

Universidade Federal Fluminense
Instituto de Ciência e Tecnologia
Bacharelado em Ciência da Computação

ANA FLÁVIA NEVES GALL
LEONARDO NEVES GALL

**SISTEMA DE ALOCAÇÃO DE HORÁRIOS EM UM CURSO
UNIVERSITÁRIO DA UNIVERSIDADE FEDERAL FLUMINENSE**

Rio das Ostras – RJ

2017

ANA FLÁVIA NEVES GALL

LEONARDO NEVES GALL

**SISTEMA DE ALOCAÇÃO DE HORÁRIOS EM UM CURSO
UNIVERSITÁRIO DA UNIVERSIDADE FEDERAL FLUMINENSE**

Monografia apresentada ao Curso de Bacharelado em Ciências da Computação do Instituto de Ciência e Tecnologia da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Bacharel.

Orientador: Prof. Dr. André Renato Villela da Silva

Rio das Ostras-RJ

2017

ANA FLÁVIA NEVES GALL

LEONARDO NEVES GALL

**SISTEMA DE ALOCAÇÃO DE HORÁRIOS EM UM CURSO
UNIVERSITÁRIO DA UNIVERSIDADE FEDERAL FLUMINENSE**

Monografia apresentada ao Curso de Bacharelado em Ciências da Computação do Instituto de Ciência e Tecnologia da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Bacharel.

BANCA EXAMINADORA

Prof. Dr. André Renato Villela da Silva – Orientador
Universidade Federal Fluminense

Prof. Dr. Carlos Bazilio Martins
Universidade Federal Fluminense

Prof. Dr. Dalessandro Soares Vianna
Universidade Federal Fluminense

Rio das Ostras-RJ

2017

Dedicamos este trabalho aos nossos pais,
Arlindo José Gall e Heliane Neves Gall.

Agradecimentos

Agradecemos primeiramente a Deus, pela força e coragem durante toda esta caminhada.

Aos nossos pais, **Arlindo José Gall** e **Heliane Neves Gall** pelo amor, incentivo e apoio incondicional.

À todos os professores por nos proporcionar não só o conhecimento racional, mas a demonstração da afetividade no processo de formação profissional, em especial ao nosso professor e orientador **André Renato Villela da Silva**, por todos os ensinamentos, paciência e disponibilidade conosco.

Aos amigos e familiares, que foram pessoas fundamentais em todo esse processo de apoio e aprendizado.

Á todos que de alguma forma nos apoiaram e contribuíram para que pudéssemos alcançar essa conquista.

Resumo

Este trabalho aborda o processo de alocação de horários em uma universidade, com o intuito de construir um sistema de geração automática de grade horária para o curso de Ciência da Computação da Universidade Federal Fluminense no campus de Rio das Ostras. Esse problema, conhecido como Timetabling, é recorrente a cada início de período letivo e, além disso, quando a alocação é feita de forma manual, os responsáveis pela construção da grade de horários consomem muito tempo para encontrar uma solução razoavelmente satisfatória, a qual, geralmente, não é a melhor solução possível.

Esse problema é de difícil solução, uma vez que o conjunto de interesses particulares da instituição, professores e alunos geram várias condições conflitantes a serem atendidas. Sendo assim, neste trabalho o problema é abordado através da criação de uma formulação matemática para atender o conjunto de restrições existentes, visando gerar uma melhor distribuição das aulas no quadro de horários.

Palavras-chave: Formulação matemática, Alocação de horários, Universidade.

Abstract

This paper deals with the time allocation process in a university, in order to build a timetable automatic generation system for the Computer Science course at the Universidade Federal Fluminense in Rio das Ostras' campus. This problem, known as Timetabling Problem, recurs every beginning of academic semester and, in addition, when the allocation is done manually, the responsible people for building the class timetable consume too much time to find a reasonably satisfactory solution, which, usually, it is not the best possible solution.

This problem is difficult to solve, since the set of particular interests of the institution, professors and students generate several conflicting conditions to be met. Thus, in this study the problem is addressed by creating a mathematical formulation to meet the restrictions set, aimed at generating a better distribution of classes in the timesheet.

