

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE COMPUTAÇÃO
CURSO DE ENGENHARIA DE COMPUTAÇÃO

ALLAN WENDLAND KRETZMANN

**GERAÇÃO DE GRADES HORÁRIAS ESCOLARES: AUTOMAÇÃO
COM BASE NA META-HEURÍSTICA SIMULATED ANNEALING**

PROPOSTA DE TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO
2023

ALLAN WENDLAND KRETZMANN

GERAÇÃO DE GRADES HORÁRIAS ESCOLARES: AUTOMAÇÃO COM BASE NA META-HEURÍSTICA SIMULATED ANNEALING

Proposta de Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação da Universidade Tecnológica Federal do Paraná, como requisito parcial para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador: André Yoshiaki Kashiwabara
Universidade Tecnológica Federal do Paraná

CORNÉLIO PROCÓPIO
2023



4.0 Internacional

Esta é a mais restritiva das seis licenças principais Creative Commons. Permite apenas que outros façam download dos trabalhos licenciados e os compartilhem desde que atribuam crédito ao autor, mas sem que possam alterá-los de nenhuma forma ou utilizá-los para fins comerciais.

RESUMO

KRETZMANN, Allan. Geração de Grades Horárias Escolares: Automação com base na meta-heurística Simulated Annealing. 2023. 40 f. Proposta de Trabalho de Conclusão de Curso – Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Cornélio Procopio, 2023.

O *High School Timetabling Problem* consiste na tarefa de alocar aulas a horários letivos em instituições de ensino, considerando restrições relacionadas aos professores, salas e disciplinas. Apesar de todos os avanços tecnológicos que experienciamos na atualidade, esta tarefa ainda é realizada primordialmente de maneira manual nas escolas brasileiras, produzindo grades horárias subótimas e consequentemente trazendo frustrações a todos os envolvidos, sejam estes alunos, docentes ou diretores. Todas as dificuldades relacionadas ao problema, somadas a novos desafios como os Itinerários Formativos do Novo Ensino Médio, pedem uma solução melhor. Este trabalho teve como objetivo implementar um sistema de geração automatizada das grades horárias atendendo o maior número de objetivos possível, através da aplicação de um algoritmo de otimização baseado na meta-heurística *Simulated Annealing*, técnica de otimização inspirada no processo de recozimento de materiais na metalurgia. A solução desenvolvida é uma aplicação *web* que segue uma arquitetura cliente-servidor empregando JavaScript no *frontend* e *backend*, além do otimizador desenvolvido na linguagem C++. Espera-se que a aplicação desenvolvida passe a fazer parte do ferramental de instituições de ensino e auxilie estas a entregar horários escolares que superem os diversos desafios associados e providenciem uma experiência melhor para todos.

Palavras-chave: High School Timetabling Problem. Simulated Annealing. Otimização multi-objetivo.

ABSTRACT

KRETZMANN, Allan. School Timetable Generation: Simulated Annealing based Generation. 2023. 40 f. Proposta de Trabalho de Conclusão de Curso – Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2023.

The High School Timetabling Problem involves the task of assigning classes to class periods in educational institutions while considering constraints related to teachers, classrooms, and subjects. Despite the technological advancements we experience today, this task is still primarily done manually in Brazilian schools, resulting in suboptimal timetables and frustrations for all stakeholders, including students, teachers, and administrators. The difficulties associated with the problem, combined with new challenges provided by the new high school model implemented in the country, call for a better solution. This study has aimed to implement an automated timetabling system that satisfies as many objectives as possible by applying an optimization algorithm based on the Simulated Annealing metaheuristic, a technique inspired by the annealing process in metallurgy. The developed solution is a web application following a client-server architecture, employing JavaScript for both the frontend and backend, in addition to the optimizer implemented in the C++ language. We hope that the developed application becomes part of the toolkit for educational institutions and helps them deliver school timetables that overcome the various associated challenges and provide a better experience for all parties involved.

Keywords: High School Timetabling Problem. Simulated Annealing. Multi-objective optimization.

LISTA DE FIGURAS

Figura 1 – Aulas constantes	9
Figura 2 – Restrições	10
Figura 3 – Exemplo de região	12
Figura 4 – Exemplo de grupo de alinhamento sobre a grade horária gerada	13
Figura 5 – Efeito de grupo de alinhamento	14
Figura 6 – Configuração de preferências	16
Figura 7 – Modelo Entidade-Relacionamento	18
Figura 8 – Consultas de validação da modelagem	19
Figura 9 – Consulta SQL de grade horária com sete salas	20
Figura 10 – Resultado da consulta de uma grade no banco de dados	20
Figura 11 – Diagrama de Casos de Uso	21
Figura 12 – Tela - Listagem de Configurações de Grade	22
Figura 13 – Tela - Estrutura da Escola - Professores	23
Figura 14 – Tela - Estrutura da Escola - Turmas e Turnos	24
Figura 15 – Tela - Aulas por professor	24
Figura 16 – Tela - Restrições e Preferências	25
Figura 17 – Tela - Regiões e Grupos de Alinhamento	25
Figura 18 – Tela - Mínimos e Máximos	26
Figura 19 – Tela - Horários	26
Figura 20 – Modelo Entidade-Relacionamento com adição da tabela de usuários	27
Figura 21 – Tela de Acesso	28
Figura 22 – Método de Cadastro	29
Figura 23 – Método de Login	30
Figura 24 – Middleware de autenticação	31
Figura 25 – Middleware de validação da configuração	31
Figura 26 – Modelo Entidade-Relacionamento com Matérias	33
Figura 27 – Estrutura da Escola com Matérias	33
Figura 28 – Tela de configuração de quantidades de aulas por matéria	34
Figura 29 – Visualização de grade horária com matérias	34
Figura 30 – Mensagem de erro de validação	37
Figura 31 – Grade horária exportada para planilha	38

SUMÁRIO

1 – INTRODUÇÃO	1
1.1 DELIMITAÇÃO DO TEMA	1
1.2 PROBLEMAS E PREMISSAS	1
1.3 OBJETIVOS	2
1.3.1 Objetivo Geral	2
1.3.2 Objetivos Específicos	2
1.4 ESTRUTURA ORGANIZACIONAL	2
2 – REVISÃO DE LITERATURA	3
2.1 TIMETABLING PROBLEM	3
2.1.1 Highschool Timetabling Problem	3
2.2 NOVO ENSINO MÉDIO	3
2.3 SIMULATED ANNEALING	4
3 – DESENVOLVIMENTO	5
3.1 RECURSOS UTILIZADOS	5
3.2 MÉTODOS	5
3.3 ANÁLISE E IMPLEMENTAÇÃO	6
3.3.1 Otimizador inicial	6
3.3.2 PESOS E RESTRIÇÕES	7
3.3.2.1 Agrupamento de aulas	7
3.3.2.2 Constantes	8
3.3.2.3 Restrições	9
3.3.2.4 Regiões	11
3.3.2.5 Grupos de Alinhamento	13
3.3.2.6 Janelas	15
3.3.2.7 Preferências	15
3.3.2.8 Armazenamento de soluções	16
3.3.3 SERVIDOR E BANCO DE DADOS	18
3.3.4 INTERFACE	21
3.3.5 USUÁRIOS E AUTENTICAÇÃO	26
3.3.5.1 Adaptação do banco de dados	27
3.3.5.2 Tela de login	27
3.3.5.3 Rotas de autenticação	28
3.3.5.4 Middlewares	30
3.3.6 ALOCAÇÃO DE MATÉRIAS	32
3.3.6.1 Alteração no banco de dados	32
3.3.6.2 Alteração de telas	33
3.3.6.3 Alteração de métodos no servidor	34
3.3.6.4 Matérias no otimizador	35
3.3.7 VALIDAÇÃO DE CONFIGURAÇÕES	37
3.3.8 EXPORTAÇÃO DE GRADES HORÁRIAS	38
4 – CONSIDERAÇÕES FINAIS	39

Referências 40

1 INTRODUÇÃO

A cada ano letivo, se faz necessária a definição de grades horárias que atendam inúmeros requisitos para o bom funcionamento das mais diversas instituições de ensino. Estas exigências dependem e impactam diretamente os envolvidos, sejam estes professores, alunos, gestores, ou ainda, os responsáveis pelo planejamento dos horários. Os requisitos citados anteriormente podem mudar durante o ano, requerendo constantes atualizações das grades definidas. Esta tarefa, segundo [Bardadym \(1996\)](#), requer grande esforço quando realizada manualmente, frequentemente resultando em soluções subótimas.

Para o desenvolvimento dos horários escolares, vários critérios devem ser considerados, como o número de aulas de cada professor, salas, turnos, além das restrições associadas a cada professor em particular. A combinação destes fatores para a produção de um horário otimizado é conhecida como *school timetabling problem* ([FONSECA; SANTOS; CARRANO, 2016](#)). Para a escrita deste trabalho, foi realizada a leitura de artigos que abordam o tema em questão, deste modo os autores [Fonseca, Santos e Carrano \(2016\)](#), [Tan et al. \(2021\)](#) e [Abramson \(1991\)](#) se mostraram imprescindíveis para a sua realização.

Ao decorrer da revisão bibliográfica, foi reafirmada a relevância desta pesquisa e da elaboração de um projeto que solucione o problema, dada a complexidade e recorrência deste. [Poulsen \(2012\)](#) estima que a maioria das instituições de ensino brasileiras efetua essa tarefa de maneira manual, o que implica em atrasos para as escolas, além de desgaste entre os docentes envolvidos nas negociações de restrições. Isto justifica novamente a necessidade de uma solução automatizada que entregue grades otimizadas em um tempo hábil.

Diante deste contexto, fica evidente a necessidade de solucionar o seguinte problema: como automatizar a geração de grades horárias que atendam as diversas demandas das instituições de ensino? Por ser um problema combinatório, consideram-se técnicas viáveis *Simulated Annealing*, Busca Tabu, Algoritmos Genéticos, entre outros ([TAN et al., 2021](#)).

Como critério metodológico, será aplicado o procedimento de levantamento de requisitos com os encarregados por planejar as grades horárias nas escolas, para embasar uma análise quantitativa que traga informações para o desenvolvimento do projeto. Os dados coletados, juntamente com os conhecimentos obtidos a partir da revisão bibliográfica viabilizarão identificar a melhor alternativa para a solução do problema. Por fim ocorrerá a implementação da aplicação.

1.1 DELIMITAÇÃO DO TEMA

O presente trabalho visa abordar o tema de otimização de grades horárias escolares, tendo em mente especificamente as necessidades de instituições brasileiras de ensino fundamental e médio.

1.2 PROBLEMAS E PREMISSAS

Como mencionado anteriormente, as instituições de ensino necessitam de constantes atualizações de suas grades horárias. Este processo envolve muitas variáveis, e impacta diretamente alunos, professores e gestores. Devido a esta complexidade, o problema já é conhecido na literatura como *school timetabling problem*, sendo dividido nos tipos *exam timetabling*, *course timetabling* e *high school timetabling* ([TAN et al., 2021](#)). Este trabalho tem como foco esta

última variação, que, segundo os mesmos autores, é definido em termos da disponibilidade dos professores, número de salas, número de aulas por professor em cada sala e restrições.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

Desenvolver uma aplicação que automatize e simplifique o processo de criação de grades horárias escolares, otimizando diferentes características de qualidade destas, fazendo com que tais sejam livres de conflitos e atendam a restrições impostas pelas instituições de ensino.

