serem processados por um solver, eliminando redundâncias e identificando soluções triviais.

Restrições (Constraints) Conjunto de condições que devem ser atendidas para garantir a viabilidade de uma solução de agendamento. Podem ser classificadas como rígidas (hard constraints) ou suaves (soft constraints).

Revisão Sistemática da Literatura Método estruturado de busca, seleção e análise de estudos acadêmicos relevantes para um tema específico.

Soft Constraints Restrições desejáveis, mas não obrigatórias, como preferências por horários ou proximidade entre salas consecutivas.

SQL (Structured Query Language) Linguagem utilizada para gerenciar e consultar dados em sistemas de gerenciamento de banco de dados relacionais.

Turma Grupo de estudantes inscritos em uma disciplina específica, que requerem alocação de um professor, sala e horário para suas atividades.