Keywords: Mathematical Formulation, Timetabling, University.

LISTA DE FIGURAS

| | |
|--|----|
| FIGURA 1. DIAGRAMA DE CASO DE USO | 37 |
| FIGURA 2. DIAGRAMA DE CLASSES..... | 46 |
| FIGURA 3. MODELO ENTIDADE RELACIONAMENTO | 47 |
| FIGURA 4. TELA DE LOGIN | 49 |
| FIGURA 5. TELA INICIAL | 49 |
| FIGURA 6. TELA DE CADASTRO DE PROFESSOR (1) | 50 |
| FIGURA 7. TELA DE CADASTRO DE PROFESSOR (2) | 50 |
| FIGURA 8. TELA DE CONSULTA DE PROFESSORES | 51 |
| FIGURA 9. TELA DE CADASTRO DE TURMA..... | 51 |
| FIGURA 10. TELA DE CONSULTA DE TURMAS | 52 |
| FIGURA 11. TELA DE CADASTRO DE GRUPO (1)..... | 52 |
| FIGURA 12. TELA DE CADASTRO DE GRUPO (2)..... | 53 |
| FIGURA 13. TELA DE CONSULTA DE GRUPOS..... | 53 |
| FIGURA 14. TELA DE ALTERAÇÃO DE DEPARTAMENTO | 54 |
| FIGURA 15. TELA DE GERAÇÃO DE GRADE..... | 54 |
| FIGURA 16. EXEMPLO DO PDF GERADO COM A GRADE DE HORÁRIO..... | 55 |

LISTA DE TABELAS

| | |
|---|----|
| TABELA 1. EXEMPLO DE TURMA COM PERFIL DE DUAS AULAS DE 3 HORAS..... | 35 |
| TABELA 2. EXEMPLO DE TURMA COM PERFIL DE TRÊS AULAS DE 2 HORAS..... | 35 |
| TABELA 3. EXEMPLO DE TURMA COM PERFIL DE AULA DE 2 HORAS E DE 4 HORAS | 36 |
| TABELA 4. RESULTADO DO TESTE BASEADO NO NÚMERO DE SALAS | 57 |
| TABELA 5. RESULTADO DO TESTE DE EVITAR OU PROIBIR UM DIA DA SEMANA | 57 |
| TABELA 6. RESULTADO DO TESTE BASEADO NA PROIBIÇÃO DE HORÁRIOS | 58 |
| TABELA 7. RESULTADO DO TESTE BASEADO NA REMOÇÃO DO MÁXIMO DE AULAS POR DIA..... | 59 |
| TABELA 8. RESULTADO DO TESTE BASEADO NA FLEXIBILIZAÇÃO DO MODELO..... | 59 |

SUMÁRIO

| | |
|--|-------------|
| Agradecimentos | V |
| Resumo | VI |
| Abstract | VII |
| Lista de Figuras | VIII |
| Lista de Tabelas | IX |
| 1 – Introdução | 12 |
| 2 – Descrição do problema | 14 |
| 2.1 – Revisão de Literatura..... | 16 |
| 3 – Modelagem Matemática | 21 |
| 3.1 – Comparação com outra modelagem matemática..... | 21 |
| 3.2 – Formulação proposta | 22 |
| 3.3 – Exemplo relacionado ao perfil de turmas | 34 |
| 4 – Modelagem do sistema | 37 |
| 4.1 – Requisitos | 37 |
| 4.2 – Casos de uso | 37 |
| 4.2.1 – Efetuar Login (CSU01)..... | 38 |
| 4.2.2 – Alterar professor (CSU02) | 38 |
| 4.2.3 – Gerenciar turma (CSU03) | 39 |
| 4.2.4 – Gerenciar grupo (CSU04)..... | 40 |
| 4.2.5 – Gerenciar departamento (CSU05)..... | 41 |
| 4.2.6 – Gerenciar professor (CSU06)..... | 42 |
| 4.2.7 – Gerar grade de horários (CSU07) | 43 |
| 4.3 – Diagrama de classes..... | 44 |
| 4.4 – Banco de dados | 46 |
| 4.5 – Apresentação do sistema | 48 |
| 4.5.1 – Tela de autenticação | 49 |
| 4.5.2 – Tela principal | 49 |
| 4.5.3 – Telas relacionadas ao professor | 50 |