1.3.2 Objetivos Específicos

1. Compreender quais são as reais necessidades das escolas quanto ao planejamento de grades horárias
2. Implementar métricas de qualidade para os horários, que refletem se as necessidades são atendidas com êxito
3. Aplicar um algoritmo de otimização na geração de grades que evite conflitos, atenda restrições de horários dos docentes, minimize a quantidade de janelas nas aulas dos professores, faça agrupamentos formando aulas duplas e evite excessos da mesma aula
4. Proporcionar uma experiência do usuário simples e agradável durante o uso da aplicação

1.4 ESTRUTURA ORGANIZACIONAL

O trabalho em questão está estruturado da seguinte forma: o capítulo 1 traz a introdução sobre o tema, uma visão geral sobre a pesquisa e seus objetivos; no capítulo 2 será desenvolvida a fundamentação teórica, listando todas as fontes de conhecimento consultadas para para o desenvolvimento do projeto; a forma de implementação do projeto, seus materiais e métodos serão discutidos no capítulo ??; o capítulo ?? apresentará o cronograma das atividades planejadas e por fim, o capítulo 4 trará as considerações finais.

2 REVISÃO DE LITERATURA

Este capítulo tem como objetivo explicar o problema de *timetabling* e suas especificidades no caso de grades horárias de ensino médio, assim como a técnica de otimização *Simulated Annealing* e sua aplicação ao problema.

2.1 TIMETABLING PROBLEM

O problema de *timetabling* ou alocação de horários é definido em termos de quatro conjuntos, sendo estes horários, recursos, encontros e restrições. O problema consiste em associar os encontros desejados aos horários, utilizando os recursos disponíveis e minimizando as violações das restrições.

É evidente que existem inúmeras tarefas e indústrias que dependem da alocação de horários, portanto é necessário especializar o problema. Uma das variações deste problema é o *school timetabling problem*, que segundo [Tan et al. \(2021\)](#) subdivide-se em *exam timetabling*, *course timetabling* e *high school timetabling*. Esta última subdivisão, relacionada a escolas de ensino médio, será o foco deste trabalho.

2.1.1 Highschool Timetabling Problem

No caso específico da alocação de horários para escolas de ensino médio, o problema é definido em termos da disponibilidade dos professores, número de salas, número de aulas por professor em cada sala e restrições. Adequando estas necessidades à formulação do *timetabling problem*, os encontros que desejamos alocar são entre os professores e determinada sala de aula, dados os horários de aula em que a escola opera, visando a utilização dos recursos disponíveis, como salas de aula ou laboratórios, minimizando a violação de restrições associadas aos professores e recursos ([TAN et al., 2021](#)).

Segundo [Abramson \(1991\)](#), o problema envolve agendar aulas, professores e salas de aula de tal forma que nenhum professor, turma ou sala de aula seja utilizado mais de uma vez em determinado horário. Situações em que ocorra este agendamento de recursos duplicados serão tratadas como “conflitos” neste trabalho.

Levando em consideração a necessidade de agendar diversas aulas evitando conflitos, e o atendimento de restrições variadas, o *highschool timetabling problem* é um problema de otimização multiobjetivo, que segundo [Cooper e Kingston \(1995\)](#) é da classe NP-completo.

Considerando a natureza complexa do problema, este trabalho propõe desenvolver e analisar a eficiência de um algoritmo de *Simulated Annealing*, assim como implementar uma interface para interação com este.

2.2 NOVO ENSINO MÉDIO

O Novo Ensino Médio consiste na atualização das matrizes curriculares das salas desta etapa do sistema educacional (1º, 2º e 3º anos). Formulado através da Lei nº 13.415/2017 ([BRASIL, 2017](#)), o novo modelo traz ainda mais desafios para a tarefa de organização de grades horárias escolares.

O principal destes novos desafios no contexto da criação de grades horárias é o conceito dos Itinerários Formativos, conjuntos de atividades que os estudantes podem escolher realizar durante o ensino médio, como disciplinas, projetos, oficinas, entre outros.

No caso de grandes instituições de ensino, é possível que seja viável oferecer essas atividades optativas através da criação de turmas específicas, entretanto, em escolas de pequeno porte, a adição destas turmas e quantidade relativamente pequena de estudantes por turma pode facilmente desgastar os recursos da instituição.

Tendo estas limitações em mente, observou-se que uma estratégia aplicada por escolas de porte menor é a oferta de disciplinas optativas (Itinerários Formativos) em horários simultâneos em diferentes salas, nos períodos normais de ensino. No momento em que essas aulas acontecem, cada aluno das duas turmas se locomove para a sala adequada onde a aula que escolheu será ministrada, e com isso a escola não precisa alocar salas, professores ou horários adicionais.

Evidentemente, alocar horários simultâneos para determinadas aulas dificulta ainda mais o planejamento do horário escolar, sendo assim mais uma questão que pode ser auxiliada por um sistema de otimização da grade.

2.3 SIMULATED ANNEALING

Diversos problemas podem ser resolvidos imitando o que ocorre em fenômenos físicos ou naturais. Segundo [Laarhoven \(1987\)](#), *Simulated Annealing* é uma meta-heurística que se inspira no processo de recozimento na área da metalurgia. Este processo tem como objetivo reduzir as tensões internas de determinado material, da seguinte forma:

1. O material inicia com uma alta temperatura, com seus átomos desordenados e livres para vibrarem e se deslocarem
2. O sistema é resfriado gradualmente, fazendo com que os átomos encontrem posições cada vez mais estáveis. Em outras palavras, conforme a temperatura diminui, torna-se cada vez mais improvável um átomo se deslocar para uma posição menos estável
3. Por fim, atinge-se certa temperatura em que não ocorrem mais mudanças significativas no material.

Esta meta-heurística foi escolhida para o desenvolvimento deste trabalho considerando seu potencial verificado durante a revisão de literatura, e a clareza dos paralelos que podem ser realizados com o processo de têmpera real: os átomos representam os professores e aulas, que podem ser deslocados na grade horária até que atinjam uma posição estável, ou seja, que viole o mínimo possível de restrições do problema.

3 DESENVOLVIMENTO

Este capítulo tem como objetivo evidenciar todas as atividades executadas durante o desenvolvimento do projeto. A [Seção 3.1](#) apresentará uma listagem dos *softwares* utilizados no desenvolvimento do projeto; a [Seção 3.2](#) explicitará o método de desenvolvimento, e a [Seção 3.3](#) evidenciará a execução das atividades propriamente ditas.

3.1 RECURSOS UTILIZADOS

Tratando-se do desenvolvimento de um projeto *full stack*, utilizaram-se diversas tecnologias e ferramentas, sendo as de maior importância:

- **VSCode**: Editor de código-fonte desenvolvido pela Microsoft, escolhido por sua versatilidade e aplicabilidade para desenvolver todos os componentes do sistema
- **DBeaver**: Ferramenta de código aberto de administração de bancos de dados, utilizada para validação da modelagem e interações manuais com a base de dados
- **PostgreSQL**: Sistema gerenciador de banco de dados relacional, escolhido por sua robustez, possibilitará a persistência dos dados da aplicação em tabelas
- **JavaScript**: Linguagem de programação ubíqua na engenharia de software, escolhida por sua versatilidade para desenvolver a interface *web* e servidor
- **VueJS**: *Framework* para desenvolvimento frontend na linguagem JavaScript, escolhido por sua versatilidade e familiaridade do graduando com este
- **VuetifyJS**: *Framework* de componentes e estilização para o VueJS, utilizado para padronizar o design da aplicação
- **NodeJS**: *Runtime* de JavaScript que permite a utilização dessa linguagem para escrever aplicações no lado do servidor;
- **ExpressJS**: *Framework* para desenvolvimento backend na linguagem JavaScript, utilizando a plataforma NodeJS
- **C++**: Linguagem de programação compilada, escolhida por sua alta performance para o desenvolvimento do otimizador
- **G++**: Compilador utilizado para converter o código fonte em C++ do otimizador para um arquivo executável
- **GDB**: Debugador da linguagem C++, utilizado para localizar problemas no código fonte do otimizador

3.2 MÉTODOS

Para o processo de software da aplicação, optou-se pelo processo de desenvolvimento incremental. Este processo consiste na divisão da implementação do projeto em incrementos, os quais são executados linearmente, porém de forma escalonada. (PRESSMAN; MAXIM, 2016)

Este processo de software foi escolhido por possibilitar o planejamento de execução do projeto em etapas lógicas, que podem ser sobrepostas através de escalonamento, de acordo com as necessidades encontradas. Conforme os requisitos levantados na seção anterior, dividiram-se as tarefas de desenvolvimento nos incrementos pertinentes:

- **Incremento 1**: Desenvolvimento inicial do otimizador, aplicando Simulated Annealing apenas para a resolução de conflitos;
- **Incremento 2**: Implementação de restrições no otimizador;
- **Incremento 3**: Criação do servidor e banco de dados;

- **Incremento 4:** Desenvolvimento da interface *web*;
- **Incremento 5:** Adição do sistema de usuários;
- **Incremento 6:** Adaptação da modelagem para incluir matérias aos horários alocados pelo otimizador;
- **Incremento 7:** Implementação de validações das configurações de grade inseridas pelo usuário;
- **Incremento 8:** Desenvolvimento de sistema de exportação de grades horárias.

3.3 ANÁLISE E IMPLEMENTAÇÃO

3.3.1 Otimizador inicial

A primeira parte do projeto desenvolvida foi a versão inicial do otimizador, cuja tarefa era gerar uma grade horária válida, evitando conflitos, ou seja, professores alocados para mais uma turma ao mesmo tempo. Para realizar esta tarefa, aplicou-se a meta-heurística de *simulated annealing*, originando o algoritmo 1.

Algoritmo 1: Otimizador de grades inicial

Input: Lista de professores LP , lista de turmas LT , matriz de aulas por professor por turma MA , temperatura inicial TI , Taxa de resfriamento TR

Output: Grade horária de professores otimizada

$temperatura \leftarrow TI$

$grade \leftarrow \text{CriaGradeInicial}(LP, LT, MA)$

$minConflitos \leftarrow \text{NumeroConflitos}(grade)$

while condição de parada não atingida **do**

for passo = 0 até numeroPassos **do**

$turma \leftarrow \text{EscolheTurmaAleatoria}()$

$linhas \leftarrow \text{EscolheHorariosAleatoriosValidos}(sala)$

$\delta \leftarrow \text{CalculaDelta}(sala, linhas)$

$probabilidade \leftarrow e^{-\delta/temperatura}$

$valorAceite \leftarrow \text{Aleatorio}(0, 1)$

if $\delta < 0$ ou $probabilidade \geq valorAceite$ **then**

$\text{PermutaProfessores}(sala, linha1, linha2)$

if $\text{NumeroConflitos}(grade) < minConflitos$ **then**

$\text{Imprime}(grade)$

$minConflitos \leftarrow \text{NumeroConflitos}(grade)$

end

end

end

$temperatura \leftarrow temperatura * TR$

end

No algoritmo 1, o valor da taxa de resfriamento é de 0,99, sendo a temperatura inicial e o número de passos por iteração escolhidos empiricamente. Além disso, durante o desenvolvimento deste primeiro incremento, utilizou-se como condição de parada o esgotamento da temperatura, ou seja, o algoritmo finaliza sua execução assim que a temperatura atinge um valor próximo de zero, quando não ocorrem mais permutações de professores.

Explicando melhor o algoritmo, o método “CriaGradeInicial” gera uma matriz com as turmas e números de aulas de cada professor alocados corretamente. Esta grade inicial provavelmente possui inúmeros conflitos, portanto são aplicados os passos de otimização.