| | |
|--|-----------|
| 4.5.4 – Telas relacionadas à turma..... | 51 |
| 4.5.5 – Telas relacionadas ao grupo..... | 52 |
| 4.5.6 – Tela de departamento..... | 54 |
| 4.5.7 – Tela de geração da grade | 54 |
| 4.5.8 – Quadro de horários gerado | 55 |
| 5 – Resultados computacionais..... | 56 |
| 5.1 – Teste baseado no número de salas | 56 |
| 5.2 – Teste baseado em evitar e proibir dias da semana..... | 57 |
| 5.3 – Teste baseado na proibição de horários específicos | 58 |
| 5.4 – Teste baseado na remoção do máximo de aulas por dia..... | 58 |
| 5.5 – Teste baseado na flexibilização do modelo | 59 |
| 6 – Conclusão | 61 |
| 6.1 – Trabalhos futuros..... | 61 |
| Referências Bibliográficas | 63 |

1 – Introdução

O problema da geração de grade de horários em qualquer instituição de ensino é antigo e recorrente a cada início de período letivo. Os responsáveis pela construção da grade consomem horas do seu tempo para encontrar uma solução razoavelmente satisfatória, a qual, na maioria das vezes, não é a solução ótima. Trata-se de um problema combinatório complexo [1], uma vez que o conjunto de interesses particulares da instituição, professores e alunos geram várias condições conflitantes a serem atendidas. Devido a isso, com a realização do serviço de forma manual, é muito difícil encontrar a alocação de horários ideal. Em outras palavras, é árduo construir uma grade de horários em que cada um dos envolvidos fique o mais satisfeito possível.

O processo de alocação de horários é denominado Timetabling. Timetabling pode ser descrito como o problema de se alocar o encontro de determinados recursos dentro de um determinado espaço de tempo [2]. Conforme [1] e [3], este problema, no caso de instituições de ensino, baseia-se em agendar uma sequência de encontros, como aulas ou outro tipo de atividade escolar, entre a instituição, os professores e os alunos em um período de tempo determinado, de forma a satisfazer restrições de diversos tipos.

Apesar de este ser um tema muito estudado, existem diversas diferenças entre os muitos estudos de caso realizados. As principais divergências devem-se à complexidade adicional que é imposta pelas estruturas de cada curso/universidade, gerando uma variedade de restrições [4]. Além disso, o problema de alocação de horário pode envolver múltiplos subproblemas com características muito diferentes.

Este trabalho descreve um sistema de geração automática de grade de horários, desenvolvido para atender as necessidades do curso de Ciência da Computação da Universidade Federal Fluminense (UFF), no Centro Universitário de Rio das Ostras. O problema é abordado através da criação de uma formulação matemática para atender o conjunto de restrições existentes e essa formulação é processada pelo otimizador Cplex [5]. Além disso, foi desenvolvido um sistema destinado à coleta dos diversos dados envolvidos no problema.

A situação para esta universidade complica-se também pelo fato de que um número considerável de professores reside em outras cidades, o que insere novas restrições, como o desejo por minimizar a quantidade de dias que se deve ir à instituição.

Recentemente, o curso passou a utilizar um sistema para geração automatizada da grade de horários [6], onde a solução é obtida de forma heurística a partir de sugestões disponibilizadas pelas partes envolvidas. Este sistema será descrito de forma mais detalhada no Capítulo 2.

Este trabalho está dividido como se segue: no Capítulo 2 é apresentada a descrição do problema, assim como a revisão da literatura; o Capítulo 3 aborda a metodologia utilizada para resolver o problema de alocação de horários da universidade, ou seja, descreve a modelagem matemática a partir das restrições consideradas; o Capítulo 4 apresenta a modelagem do sistema, desde o levantamento dos requisitos até a apresentação das telas. No Capítulo 5 são apresentados os resultados computacionais obtidos através dos testes; por último, o Capítulo 6 apresenta as considerações finais e propostas de trabalhos futuros.

2 – Descrição do problema

O objetivo deste trabalho é disponibilizar um modelo que possibilite a construção de grades horárias de forma mais eficiente, de acordo com as restrições impostas pela universidade em questão. A necessidade da universidade é atender ao máximo possível a preferência dos professores e dos grupos de turmas de ministrar ou possuir aulas em dias da semana e horários especificados.

É importante mencionar alguns conceitos básicos e específicos em relação ao problema. As aulas são ministradas de segunda à sexta-feira com a disponibilidade de horário de 7 às 22 horas, sendo considerados normalmente 7 horários de 2 horas cada. As turmas são divididas em grupos, que podem ser os próprios períodos das mesmas ou algum outro (grupo) conjunto, como um grupo de optativas, por exemplo. Entretanto, uma turma não precisa estar obrigatoriamente associada a um grupo. Um grupo, por sua vez, consiste em um conjunto de turmas que não podem ter conflito de horário entre si, ou seja, que não podem estar alocadas no mesmo dia e horário.

A definição de qual professor será responsável por cada turma é realizada previamente. Porém, os professores possuem outras atividades acadêmicas além das aulas, como projetos de extensão, iniciação científica, mestrado, entre outros. Sendo assim, os mesmos não possuem total disponibilidade de horário para ministrar suas aulas, havendo restrições para essa alocação.

No problema abordado também é considerada a preferência de como cada professor gostaria que as aulas de suas turmas fossem alocadas, criando seis possíveis perfis para as mesmas através de uma variável de controle, que pode variar de 0 a 5. As turmas possuem cargas horárias de 2, 4 ou 6 horas semanais. Sendo assim, turmas que tenham carga horária de 6 horas semanais podem ter três perfis diferentes: duas aulas por semana de 3 horas cada (*controle = 1*), uma aula de 4 horas e outra aula de 2 horas por semana (*controle = 2*) ou três aulas por semana de 2 horas cada (*controle = 3*). Turmas que tenham carga horária de 2 horas semanais estarão aptas apenas para um perfil: uma aula por semana de 2 horas (*controle = 4*). Por fim, turmas que tenham carga horária de 4 horas semanais podem ter dois perfis distintos: duas aulas por semana de 2 horas cada (*controle = 0*) ou uma aula por semana de 4 horas (*controle = 5*).

Para a criação dos perfis de cada turma é necessário criar uma variável que realiza o controle dos mesmos. Além disso, para permitir a alocação de aulas de 3 horas consecutivas, é necessária a criação de variáveis que indiquem que uma turma possui aula apenas na metade de um horário, uma vez que os horários são definidos com tamanho de 2 horas cada. Dessa maneira, uma aula de 3 horas é composta por um horário inteiro e a metade de um horário, podendo ser a primeira metade do horário seguinte ou a segunda metade do horário anterior.

A solução do problema abrange também a questão de horários impedidos para alguns grupos de turmas, assim como horários proibidos para qualquer turma do departamento. É possível justificar a ocorrência de alguns horários não estarem disponíveis para um determinado grupo de turma através de um exemplo envolvendo turmas dos últimos períodos do curso de Computação. Nessa fase da graduação, os alunos normalmente estão interessados em estagiar em alguma empresa, necessitando de maior flexibilidade nos horários da faculdade. À vista disso, é interessante proibir que as turmas desses grupos tenham aulas durante a parte da manhã e tarde, forçando a alocação do horário para a noite e permitindo que os alunos conciliem as duas atividades.

O fundamento para o caso de horários proibidos para qualquer turma do departamento, por sua vez, deve-se à necessidade de reservar tempo para um evento específico do departamento, como horário de reunião. Dessa maneira, nenhum professor do departamento de Computação estará disponível para ministrar aulas nesse horário.