Para cada passo de otimização, são escolhidas aleatoriamente uma turma e duas linhas (posições) da grade horária. Com estas informações, é calculada a variação do número de conflitos que a permutação dos professores nas linhas escolhidas ocasionaria. De acordo com este valor de variação, é determinado se a troca dos professores deve ou não ser realizada: uma troca que diminua o número de conflitos sempre é aceita, enquanto uma troca que aumenta o número de conflitos pode ser aceita probabilisticamente, de acordo com o valor da temperatura na iteração atual.

3.3.2 PESOS E RESTRIÇÕES

Apesar da importância da resolução de conflitos, existem diversas outras nuances durante o planejamento das grades horárias que precisam ser levadas em conta para que as grades produzidas sejam aplicáveis na prática.

Para possibilitar a otimização simultânea de várias características das grades, optou-se por utilizar um sistema de métricas, cada qual mensura quantitativamente determinada característica do horário, e contém um peso que define a sua importância para a grade como um todo. Com estas métricas, para obter o custo de determinada solução, ou seja, a representação da quantidade de problemas que apresenta, basta tomar a média ponderada das métricas.

Adicionalmente, as métricas podem ser rígidas ou suaves. As métricas rígidas precisam ser perfeitamente atendidas para que a grade horária seja considerada viável, enquanto as métricas suaves promovem a melhoria da qualidade, mas não necessariamente precisam ser perfeitamente atendidas.

As próximas seções terão como objetivo explicar cada uma destas métricas e os desafios associados. A [Subsubseção 3.3.2.8](#) entrará em mais detalhes sobre como as grades horárias passaram a ser salvas após a implementação das métricas.

3.3.2.1 Agrupamento de aulas

Observando grades horárias escolares existentes, é possível notar que existe uma motivação para realizar agrupamentos, formando aulas duplas. Isso aumenta a produtividade das aulas, à medida que diminui trocas e deslocamento de professores entre as salas. Para produzir estes agrupamentos, foram adicionadas algumas métricas que fazem o otimizador penalizar:

1. Aulas separadas;
2. Aulas desagrupadas;
3. Excessos de aulas iguais para determinada turma em um dia;
4. Dias com todas aulas planejadas distintas para determinada turma;

Para contextualizar estas situações, a seguir serão apresentados alguns quadros.

Quadro 1 – Exemplo de dia com aulas separadas.

Aula	Professor	Matéria
1	Marcos	Matemática
2	Fábio	Física
3	Marcos	Matemática
4	Luciana	Português
5	Luciana	Português
6	Luciana	Português

Fonte: Autoria própria

Observando o quadro 1, é possível notar:

- Aulas separadas: 1, 2 e 3, visto que não estão em nenhum agrupamento;
- Aulas desagrupadas: 1 e 3, pois consistem em múltiplas aulas iguais, no mesmo dia, que não foram agrupadas;
- Excessos de aulas: 4, 5, 6, pois consiste em uma aula tripla, algo considerado não desejável neste trabalho

Por fim, a situação de dias com todas aulas diferentes, que serão referidos como "Dias fragmentados" neste trabalho, pode ser observada no quadro 2.

Quadro 2 – Exemplo de dia fragmentado.

Aula	Professor	Matéria
1	Marcos	Matemática
2	Fábio	Física
3	Luciana	Português
4	Roberto	Geografia
5	Renato	História


Fonte: Autoria própria

3.3.2.2 Constantes

Durante o desenvolvimento, concebeu-se o conceito de "Aulas constantes", como aulas que absolutamente devem ser alocadas em determinada posição da grade horária. Isto é útil para guiar o otimizador rumo a uma solução desejada, quando já são conhecidas algumas aulas que devem ser fixas.

Como exemplo de caso de uso, uma escola com seis aulas diárias pode ter alguns dias da semana com menos aulas, e pode ser interessante definir explicitamente quais dias devem ter a última aula da grade horária não alocada (vazia). A figura 1 mostra um exemplo de configuração de aulas constantes para determinada grade horária.

Figura 1 – Aulas constantes

Constantes 						
Dia	Aula	6 Ano	1 EM	7 Ano	8 Ano	9 Ano
Segunda-feira	1	Verônica				
	2	Verônica				
	3					
	4					
	5					
	6					
Terça-feira	1	Luciana	Adriana			
	2	Luciana	Adriana			
	3					
	4					
	5					
	6					
Quarta-feira	1					
	2					
	3					
	4					
	5					
	6					
Quinta-feira	1					
	2					
	3					
	4					
	5					
	6					
Sexta-feira	1					
	2					
	3					
	4					
	5					
	6					

Fonte: Autor

Considerando a natureza absoluta das aulas constantes, estas não são consideradas um métrica de qualidade, e não possuem um peso próprio. Em vez disso, no método “Gera-GradelInicial” do algoritmo 1, o otimizador já aloca as aulas constantes nas posições desejadas, e a função “EscolheHorariosAleatoriosValidos” não seleciona posições que estejam alocadas com aulas constantes. Desta forma, um vez que as aulas constantes são posicionadas na grade horária, estas nunca são movidas durante os passos de otimização.

3.3.2.3 Restrições

As restrições representam o oposto das aulas constantes: posições em que determinadas aulas não devem ser alocadas. Implementaram-se no otimizador restrições suaves e rígidas,

cada uma podendo também receber um peso customizado. Dessa forma, é possível configurar o otimizador para nunca alocar determinada aula em certa posição da grade, ou apenas evitar isso.

Como exemplo de uso dessa funcionalidade, a figura 2 demonstra uma configuração de restrições para determinado professor que não pode ser alocado nas duas últimas aulas de qualquer dia da grade horária.

Figura 2 – Restrições

Restrições ☒ Proibir ☐ Evitar

Dia	Aula	6 Ano	1 EM	7 Ano	8 Ano	9 Ano
Segunda-feira	1					
	2					
	3					
	4					
	5					
	6					
Terça-feira	1					
	2					
	3					
	4					
	5					
	6					
Quarta-feira	1					
	2					
	3					
	4					
	5					
	6					
Quinta-feira	1					
	2					
	3					
	4					
	5					
	6					
Sexta-feira	1					
	2					
	3					
	4					
	5					
	6					

No caso das restrições, a métrica associada mensura a quantidade de restrições violadas, ou seja, posições em que determinada aula foi alocada, mas não deveria ter sido.

3.3.2.4 Regiões

O conceito de regiões foi concebido como uma forma de proporcionar ainda mais controle ao usuário, sobre o posicionamento das aulas na grade horária. Cada região consiste em um grupo arbitrário de posições da grade horária, associado a uma regra relacionada a uma quantidade de aulas. Com as regiões, é possível determinar mínimos, máximos ou quantidades exatas de aulas que devem ser alocadas em certas posições da grade horária.

Como exemplo de uso das regiões, a figura 3 demonstra uma configuração utilizada para assegurar que em todos os dias da grade horária, o professor tenha alguma aula alocada no primeiro horário.

Figura 3 – Exemplo de região

Região (Verônica - Exatamente 5 aulas) 🗑

Dia	Aula	6 Ano	1 EM	7 Ano	8 Ano	9 Ano
Segunda-feira	1					
	2					
	3					
	4					
	5					
	6					
Terça-feira	1					
	2					
	3					
	4					
	5					
	6					
Quarta-feira	1					
	2					
	3					
	4					
	5					
	6					
Quinta-feira	1					
	2					
	3					
	4					
	5					
	6					
Sexta-feira	1					
	2					
	3					
	4					
	5					
	6					

Fonte: Autor

A região configurada na 3 pode ser interpretada da seguinte forma: a professora “Verônica” deve ter exatamente cinco aulas alocadas dentro das posições marcadas pela cor azul. Como o otimizador não permite conflitos, cada uma das cinco aulas deverá ser alocada em um dos diferentes dias, garantindo que a professora terá uma aula agendada no primeiro horário de cada um dos dias.

Neste caso, a métrica associada mensura o erro das regiões, ou seja, a diferença entre

O efeito da configuração do grupo de alinhamento da figura 4, pode ser observado na grade horária final gerada na figura 5, com as aulas corretamente alocadas em horários simultâneos.

Figura 5 – Efeito de grupo de alinhamento

Dia	Aula	6 Ano		7 Ano	
Segunda-feira	1	Jack	Português	Renata	História
	2	Jack	Português	Renata	História
	3	Renata	História	Verônica	Geografia
	4	Verônica	Geografia	Adriana	Ciências
	5	Verônica	Geografia	Adriana	Ciências
	6				
Terça-feira	1	Jack	Português	Jéssica	Artes
	2	Jack	Português	Jéssica	Artes
	3	Jéssica	Inglês	Rosa	Matemática
	4	Rosa	Matemática	Luciana	Of. Texto
	5	Rosa	Matemática	Beatriz	ED. Física
	6	Beatriz	ED. Física		
Quarta-feira	1	Renata	História	Jack	Português
	2	Renata	História	Jack	Português
	3	Rosa	Matemática	Renata	História
	4	Luciana	Of. Texto	Rosa	Matemática
	5	Verônica	Geografia	Adriana	Ciências
	6				
Quinta-feira	1	Adriana	Ciências	Jack	Espanhol
	2	Adriana	Ciências	Jack	Português
	3	Jack	Português	Jéssica	Filosofia
	4	Jéssica	Inglês	Rosa	Matemática
	5	Jéssica	Filosofia	Verônica	Geografia
	6			Verônica	Geografia
Sexta-feira	1	Rosa	Matemática	Jéssica	Inglês
	2	Rosa	Matemática	Jéssica	Inglês
	3	Jack	Espanhol	Rosa	Matemática
	4	Adriana	Ciências	Rosa	Matemática
	5	Cristiane	Artes	Jack	Português
	6	Cristiane	Artes	Jack	Português

Fonte: Autor

Implementaram-se duas métricas de qualidade relacionadas aos grupos de alinhamento, uma que mensura a quantidade de grupos formados e a quantidade de aulas “desalinhadas”, ou seja, aulas que deveriam ser agendadas no mesmo horário, mas que não foram.

No caso da 5, foram formados dois grupos (demarcados em vermelho), e nenhuma aula ficou desalinhada, ou seja, todas as seis aulas configuradas no grupo de alinhamento foram corretamente alocadas em horários simultâneos.

3.3.2.6 Janelas

As janelas consistem em situações em que a jornada de trabalho de determinado professor apresenta um horário sem aulas alocadas, fazendo com que o docente fique ocioso. O quadro 3 exemplifica esta situação.

Quadro 3 – Exemplo de dia com janela.

Aula	6ºAno	7ºAno	8ºAno
1	Jéssica	Rosa	Adriana
2	Jéssica	Rosa	Adriana
3	Fábio	Jéssica	Rosa
4	Adriana	Jéssica	Rosa
5	Beatriz	Adriana	Jéssica

Fonte: Autoria própria

No quadro 3, a professora Adriana tem uma janela na terceira aula, visto que tem aulas alocadas antes e depois (aulas 1, 2, 4 e 5). Em contrapartida, apesar de não ter nenhuma aula planejada no quinto horário, Rosa não tem janelas neste exemplo, pois pode encerrar sua jornada de trabalho na quarta aula.

A métrica de janelas mensura o número de horários na grade que apresentam estes horários ociosos, para cada professor.

3.3.2.7 Preferências

As preferências consistem em configurações que informam ao otimizador características preferíveis para a alocação das aulas de cada professor. Implementaram-se os seguintes tipos de preferência no otimizador:

1. Preferência de primeiras aulas;
2. Preferência de últimas aulas;
3. Preferência específica de última aula

A Figura 6 representa um exemplo de configuração de preferências para determinado professor. Neste caso, a configuração informa ao otimizador que na segunda-feira, as aulas de “Verônica” devem ser alocadas desde o início do dia, não ultrapassando um total de aulas; e que na turma do “6º Ano”, caso haja uma aula dessa professora, esta deve ser alocada na última aula.

Vale ressaltar que o diferencial da preferência específica de última aula é que esta considera a possibilidade de aulas vazias, ou seja, caso o último horário do dia não tenha uma aula alocada, a preferência considera que a aula deve ser alocada na penúltima aula.

Figura 6 – Configuração de preferências

Preferências					Verônica ▾
Dia	6 Ano	7 Ano	8 Ano	9 Ano	1 EM
Segunda-feira	2 primeiras aulas	2 primeiras aulas	2 primeiras aulas	2 primeiras aulas	2 primeiras aulas
Terça-feira	Forçar última aula				
Quarta-feira					
Quinta-feira					
Sexta-feira					

Fonte: Autor

3.3.2.8 Armazenamento de soluções

Com a adição das métricas, o critério para considerar uma grade horária como uma solução válida ficou mais estrito, e em muitos casos tornou-se impossível obter uma grade que atenda perfeitamente todas as nuances. Tendo isto em mente, foi necessário implementar uma funcionalidade de armazenamento de grades horárias um pouco mais detalhada. O resultado disso, pode ser visto no algoritmo [2](#).

Algoritmo 2: Otimizador com persistência de grades horárias

Input: Lista de professores LP , lista de turmas LT , matriz de aulas por professor por turma MA , temperatura inicial TI , Taxa de resfriamento TR

Output: Grade horária de professores otimizada

$temperatura \leftarrow TI$

$grade \leftarrow \text{CriaGradeInicial}(LP, LT, MA)$

$iteracoesSemAlteracao \leftarrow 0$

$solucoes \leftarrow$ lista vazia

while condição de parada não atingida **do**

$deltaTotal \leftarrow 0$

for passo = 0 até numeroPassos **do**

$turma \leftarrow grade.EscolheTurmaAleatoria()$

$linhas \leftarrow grade.EscolheHorariosAleatoriosValidos(sala)$

$delta \leftarrow grade.CalculaDelta(sala, linhas)$

$probabilidade \leftarrow e^{-delta/temperatura}$

$valorAceite \leftarrow \text{Aleatorio}(0, 1)$

if $delta < 0$ ou $probabilidade \geq valorAceite$ **then**

$grade.PermutaProfessores(sala, linha1, linha2)$

$deltaTotal \leftarrow deltaTotal + delta$

if grade não existe na lista de soluções **then**

 insere grade na lista de soluções

 limita lista de soluções às 100 melhores grades

end

end

end

if $delta = 0$ **then**

$iteracoesSemAlteracao \leftarrow iteracoesSemAlteracao + 1$

else

$iteracoesSemAlteracao \leftarrow 0$

end

if $iteracoesSemAlteracao \geq 15$ **then**

$salvaGradesRelevantes()$

 apaga lista de soluções

$temperatura \leftarrow TI$

$iteracoesSemAlteracao \leftarrow 0$

end

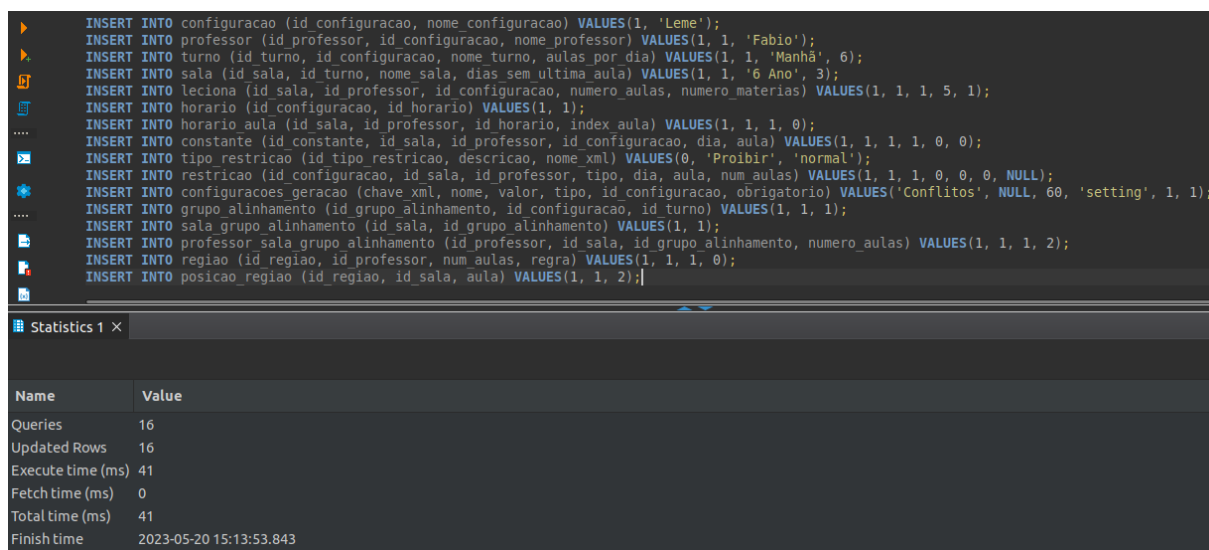
$temperatura \leftarrow temperatura * TR$

end

Na [Figura 7](#), é possível notar a presença das entidades elacionadas às diversas métricas de qualidade comentadas na [Subseção 3.3.2](#), como restrições, contrantes, regiões, grupos de alinhamento, entre outras.

Para validar a modelagem do banco de dados, realizaram-se inserções de informações de exemplo nas diferentes tabelas. A [Figura 8](#) mostra algumas dessas inserções e os vínculos instituídos pelos identificadores escolhidos.

Figura 8 – Consultas de validação da modelagem



```

INSERT INTO configuracao (id configuracao, nome configuracao) VALUES(1, 'Leme');
INSERT INTO professor (id professor, id configuracao, nome professor) VALUES(1, 1, 'Fabio');
INSERT INTO turno (id turno, id configuracao, nome turno, aulas por dia) VALUES(1, 1, 'Manhã', 6);
INSERT INTO sala (id sala, id turno, nome sala, dias sem ultima aula) VALUES(1, 1, '6 Ano', 3);
INSERT INTO leciona (id sala, id professor, id configuracao, numero aulas, numero materias) VALUES(1, 1, 1, 5, 1);
INSERT INTO horario (id configuracao, id horario) VALUES(1, 1);
INSERT INTO horario aula (id sala, id professor, id horario, index aula) VALUES(1, 1, 1, 0);
INSERT INTO constante (id constante, id sala, id professor, id configuracao, dia, aula) VALUES(1, 1, 1, 1, 0, 0);
INSERT INTO tipo_restricao (id tipo restricao, descricao, nome xml) VALUES(0, 'Proibir', 'normal');
INSERT INTO restricao (id configuracao, id sala, id professor, tipo, dia, aula, num aulas) VALUES(1, 1, 1, 0, 0, 0, NULL);
INSERT INTO configuracoes_geracao (chave xml, nome, valor, tipo, id configuracao, obrigatorio) VALUES('Conflitos', NULL, 60, 'setting', 1, 1);
INSERT INTO grupo_alinhamento (id grupo alinhamento, id configuracao, id turno) VALUES(1, 1, 1);
INSERT INTO sala_grupo_alinhamento (id sala, id grupo alinhamento) VALUES(1, 1);
INSERT INTO professor_sala_grupo_alinhamento (id professor, id sala, id grupo alinhamento, numero aulas) VALUES(1, 1, 1, 2);
INSERT INTO regioao (id regioao, id professor, num aulas, regra) VALUES(1, 1, 1, 0);
INSERT INTO posicao_regiao (id regioao, id sala, aula) VALUES(1, 1, 2);

```

Name	Value
Queries	16
Updated Rows	16
Execute time (ms)	41
Fetch time (ms)	0
Total time (ms)	41
Finish time	2023-05-20 15:13:53.843

Fonte: Autor

Validou-se também o armazenamento das grades horárias no banco de dados. A Figura 9 traz um exemplo de consulta de uma grade horária com sete salas:

Figura 9 – Consulta SQL de grade horária com sete salas

```
SELECT
  CASE (a.n / 6)
    WHEN 0 THEN 'Segunda'
    WHEN 1 THEN 'Terça'
    WHEN 2 THEN 'Quarta'
    WHEN 3 THEN 'Quinta'
    WHEN 4 THEN 'Sexta'
    ELSE NULL
  END AS "DiaSemana",
  (a.n % 6) + 1 AS "Aula",
  p1.nome_professor AS "6Ano",
  p2.nome_professor AS "7Ano",
  p3.nome_professor AS "8Ano",
  p4.nome_professor AS "9Ano",
  p5.nome_professor AS "1EM",
  p6.nome_professor AS "2EM",
  p7.nome_professor AS "3EM"
FROM generate_series(0, 29) AS a(n)
LEFT JOIN horario_aula ha1 ON ha1.id_sala = 2 AND ha1.index_aula = a.n AND ha1.id_horario = 10848
LEFT JOIN professor p1 ON p1.id_professor = ha1.id_professor
LEFT JOIN horario_aula ha2 ON ha2.id_sala = 3 AND ha2.index_aula = a.n AND ha2.id_horario = 10848
LEFT JOIN professor p2 ON p2.id_professor = ha2.id_professor
LEFT JOIN horario_aula ha3 ON ha3.id_sala = 4 AND ha3.index_aula = a.n AND ha3.id_horario = 10848
LEFT JOIN professor p3 ON p3.id_professor = ha3.id_professor
LEFT JOIN horario_aula ha4 ON ha4.id_sala = 5 AND ha4.index_aula = a.n AND ha4.id_horario = 10848
LEFT JOIN professor p4 ON p4.id_professor = ha4.id_professor
LEFT JOIN horario_aula ha5 ON ha5.id_sala = 6 AND ha5.index_aula = a.n AND ha5.id_horario = 10848
LEFT JOIN professor p5 ON p5.id_professor = ha5.id_professor
LEFT JOIN horario_aula ha6 ON ha6.id_sala = 7 AND ha6.index_aula = a.n AND ha6.id_horario = 10848
LEFT JOIN professor p6 ON p6.id_professor = ha6.id_professor
LEFT JOIN horario_aula ha7 ON ha7.id_sala = 8 AND ha7.index_aula = a.n AND ha7.id_horario = 10848
LEFT JOIN professor p7 ON p7.id_professor = ha7.id_professor
ORDER BY a.n;
```

Fonte: Autor

A consulta anterior traz como resultado a tabela visível na Figura 10, com linhas e colunas correspondentes a horários de aulas e salas respectivamente.