Ademais, existem grupos de turmas entre os quais não é desejada a existência de conflito de horário. Isto é compreensível visto que, frequentemente, os alunos não acompanham a grade curricular corretamente e precisam cursar matérias de períodos consecutivos. Por exemplo, muitos alunos obtêm reprovações em disciplinas do primeiro período e precisam cursar disciplinas do segundo e primeiro períodos simultaneamente. Logo, seria interessante evitar conflitos entre esses grupos.

Existe também a limitação de um número de salas disponíveis, bem como o fato de que as turmas de outros departamentos já possuem o seu horário previamente definido, de forma que essas alocações não são de responsabilidade do modelo. Em contrapartida, os horários estabelecidos devem ser respeitados pelo mesmo. Com esse propósito, foi criado um professor p' fictício que representa todos os professores de outros departamentos, onde as penalidades do mesmo não são contabilizadas e as restrições específicas para o departamento de Computação não são aplicadas para as turmas pertencentes a ele. Dessa maneira, encontrar uma solução que ampare todas as restrições do problema se torna uma tarefa cada vez mais complexa.

2.1 – Revisão de Literatura

Diversos estudos exploram a questão de como realizar alocações de horários em instituições de ensino. Alguns deles se baseiam na criação de uma formulação matemática que atenda as restrições existentes.

O estudo relatado por [7] apresenta uma abordagem de programação inteira para o problema de alocação de horários de curso universitário, em que palestras semanais têm que ser programadas em um período e atribuídas às salas. Foram relatadas as restrições *hard* e *soft* (restrições que implicam na viabilidade e na qualidade da solução, respectivamente) desejadas e a formulação matemática foi criada para atender as mesmas. As principais restrições *hard* são para que não haja duas aulas em uma mesma sala no mesmo intervalo de tempo, e também para que cursos de um mesmo currículo atribuídos a um mesmo horário. Além disso, um professor não pode estar dando duas aulas ou mais simultaneamente. O problema é dividido em duas etapas ao invés de resolver diretamente uma formulação para atribuição de curso, horário e sala. Na primeira etapa, combinam-se os cursos com os horários, para que então, num segundo passo, estes pares possam ser atribuídos a alguma sala. A palestra deve ser alocada em uma sala que forneça os recursos necessários para a mesma, como projetor, computador, quadro, etc. Além disso, deve haver lugares suficientes para acolher esta palestra.

Existem também quatro restrições *soft* definidas nesse estudo, que são capacidade da sala, mínimo de dias trabalhados, compactação de currículo e estabilidade de sala, que consiste em incentivar que todas as palestras de um curso ocorram em uma mesma sala. A capacidade da sala deve fornecer os lugares conforme o solicitado por cada curso atribuído. O mínimo de dias trabalhados tem o objetivo de definir um número mínimo de dias entre os quais as palestras de um mesmo curso devem ser distribuídas. A outra restrição, de compactação de currículo, diz que para cada currículo, os cursos devem ocorrer consecutivamente ao longo de um dia, de modo a evitar intervalos de tempo entre os mesmos. A solução utiliza também a preferência de professores, mas o estudo não detalha como a mesma é realizada.

Os autores relataram diferentes conjuntos de experiências, utilizando instâncias já abordadas em outros estudos. Em geral, os tempos de execução foram bastante curtos e, em comparação aos resultados obtidos nas pesquisas anteriores, esta abordagem apresentou bons resultados.

O estudo realizado por [8] também abordou o problema de Timetabling através de programação inteira. Neste caso, o problema consistia em atribuir cursos, com horários pré-fixados, aos estudantes do Departamento de Design Industrial da *Eindhoven University of Technology*, na Holanda. Para fazer essa atribuição, cada aluno deve fazer uma lista de 10 cursos de sua preferência e o departamento faz a inscrição a partir disto. Inicialmente, esse processo era feito manualmente, de modo que, com o crescimento do número de alunos, muitos destes passaram a ficar insatisfeitos com as inscrições nos cursos. O problema pode ser modelado numa primeira formulação como um problema de fluxo de rede, sendo resolvido em tempo polinomial. Essa modelagem não era completa e, então foi desenvolvida uma formulação matemática completa para o problema com restrições adicionais, tornando-o mais complexo.