Figura 10 – Resultado da consulta de uma grade no banco de dados

DiaSemana	Aula	6Ano	7Ano	8Ano	9Ano	1EM	2EM	3EM
Segunda	1	Renata	Rosa	Cristiane	Luciana	Adriana	Jéssica	Glaucia
Segunda	2	Renata	Rosa	Cristiane	Luciana	Glaucia	Jéssica	Adriana
Segunda	3	Beatriz	Renata	Rosa	Jéssica	Glaucia	Luciana	Adriana
Segunda	4	Cristiane	Renata	Rosa	Adriana	Jéssica	Glaucia	Luciana
Segunda	5	Cristiane	Adriana	Beatriz	Adriana	Jéssica	Glaucia	Luciana
Segunda	6	[NULL]	Beatriz	[NULL]	[NULL]	Luciana	Jéssica	Adriana
Terça	1	Rosa	Luciano	Luciana	Renata	Fabio	Adriana	Jéssica
Terça	2	Rosa	Adriana	Luciana	Fabio	Luciano	Glaucia	Jéssica
Terça	3	Luciano	Adriana	Renata	Fabio	Jéssica	Glaucia	Luciana
Terça	4	Luciano	Renata	Adriana	Jéssica	Fabio	Luciana	Glaucia
Terça	5	Adriana	Jéssica	Luciana	Beatriz	Fabio	Luciano	Glaucia
Terça	6	Adriana	[NULL]	[NULL]	[NULL]	Glaucia	Luciano	Jéssica
Quarta	1	Luciano	Jack	Rosa	Luciana	Adriana	Renata	Jéssica
Quarta	2	Rosa	Jack	Jéssica	Luciano	Adriana	Renata	Luciana
Quarta	3	Jack	Luciano	Jéssica	Fabio	Luciana	Adriana	Renata
Quarta	4	Jéssica	Luciano	Luciana	Fabio	Jack	Adriana	Renata
Quarta	5	Jéssica	Jack	Luciano	Luciana	Glaucia	Fabio	Adriana
Quarta	6	[NULL]	[NULL]	Luciano	Luciana	Glaucia	Jack	Adriana
Quinta	1	Luciana	Jack	Rosa	Renata	Jéssica	Fabio	Adriana
Quinta	2	Jéssica	Jack	Rosa	Renata	Adriana	Luciana	Fabio
Quinta	3	Jack	Rosa	Renata	Jéssica	Fabio	Adriana	Luciana
Quinta	4	Jack	Rosa	Renata	Jéssica	Luciana	Adriana	Fabio
Quinta	5	Adriana	Jéssica	Luciana	Jack	Luciano	Glaucia	Fabio
Quinta	6	[NULL]	[NULL]	Jack	Luciana	Luciano	Adriana	Glaucia
Sexta	1	Jack	Rosa	Luciana	Jéssica	Adriana	Renata	Luciano
Sexta	2	Renata	Jack	Jéssica	Fabio	Adriana	Luciana	Luciano
Sexta	3	Rosa	Jéssica	Luciano	Adriana	Renata	Luciana	Fabio
Sexta	4	Rosa	Luciana	Adriana	Luciano	Renata	Jéssica	Jack
Sexta	5	Jack	Jéssica	Adriana	Luciano	Luciana	Fabio	Glaucia
Sexta	6	Jack	Jéssica	[NULL]	[NULL]	Luciana	Fabio	Luciano

Fonte: Autor

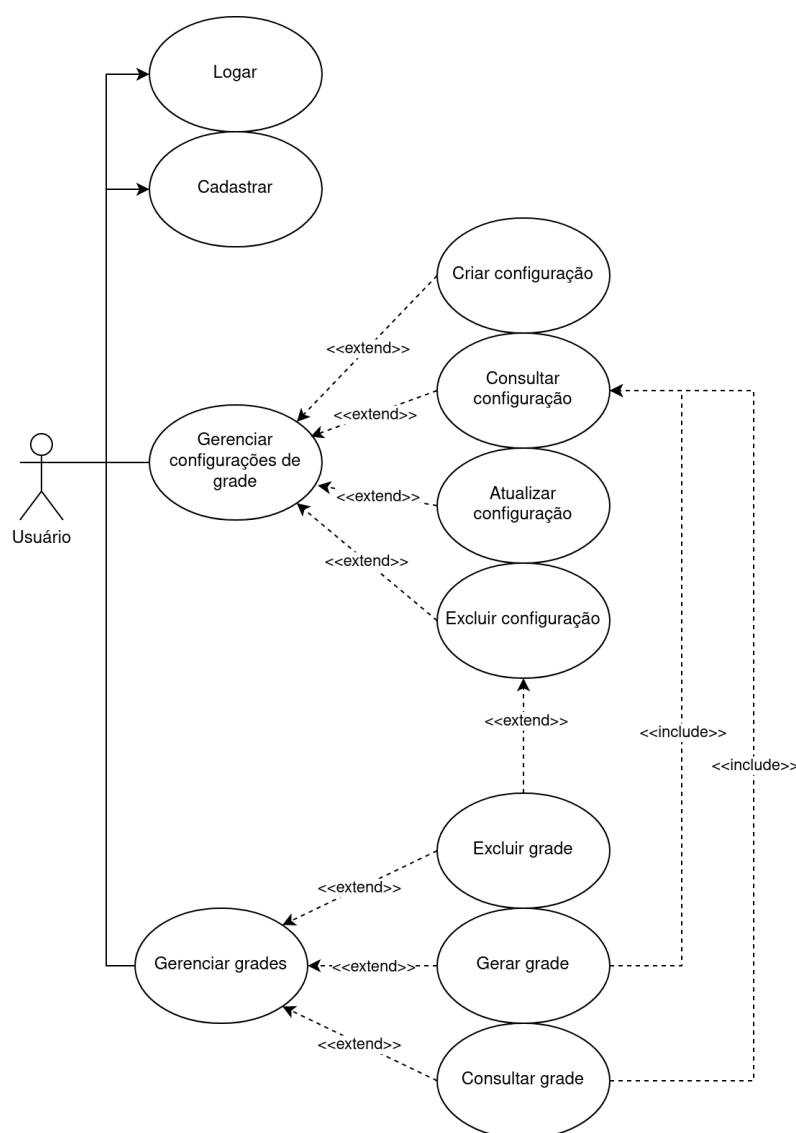
3.3.4 INTERFACE

A interface *web* é responsável pela interação do usuário final com a aplicação. Portanto, implementaram-se funcionalidades que possibilitam a configuração das restrições e características das grades horárias a serem geradas, além de mostrar ou exportar tais grades após a geração.

Para o desenvolvimento deste componente, optou-se pelo framework *Vue.js*, devido à riqueza de seu ecossistema de desenvolvimento *frontend* e familiaridade do graduando com este. Complementando este framework, utilizou-se também o *Vuetify.js*, a fim de facilitar e padronizar o design da aplicação.

A [Figura 11](#) mostra os casos de uso que a interface *web* é responsável por disponibilizar:

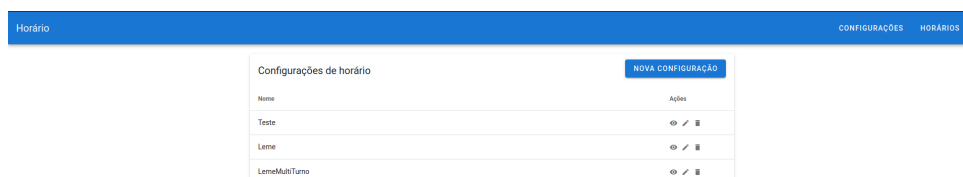
Figura 11 – Diagrama de Casos de Uso












Fonte: Autor

A Figura 12 mostra a tela de listagem de configurações de grade. Como comentado anteriormente, alguns dos casos de uso da aplicação envolvem o gerenciamento de configurações de grades horárias, as quais são listadas nessa tela.

Figura 12 – Tela - Listagem de Configurações de Grade



Configurações de horário		NOVA CONFIGURAÇÃO
Nome	Ações	
Teste	  	
Leme	  	
LemeMultiTurno	  	

Fonte: Autor

A tela mostrada na Figura 13 é responsável por permitir que o usuário cadastre os professores da escola. Nessa tela é possível notar que a aplicação foi estruturada seguindo uma noção de etapas de configuração até a geração da grade horária final. No topo da tela, a etapa “Estrutura da Escola” encontra-se selecionada.

Figura 13 – Tela - Estrutura da Escola - Professores

Professores		NOVO PROFESSOR
Nome	Ações	
Fabio	✍️ 🗑️	
Adriana	✍️ 🗑️	
Luciana	✍️ 🗑️	
Luciano	✍️ 🗑️	
Jack	✍️ 🗑️	
Glauca	✍️ 🗑️	
Renata	✍️ 🗑️	
Beatriz	✍️ 🗑️	
Rosa	✍️ 🗑️	
Jessica	✍️ 🗑️	
Cristiane	✍️ 🗑️	

Turnos			NOVO TURNO
Nome	Aulas por dia	Ações	
.. ..	-	-	

Fonte: Autor

Na Figura 14, tem-se a continuação da tela de estrutura da escola, mostrada na Figura 13. Esta parte da tela é responsável pela configuração das turmas e turnos da escola.

Figura 14 – Tela - Estrutura da Escola - Turmas e Turnos

Horário

CONFIGURAÇÕES

HORÁRIOS

Cristiane

Turnos

NOVO TURNO

Nome	Aulas por dia	Ações
Manhã	5	<div></div>

Informações do turno: Manhã

Salas

NOVA SALA

Nome	Salas sem última aula	Ações
6 Ano	3	<div></div>

SALVAR

PRÓXIMA ETAPA

GERAR DADOS EXEMPLO

Fonte: Autor

Na tela da Figura 15, o usuário pode realizar a configuração de número de aulas que cada professor deve ministrar em cada sala, informação fundamental para a geração das grades horárias.

Figura 15 – Tela - Aulas por professor

Horário

CONFIGURAÇÕES

HORÁRIOS

1 Estrutura da Escola

2 Aulas/Matérias

3 Aulas constantes

4 Restrições

5 Regras de preenchimento

6 Mínimos/Máximos

7 Horários

Informações do turno: Manhã

Aulas por professor

Professor	6 Ano	7 Ano	8 Ano	9 Ano	1 EM	2 EM	3 EM	Total
Fabio	0	0	0	5	4	4	4	17
Adriana	4	3	3	3	6	6	6	31
Luciana	1	1	6	6	5	5	5	29
Luciano	3	3	3	3	3	2	3	20
Jack	6	6	1	1	1	1	1	17
Glauucia	0	0	0	0	5	5	5	15
Renata	3	3	3	3	2	3	2	19
Betriz	1	1	1	1	0	0	0	4
Rosa	5	5	5	0	0	0	0	15
Jessica	3	5	3	5	4	4	4	28
Cristiane	2	0	2	0	0	0	0	4

Fonte: Autor

A tela da [Figura 16](#) é responsável pela configuração das restrições e preferências. Nesta, o usuário pode configurar horários na grade que devem ser evitados ou proibidos para determinado docente.

Figura 16 – Tela - Restrições e Preferências

Fonte: Autor

Já a tela da [Figura 17](#) possibilita a configuração das regiões e grupos de alinhamento:

Figura 17 – Tela - Regiões e Grupos de Alinhamento

Fonte: Autor

A última tela relacionada a configurações pode ser vista na [Figura 18](#), e permite a definição de quantidade mínimas e máximas de aulas para cada professor em cada dia da grade horária:

A etapa final no fluxo da aplicação é representada pela tela da [Figura 19](#). Nesta, o usuário pode requisitar a geração da grade horária utilizando as configurações realizadas nas etapas anteriores, e acessar as grades geradas anteriormente.

Figura 18 – Tela - Mínimos e Máximos

Horário											CONFIGURAÇÕES	SAIR
<div> Estrutura da Escola Aulas/Matérias Aulas constantes Restrições Regras de preenchimento Mínimos/Máximos Horários </div>												
Mínimos						Limites diários						
Professor	Segunda	Terça	Quarta	Quinta	Sexta	Professor	Segunda	Terça	Quarta	Quinta	Sexta	
Verônica	0	0	0	0	0	Verônica	0	0	0	0	0	
Adriana	0	0	0	0	0	Adriana	0	0	0	0	0	
Luciana	0	0	0	0	0	Luciana	0	0	0	0	0	
Luciano	0	0	0	0	0	Luciano	0	0	0	0	0	
Jack	0	0	0	0	0	Jack	0	0	0	0	0	
Glaucia	0	0	0	0	0	Glaucia	0	0	0	0	0	
Renata	0	0	0	0	0	Renata	0	0	0	0	0	
Beatriz	0	0	0	0	0	Beatriz	0	0	0	0	0	
Rosa	0	0	0	0	0	Rosa	0	0	0	0	0	
Jéssica	0	0	0	0	0	Jéssica	0	0	0	0	0	
Cristiane	0	0	0	0	0	Cristiane	0	0	0	0	0	
Fabio	0	0	0	0	0	Fabio	0	0	0	0	0	

SALVAR

Fonte: Autor

Figura 19 – Tela - Horários

Horário

CONFIGURAÇÕES

HORÁRIOS

1

Estrutura da Escola

2

Aulas/Matérias

3

Aulas constantes

4

Restrições

5

Regras de preenchimento

6

Mínimos/Máximos

7

Horários

Horário (10832)

Manhã

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

1001

1002

1003

1004

1005

1006

1007

1008

1009

1010

1011

1012

1013

1014

1015

1016

1017

1018

1019

1020

1021

1022

1023

1024

1025

1026

1027

1028

1029

1030

1031

1032

1033

1034

1035

1036

1037

1038

1039

1040

1041

1042

1043

1044

1045

1046

1047

1048

1049

1050

1051

1052

1053

1054

1055

1056

1057

1058

1059

1060

1061

1062

1063

1064

1065

1066

1067

1068

1069

1070

1071

1072

1073

1074

1075

1076

1077

1078

1079

1080

1081

1082

1083

1084

1085

1086

1087

1088

1089

1090

1091

1092

1093

1094

1095

1096

1097

1098

1099

1100

1101

1102

1103

1104

1105

1106

1107

1108

1109

1110

1111

1112

1113

1114

1115

1116

1117

1118

1119

1120

1121

1122

1123

1124

1125

1126

1127

1128

1129

1130

1131

1132

1133

1134

1135

1136

1137

1138

1139

1140

1141

1142

1143

1144

1145

1146

1147

1148

1149

1150

1151

1152

1153

1154

1155

1156

1157

1158

1159

1160

1161

1162

1163

1164

1165

1166

1167

1168

1169

1170

1171

1172

1173

1174

1175

1176

1177

1178

1179

1180

1181

1182

1183

1184

1185

1186

1187

1188

1189

1190

1191

1192

1193

1194

1195

1196

1197

1198

1199

1200

1201

1202

1203

1204

1205

1206

1207

1208

1209

1210

1211

1212

1213

1214

1215

1216

1217

1218

1219

1220

1221

1222

1223

1224

1225

1226

1227

1228

1229

1230

1231

1232

1233

1234

1235

1236

1237

1238

1239

1240

1241

1242

1243

1244

1245

1246

1247

1248

1249

1250

1251

1252

1253

1254

1255

1256

1257

1258

1259

1260

1261

1262

1263

1264

1265

1266

1267

1268

1269

1270

1271

1272

1273

1274

1275

1276

1277

1278

1279

1280

1281

1282

1283

1284

1285

1286

1287

1288

1289

1290

1291

1292

1293

1294

1295

1296

1297

1298

1299

1300

1301

1302

1303

1304

1305

1306

1307

1308

1309

1310

1311

1312

1313

1314

1315

1316

1317

1318

1319

1320

1321

1322

1323

1324

1325

1326

1327

1328

1329

1330

1331

1332

1333

1334

1335

1336

1337

1338

1339

1340

1341

1342

1343

1344

1345

1346

1347

1348

1349

1350

1351

1352

1353

1354

1355

1356

1357

1358

1359

1360

1361

1362

1363

1364

1365

1366

1367

1368

1369

1370

1371

1372

1373

1374

1375

1376

1377

1378

1379

1380

1381

1382

1383

1384

1385

1386

1387

1388

1389

1390

1391

1392

1393

1394

1395

1396

1397

1398

1399

1400

1401

1402

1403

1404

1405

1406

1407

1408

1409

1410

1411

1412

1413

1414

1415

1416

1417

1418

1419

1420

1421

1422

1423

1424

1425

1426

1427

1428

1429

1430

1431

1432

1433

1434

1435

1436

1437

1438

1439

1440

1441

1442

1443

1444

1445

1446

1447

1448

1449

1450

1451

1452

1453

1454

1455

1456

1457

1458

1459

1460

1461

1462

1463

1464

1465

1466

1467

1468

1469

1470

1471

1472

1473

1474

1475

1476

Fonte: Autor

3.3.5 USUÁRIOS E AUTENTICAÇÃO

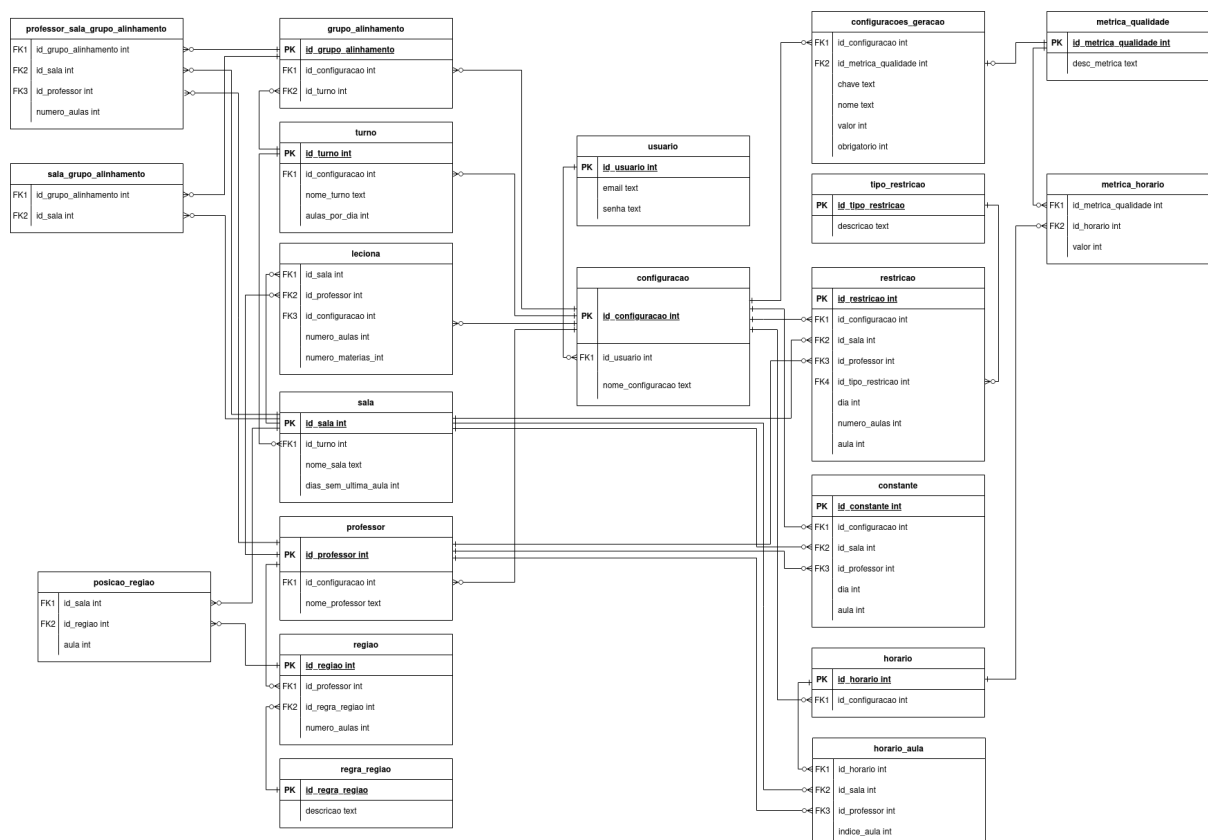
Para controlar a visibilidade de informações no sistema desenvolvido, implementou-se um sistema de usuários. A melhoria consistiu na:

1. Criação de tabela de usuários no banco de dados;
2. Associação da tabela “configuracao” com a tabela de usuários, para que fosse possível armazenar o usuário responsável por cada configuração;
3. Criação da tela de login na interface web;
4. Criação das rotas de cadastro e *login* no servidor
5. Criação do *middleware* de autenticação
6. Criação do *middleware* de validação da configuração

3.3.5.1 Adaptação do banco de dados

A aplicação dos itens 1 e 2 foi realizada diretamente no banco de dados, através da criação da tabela mencionada e a chave estrangeira possibilitando a associação de cada usuário a múltiplas configurações. Após estas alterações, o diagrama entidade relacionamento do banco de dados passa a ser representado pela figura 20.

Figura 20 – Modelo Entidade-Relacionamento com adição da tabela de usuários



Fonte: Autor

3.3.5.2 Tela de login

Como pode ser visto na figura 21, desenvolveu-se uma tela simples de *login*, a qual também desempenha a função de cadastro de novos usuários.

Figura 21 – Tela de Acesso

A imagem mostra a interface de login de um sistema web. No topo, há uma barra azul com o texto "Horário" à esquerda e "CONFIGURAÇÕES" à direita. Centralizado na página, há um formulário branco com o título "Bem vindo". Abaixo do título, há dois campos de entrada: "E-mail" e "Senha". Na base do formulário, há dois botões: "ENTRAR" (cinza) e "CRIAR MINHA CONTA" (azul).

Fonte: Autor

3.3.5.3 Rotas de autenticação

Para a implementação das rotas de cadastro e *login*, utilizou-se além do *framework ExpressJS*, os pacotes *JWT (Json Web Token)* e *bcrypt*. O pacote *JWT* é utilizado para gerar *tokens*, que são enviados para a interface web e podem ser utilizados para autenticar os usuários; já o *bcrypt* é utilizado para gerar e validar as *hashes* das senhas dos usuários, para que nunca sejam armazenadas senhas em texto pleno no banco de dados.

A função utilizada pela nova rota de cadastro pode vista na figura 22. A função “register” realiza a validação dos parâmetros, e cria um usuário no banco de dados, caso já não exista outro com o mesmo e-mail. Além disso, a função retorna um *token JWT* para a interface web, para que seja possível verificar a autenticidade das próximas requisições realizadas pelo usuário.

Figura 22 – Método de Cadastro

```
const register = async (req, res) => {
  const { email, senha } = req.body;
  if (email == null || senha == null) {
    return res.status(400).json({ message: "E-mail e senha são obrigatórios" });
  }

  const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
  if (!emailRegex.test(email)) {
    return res.status(400).json({ message: "E-mail inválido" });
  }

  if (senha.length < 6) {
    return res
      .status(400)
      .json({ message: "A senha deve conter pelo menos 6 caracteres" });
  }

  const resUsuario = await pool.query(
    "select 1 from usuario where email = $1",
    [email]
  );
  if (resUsuario.rows.length > 0) {
    return res.status(400).json({
      message: "Este e-mail já está sendo utilizado por outro usuário",
    });
  }

  const saltRounds = 10;
  const hash = await bcrypt.hash(senha, saltRounds);

  const resUsuarioInserido = await pool.query(
    "insert into usuario (email, senha) values ($1, $2) returning id_usuario",
    [email, hash]
  );
  const id_usuario = resUsuarioInserido.rows[0].id_usuario;

  const secret = process.env.JWT_SECRET;
  const token = jwt.sign({ id_usuario }, secret);
  return res
    .status(200)
    .json({ token, message: "Usuário registrado com sucesso" });
};
```

Fonte: Autor

A função de *login*, visível na figura 23 é similar, realizando validação dos parâmetros, autenticação do usuário através da comparação de senhas utilizando o pacote *bcrypt* e geração de *token* JWT. Vale citar que em ambas as rotas de autenticação, é inserido no *payload* do *token* o identificador do usuário, que posteriormente pode ser utilizado pelos *middlewares* para a realização de validações.