Na primeira formulação, as únicas restrições são de que um aluno não pode fazer dois cursos ao mesmo tempo e há um número máximo de alunos que podem ser atribuídos ao curso. Esse número máximo depende da preferência do professor e da capacidade da sala do curso. Cada curso tem um horário semanal já fixado que consiste em duas horas consecutivas. O que se notou foi que quando houvesse um número de alunos abaixo do limite inferior do mesmo, o curso poderia ser cancelado, exceto se este fosse obrigatório para alguns alunos. Além disso, esta formulação considerava que todos os cursos possuíam a mesma carga horária, o que não condiz com a realidade. Devido a isso, o problema não pode ser resolvido com a primeira formulação, necessitando de novas restrições.

O ano acadêmico é dividido em um determinado número de períodos, que são chamados de blocos e um curso pode ter várias seções com professores diferentes dando o mesmo conteúdo. Na segunda formulação, as restrições *hard* consistem em manter o número de alunos em uma seção abaixo do tamanho máximo definido, fazer com que a carga de horário atribuída aos estudantes seja menor ou igual a que foi requerida, seções atribuídas a um determinado aluno não entrem em conflito de tempo, os alunos só são atribuídos a uma seção se esta estiver na sua lista de preferências e também só são atribuídos a uma única seção de um mesmo curso. Além destas restrições *hard*, duas restrições *soft* são incluídas, que consistem em todos os cursos considerados urgentes serem disponibilizados e todos os limites inferiores de tamanho das seções serem respeitados.

O tempo de execução para resolver os casos foi insignificante e a administração e os alunos, em sua maioria, ficaram muito satisfeitos com os horários gerados pelo programa. Formulações matemáticas para problemas semelhantes também foram utilizados em [9].

Enquanto alguns estudos exploram a resolução do problema através de formulação matemática, outros focam na solução do mesmo por meio do uso de heurísticas. O estudo relatado por [6] consiste na geração do quadro de horários da Universidade Federal Fluminense através de técnicas heurísticas. Em contrapartida, o resultado obtido através de heurísticas ainda pode causar insatisfação entre os envolvidos. A solução encontrada para esse problema foi tornar o processo mais colaborativo e democrático, de forma que os alunos pudessem interagir e opinar no quadro de horários que irão cursar. Para a geração foram desenvolvidos métodos heurísticos, cujas variáveis possuem restrições com importâncias diversas. Neste contexto, o Auxílio Multicritério à Decisão (AMD) se apresenta como uma alternativa para a definição da importância de tais restrições. Além disso, foi utilizado o Método de Análise Hierárquica (AHP – *Analytic Hierarchy Process*) para definição dos pesos das restrições *soft*.

O sistema, que possui o nome de Colaborário, foi dividido em dois módulos. Um desenvolvido com o Framework FINNESS, que é um framework para a construção de meta-heurísticas baseado em Busca Local e que é responsável pela geração do quadro de horários; e o outro é um módulo Web que utiliza banco de dados para armazenamento das informações necessárias. Em relação à primeira etapa, o coordenador é responsável por efetuar o cadastro dos dados necessários e os professores e alunos podem editar seus dados, assim como os alunos podem escolher as disciplinas que pretendem cursar. Os dados adquiridos nessa etapa servirão de entrada para a heurística, que gerará um quadro inicial eficaz, onde serão respeitadas todas as restrições *hard* e o máximo de restrições *soft* possíveis. Na segunda etapa, com o quadro de horário gerado na primeira, os alunos e professores poderão sugerir mudanças para as suas turmas. Essas mudanças visam agradar o maior número possível de indivíduos. Dessa forma, o cálculo do peso da sugestão levará em consideração a quantidade de votos a favor ou contra a mesma.

As meta-heurísticas implementadas para a geração do quadro de horários foram GRASP, ILS e VNS e foram desenvolvidos um método Construtivo e um método de Busca Local que analisa diferentes estruturas de vizinhança. O primeiro serve para construir uma solução inicial e o segundo para refinar essa solução construída. Os resultados obtidos foram satisfatórios, uma vez que foi possível elaborar e comunicar os horários com maior qualidade, atendendo às exigências necessárias, bem como às preferências estabelecidas pelos professores e alunos. Outras abordagens heurísticas podem ser vistas em [10, 11].