Figura 23 – Método de Login

```
const login = async (req, res) => {
  const { email, senha } = req.body;
  if (email == null || senha == null) {
    return res.status(400).json({ message: "E-mail e senha são obrigatórios" });
  }

  const resUsuario = await pool.query(
    "select id_usuario, senha from usuario where email = $1",
    [email]
  );
  if (resUsuario.rows.length === 0) {
    return res.status(400).json({ message: "Credenciais inválidas" });
  }

  const { senha: hash, id_usuario } = resUsuario.rows[0];
  const correctPassword = await bcrypt.compare(senha, hash);

  if (!correctPassword) {
    return res.status(400).json({ message: "Credenciais inválidas" });
  }

  const secret = process.env.JWT_SECRET;
  const token = jwt.sign({ id_usuario }, secret);

  return res
    .status(200)
    .json({ token, message: "Login realizado com sucesso" });
};
```

Fonte: Autor

3.3.5.4 Middlewares

Foram criadas duas funções *middleware* relacionadas aos usuários. A primeira é responsável por assegurar que apenas usuários propriamente autenticados tenham acesso aos recursos protegidos do sistema. Como pode ser visto na figura 24, essa verificação é realizada através da verificação da presença de um *token* JWT válido no *header* "authorization" da requisição.

Figura 24 – Middleware de autenticação

```
const jwt = require("jsonwebtoken");

function authenticateToken(req, res, next) {
  const authHeader = req.headers["authorization"];
  const token = authHeader && authHeader.split(" ")[1];

  if (token == null) return res.sendStatus(401);

  jwt.verify(token, process.env.JWT_SECRET, (err, data) => {
    if (err || data.id_usuario == null) return res.sendStatus(403);
    req.id_usuario = data.id_usuario;
    next();
  });
}

module.exports = authenticateToken;
```

Fonte: Autor

O outro middleware criado tem como objetivo assegurar que um usuário possa consultar apenas as configurações pelas quais seja responsável. Isso é feito utilizando o identificador do usuário, presente no *payload* do *token* JWT, conforme a figura 25. Caso o usuário não tenha um vínculo com a configuração que está tentando acessar, a requisição é bloqueada.

Figura 25 – Middleware de validação da configuração

```
const pool = require("../config/db");

async function validateConfigId(req, res, next) {
  const ids = [req.params.id, req.body.id_configuracao];
  if (ids.every((id) => id == null)) {
    return next();
  }
  const { id_usuario } = req;
  for (let id_configuracao of ids) {
    if (id_configuracao == null) {
      continue;
    }

    const result = await pool.query(
      "select 1 from configuracao where id_configuracao = $1 and id_usuario = $2",
      [id_configuracao, id_usuario]
    );
    if (result.rows.length == 0) {
      return res.sendStatus(401);
    }
  }
  return next();
}

module.exports = validateConfigId;
```

Fonte: Autor

3.3.6 ALOCAÇÃO DE MATÉRIAS

Durante os incrementos anteriores, o otimizador gerava grades horárias que continham apenas os nomes dos professores alocados para aula. Em outras palavras, as grades eram matrizes nas quais as linhas eram os horários disponíveis, as colunas eram as salas e cada posição na matriz representava qual professor deveria ministrar a aula naquele horário.

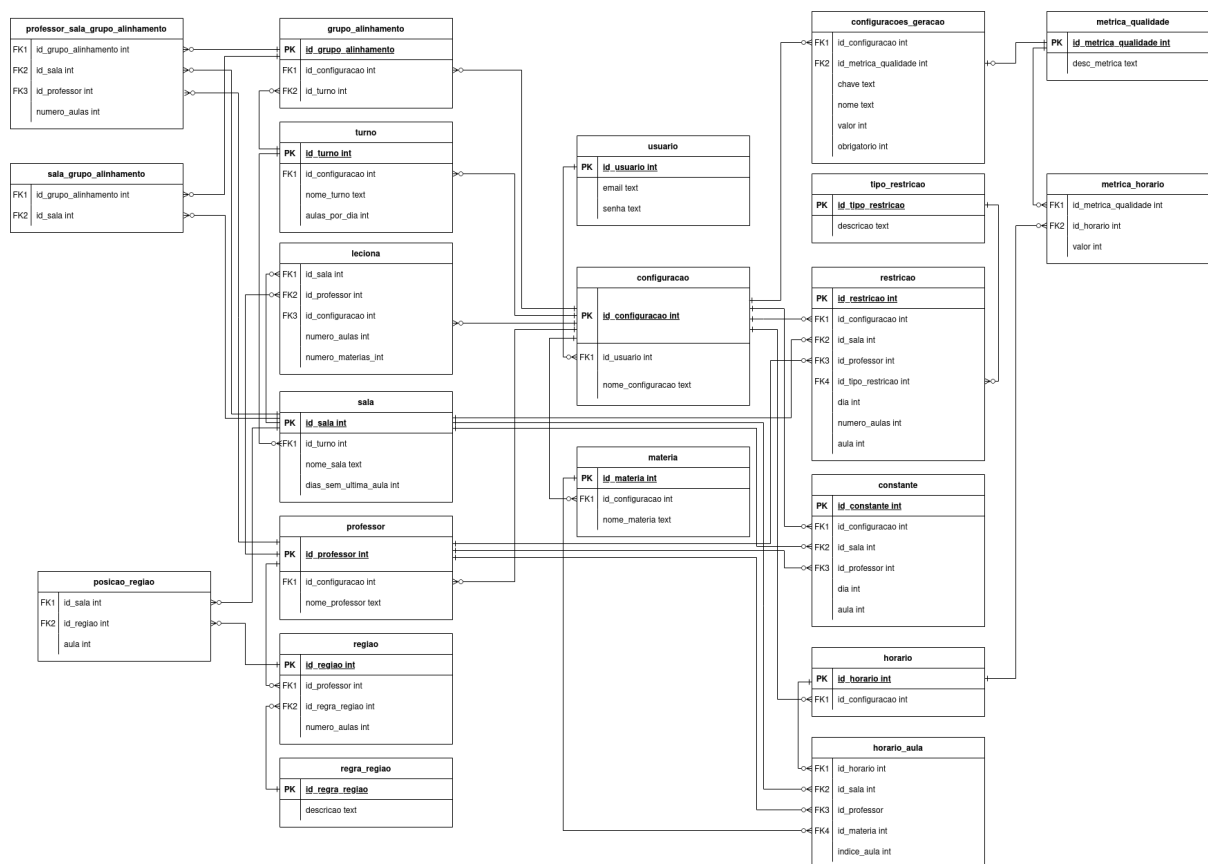
A alocação dos professores é muito importante para a resolução do problema, pois apresenta a maior parte das restrições e dificuldades relacionadas, como os conflitos, por exemplo. Entretanto, em termos de completude, as grades horárias devem ter também a alocação de matérias em cada horário de aula, visto que cada professor pode ministrar aulas de mais de uma matéria. As alterações necessárias para possibilitar isso foram:

1. Alterar a modelagem do banco de dados para comportar informações relacionadas às matérias;
2. Alterar telas da interface web para que fosse possível configurar as matérias;
3. Alterar rotas do servidor para persistir as matérias;
4. Alterar código do otimizador para produzir grades horárias com matérias

3.3.6.1 Alteração no banco de dados

Para armazenar informações relacionadas às matérias, a modelagem do banco de dados foi alterada com a criação de uma tabela para armazenar as matérias, e a adição de uma coluna à tabela “horario” aula para que seja possível especificar matérias alocadas nas grades horárias. Com estas alterações, a modelagem do banco de dados passa a ser representada pela figura [26](#).

Figura 26 – Modelo Entidade-Relacionamento com Matérias

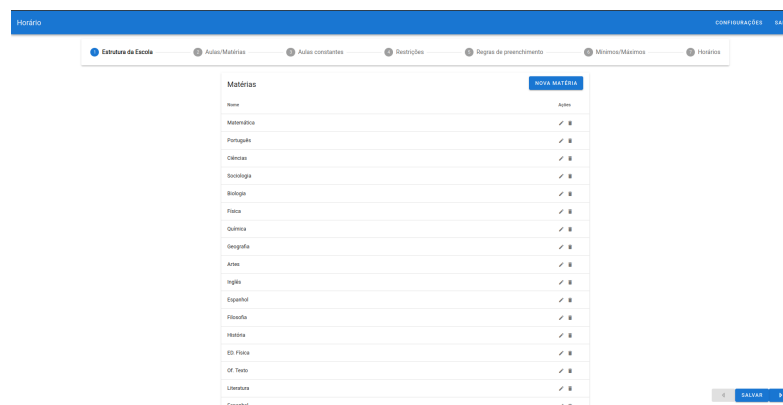


Fonte: Autor

3.3.6.2 Alteração de telas

Com a adição do conceito das matérias ao sistema, algumas telas da interface precisaram ser modificadas. A primeira destas, foi a tela inicial da configuração, ou seja, a tela de "Estrutura da Escola", cuja alteração pode ser vista na figura 27 consistiu na adição de uma seção para cadastro e visualização de matérias, semelhante ao componente de cadastro de professores.

Figura 27 – Estrutura da Escola com Matérias



Fonte: Autor

Além da tela de estrutura, a segunda aba da configuração ("Aulas/Matérias") foi alterada para que fosse possível vincular matérias aos professores, e configurar a quantidade de aulas de cada matéria deve ser ministrada semanalmente, conforme a figura 28.

Figura 28 – Tela de configuração de quantidades de aulas por matéria

Fonte: Autor

Por fim, na tela final da configuração, responsável por exibir as grades horárias geradas, foi necessário alterar o componente da grade para incluir, além dos nomes dos professores, os nomes das matérias alocadas para cada horário, como pode ser visto na figura 29.

Figura 29 – Visualização de grade horária com matérias

Fonte: Autor

3.3.6.3 Alteração de métodos no servidor

A adição das matérias também envolveu algumas alterações no servidor. As rotas alteradas para acomodar a melhoria foram rotas de armazenamento e consulta da estrutura da escola, aulas e grades horárias.

Evidentemente, foi necessário criar rotas também para o gerenciamento de matérias, e as demais rotas do sistema precisaram ter suas funções de tratamento modificadas para interagir corretamente com esta nova tabela.

3.3.6.4 Matérias no otimizador

Após as alterações na interface, servidor e banco de dados, o sistema estava pronto para lidar com as informações relacionadas às matérias, faltando apenas a incorporação destas na geração de grades horárias por parte do otimizador.

Para realizar isso, o otimizador foi alterado para alocar também matérias na grade horária, de forma que enquanto antes a grade horária armazenava em cada posição um professor que foi alocado para aquele horário, agora passa a armazenar uma dupla de professor e matéria alocados. Durante os passos de otimização, estas duplas são permutadas, garantindo assim que cada matéria sempre corresponde a um professor compatível. Com estas alterações, o algoritmo 3 passa a representar o otimizador.

Algoritmo 3: Otimizador com alocação de matérias

Input: Lista de professores LP , lista de turmas LT , matriz de aulas e matérias por professor por turma MA , temperatura inicial TI , Taxa de resfriamento TR

Output: Grade horária de matérias e professores otimizada

$temperatura \leftarrow TI$

$grade \leftarrow \text{CriaGradeInicial}(LP, LT, MA)$

$iteracoesSemAlteracao \leftarrow 0$

$solucoes \leftarrow$ lista vazia

while condição de parada não atingida **do**

$deltaTotal \leftarrow 0$

for $passo = 0$ até $numeroPassos$ **do**

$turma \leftarrow grade.EscolheTurmaAleatoria()$

$duplas \leftarrow grade.EscolheDuplasAleatoriasValidas(turma)$

$delta \leftarrow grade.CalculaDelta(turma, duplas)$

$probabilidade \leftarrow e^{-delta/temperatura}$

$valorAceite \leftarrow \text{ValorAleatorioEntre}(0, 1)$

if $delta < 0$ ou $probabilidade \geq valorAceite$ **then**

$grade.PermutaDuplas(sala, duplas)$

$deltaTotal \leftarrow deltaTotal + delta$

if grade não existe na lista de soluções **then**

 insere grade na lista de soluções

 limita lista de soluções às 100 melhores grades

end

end

end

if $delta = 0$ **then**

$iteracoesSemAlteracao \leftarrow iteracoesSemAlteracao + 1$

else

$iteracoesSemAlteracao \leftarrow 0$

end

if $iteracoesSemAlteracao \geq 15$ **then**

$salvaGradesRelevantes()$

 apaga lista de soluções

$temperatura \leftarrow TI$

$iteracoesSemAlteracao \leftarrow 0$

end

$temperatura \leftarrow temperatura * TR$

end

3.3.7 VALIDAÇÃO DE CONFIGURAÇÕES

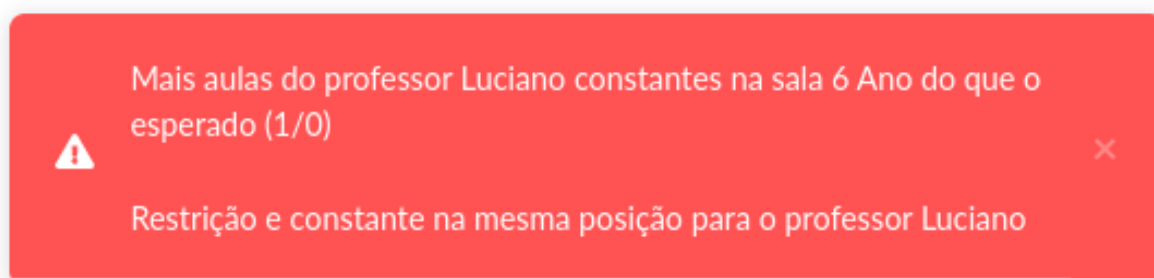
Devido à grande quantidade de configurações necessárias para parametrizar a geração de uma grade horária, erros de configurações por parte dos usuários são inevitáveis. Tendo isto em mente, implementou-se uma seção de código no otimizador, responsável pela validação das informações providenciadas pelo usuário.

Vale ressaltar que esta validação não garante que é possível gerar uma grade horária de acordo com as configurações providas, mas ela deve evitar alguns dos erros mais comuns. As validações realizadas foram:

1. Existem turnos, turmas, professores e matérias configurados;
2. O total de aulas configurado para cada sala deve ser correto, de acordo com o número de dias da grade horária e quantidade de aulas por dia;
3. Para cada sala, nenhum professor pode ter mais aulas agendadas do que a sala acomoda;
4. Nenhum professor pode ter mais aulas constantes configuradas do que o seu total de aulas naquela sala;
5. Não pode haver uma restrição e aula constante para determinado professor na mesma posição da grade horária;
6. Cada grupo de alinhamento deve relacionar duas turmas diferentes, e referenciar o alinhamento da mesma quantidade de aulas em ambas as salas, sendo esta quantidade compatível com o total de aulas configurado para cada professor;
7. O número de aulas esperado para cada região deve ser compatível com o total de aulas configurado para as salas englobadas na região

Estas validações são executadas antes do início das otimizações das grades horárias, e caso haja algum erro de configuração, é exibida uma mensagem alertando o usuário, conforme a [Figura 30](#), e a geração da grade horária é cancelada.

Figura 30 – Mensagem de erro de validação



Fonte: Autor

3.3.8 EXPORTAÇÃO DE GRADES HORÁRIAS

Considerando a necessidade de exportar as grades horárias para fora do sistema, e a sua natureza tabular, implementou-se a funcionalidade de *download* das grades como planilhas Excel, no formato XLSX.

Para possibilitar a exportação neste formato, utilizou-se o pacote *Node ExcelJs*, que providencia a classe “WorkBook”, a qual facilita a construção de arquivos de planilhas Excel utilizando a linguagem *JavaScript*. O resultado final desta implementação pode ser visto na Figura 31.

Figura 31 – Grade horária exportada para planilha

Dia	Aula	6 Ano		7 Ano		8 Ano		9 Ano		1 EM	
Segunda-feira	1	Matemática	Rosa	Português	Jack	Geografia	Verônica	Inglês	Jéssica	Matemática	Glaucia
	2	Matemática	Rosa	Português	Jack	Geografia	Verônica	Inglês	Jéssica	Matemática	Glaucia
	3	Espanhol	Jack	Matemática	Rosa	Português	Luciana	História	Renata	Filosofia	Jéssica
	4	Português	Jack	Matemática	Rosa	Português	Luciana	História	Renata	Artes	Jéssica
	5	Ciências	Adriana	ED. Física	Beatriz	Espanhol	Jack	Português	Luciana	História	Renata
	6	Ciências	Adriana					Português	Luciana	História	Renata
Terça-feira	1	Of. Texto	Luciana	Matemática	Rosa	Geografia	Verônica	Filosofia	Jéssica	P. Vida	Fabio
	2	Geografia	Verônica	Ciências	Adriana	Matemática	Rosa	Português	Luciana	Física	Fabio
	3	Geografia	Verônica	Ciências	Adriana	Matemática	Rosa	Português	Luciana	Espanhol	Jack
	4	Matemática	Rosa	Português	Jack	Of. Texto	Luciana	Geografia	Verônica	Biologia	Adriana
	5	Matemática	Rosa	Português	Jack	Artes	Cristiane	Geografia	Verônica	Biologia	Adriana
	6	ED. Física	Beatriz			Artes	Cristiane			Sociologia	Verônica
Quarta-feira	1	Filosofia	Jéssica	Geografia	Verônica	História	Renata	Ciências	Adriana	Matemática	Glaucia
	2	Artes	Cristiane	Geografia	Verônica	Português	Luciana	Artes	Jéssica	Química	Adriana
	3	Artes	Cristiane	Espanhol	Jack	Português	Luciana	Artes	Jéssica	Química	Adriana
	4	Português	Jack	Artes	Jéssica	Ciências	Adriana	Português	Luciana	Física	Fabio
	5	Português	Jack	Artes	Jéssica	Ciências	Adriana	Of. Texto	Luciana	Física	Fabio
	6					Matemática	Rosa	Matemática	Fabio	Biologia	Adriana
Quinta-feira	1	Geografia	Verônica	Ciências	Adriana	Inglês	Jéssica	Espanhol	Jack	Matemática	Glaucia
	2	Português	Jack	Matemática	Rosa	Inglês	Jéssica	Geografia	Verônica	Matemática	Glaucia
	3	Português	Jack	Inglês	Jéssica	Matemática	Rosa	História	Renata	Geografia	Verônica
	4	História	Renata	Inglês	Jéssica	Matemática	Rosa	Matemática	Fabio	Geografia	Verônica
	5	História	Renata	Geografia	Verônica	ED. Física	Beatriz	Matemática	Fabio	Português	Luciana
	6			História	Renata					Português	Luciana
Sexta-feira	1	História	Renata	Filosofia	Jéssica	Ciências	Adriana	Matemática	Fabio	Of. Texto	Luciana
	2	Inglês	Jéssica	Of. Texto	Luciana	História	Renata	Matemática	Fabio	Química	Adriana
	3	Inglês	Jéssica	Matemática	Rosa	História	Renata	Ciências	Adriana	Português	Luciana
	4	Matemática	Rosa	História	Renata	Filosofia	Jéssica	Ciências	Adriana	Literatura	Luciana
	5	Ciências	Adriana	História	Renata	Português	Luciana	ED. Física	Beatriz	Inglês	Jéssica
	6			Português	Jack					Inglês	Jéssica

Fonte: Autor

Para possibilitar a realização do *download* das grades horárias, acrescentou-se um botão à interface. Adicionalmente, foram adicionados botões para navegar entre as diferentes grades geradas, e exibir mais informações sobre a grade selecionada.

4 CONSIDERAÇÕES FINAIS

Em conclusão, este trabalho abordou o *High School Timetabling Problem*, e a demanda associada por um software de otimização de grades horárias. Verificou-se durante a revisão de literatura a dificuldade que a tarefa de planejamento de grades horárias representa, e que grande parte das instituições de ensino do país ainda realiza essa tarefa manualmente.

O presente trabalho trouxe também algumas demonstrações do estado atual do protótipo da aplicação desenvolvida, assim como melhorias sugeridas para a continuação da pesquisa. Com o desenvolvimento dos primeiros incrementos da aplicação, foi possível verificar a viabilidade do projeto aplicando a técnica de *Simulated Annealing* para produzir e otimizar grades horárias escolares, atendendo múltiplos objetivos tradicionais do problema, assim como a nova preocupação trazida pelo Novo Ensino Médio, os Itinerários Formativos.

Resta portanto como atividade de crucial importância a coleta de dados de potenciais usuários futuros da aplicação, para que seja possível verificar a necessidade de adições e melhorias ao otimizador, e a execução dos incrementos do software a fim de cumprir os objetivos propostos.

Referências

- ABRAMSON, D. Constructing school timetables using simulated annealing: Sequential and parallel algorithms. **Management Science**, v. 37, n. 1, p. 98–113, 1991. Disponível em: <<https://EconPapers.repec.org/RePEc:inm:ormnsc:v:37:y:1991:i:1:p:98-113>>. Citado 2 vezes nas páginas 1 e 3.
- BARDADY, V. A. Computer-aided school and university timetabling: The new wave. In: BURKE, E.; ROSS, P. (Ed.). **Practice and Theory of Automated Timetabling**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996. p. 22–45. ISBN 978-3-540-70682-3. Citado na página 1.
- BRASIL. Lei nº 13.415, de 16 de fevereiro de 2017. **Diário Oficial [da] República Federativa do Brasil**, Brasília, DF, 2017. ISSN 1677-7042. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2017/lei/l13415.htm>. Citado na página 3.
- COOPER, T. B.; KINGSTON, J. H. The complexity of timetable construction problems. In: **International Conference on the Practice and Theory of Automated Timetabling**. [S.l.: s.n.], 1995. Citado na página 3.
- FONSECA, G. H.; SANTOS, H. G.; CARRANO, E. G. Integrating matheuristics and metaheuristics for timetabling. **Computers And Operations Research**, v. 74, p. 108–117, 2016. ISSN 0305-0548. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0305054816300879>>. Citado na página 1.
- LAARHOVEN, P. J. **Simulated annealing theory and applications**. [S.l.]: Kluwer, 1987. Citado na página 4.
- POULSEN, C. J. B. **Desenvolvimento de um Modelo para o School Timetabling Problem Baseado na Meta-Heurística Simulated Annealing**. 141 p. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 2012. Citado na página 1.
- PRESSMAN, R.; MAXIM, B. **Engenharia de Software - 8ª Edição**. [s.n.], 2016. ISBN 9788580555349. Disponível em: <<https://books.google.com.br/books?id=wexzCwAAQBAJ>>. Citado na página 5.
- TAN, J. S. et al. A survey of the state-of-the-art of optimisation methodologies in school timetabling problems. **Expert Systems with Applications**, v. 165, p. 113943, 2021. ISSN 0957-4174. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0957417420307314>>. Citado 2 vezes nas páginas 1 e 3.