Existem ainda estudos que analisam a solução do problema fazendo o uso de meta-heurísticas. O estudo realizado por [12] aborda o problema de alocação de horários em uma

universidade, que consiste em agendar as aulas de um conjunto de cursos em um horário semanal, de modo que cada aula seja atribuída a uma sala e a um horário respeitando o conjunto de restrições. O problema é tratado através de uma heurística híbrida, utilizando um algoritmo de Busca Tabu [13] Adaptativa para resolvê-lo. A solução usada possui quatro restrições *hard*, que consistem em manter cada aula de um curso agendada em um período distinto e a uma sala, garantir que duas aulas não sejam atribuídas a uma mesma sala num mesmo período, fazer com que aulas de cursos no mesmo currículo ou ensinadas pelo mesmo professor não sejam agendadas no mesmo período e, não atribuir aulas do curso a um período se o professor do curso não está disponível neste mesmo horário. Além dessas restrições *hard*, o problema define outras quatro restrições *soft*, que são equivalentes as abordadas no estudo [7].

Esse algoritmo se divide em três fases: inicialização, intensificação e diversificação. A primeira consiste em uma heurística gulosa para produzir rapidamente grades de horário iniciais viáveis. Nessa primeira fase, é inserida uma aula apropriada do curso no calendário, inicialmente vazio, de cada vez. Duas operações são realizadas a cada etapa, sendo a primeira selecionar uma aula de um curso que ainda não tenha sido atribuída ao calendário e, a segunda é determinar um par período-sala para esta aula. Nessa heurística, um curso com poucos períodos disponíveis e um grande número de aulas tem a prioridade. A segunda fase consiste em aplicar a Busca Tabu a partir da solução inicial, tentando encontrar uma solução melhor na vizinhança. A terceira consiste em aplicar a Busca Local Iterativa para perturbar a solução atual quando a Busca Tabu atinge a solução ideal local. Através dos testes realizados, os autores constaram que o algoritmo proposto é vantajoso em relação à Busca Tabu ou à Busca Local Iterativa sozinhos.

O estudo relatado por [14] fornece um tratamento matemático para o problema de definição de horários em universidades e aborda tal problema através de um algoritmo de duas fases baseado em meta-heurística. A qualidade de uma solução válida foi medida por um raio de viabilidade (DTF - *Distance To Feasibility*), que consiste na soma dos tamanhos de todos os eventos que não estão contidos na solução. O primeiro passo necessário é produzir uma solução válida que minimiza essa medida DTF e, para isso, é utilizada uma adaptação do algoritmo PARTIALCOL, que foi originalmente projetado para o problema de coloração de grafos. Uma solução inicial é construída alocando eventos um a um em intervalos de tempo de tal modo que nenhuma das restrições *hard* sejam violadas. Nessa primeira fase do algoritmo não são consideradas restrições *soft*. Na segunda fase do algoritmo, o resfriamento

simulado (SA - *simulated annealing*) é usado para explorar o espaço de soluções válidas / viáveis, na tentativa de minimizar o número de violações de restrições *soft*.

As restrições *hard* consistem, primeiramente, em ter um conjunto de alunos matriculados para cada evento, de modo que os eventos devem ser atribuídos a intervalos de tempo onde nenhum aluno é obrigado a participar de mais de um evento em um intervalo de tempo qualquer. Também diz que cada evento necessita de um conjunto de características da sala (capacidade, projetor, etc.) e que o mesmo só deve ser atribuído à sala que se adeque às características. É definido que alguns eventos não podem ser ensinados em determinados intervalos de tempo, além de definir também as restrições de precedência, que dizem que alguns eventos precisam ser agendados antes ou depois dos outros. Existem três restrições *soft* na solução, que consistem em alunos não participarem de eventos no último intervalo de tempo de cada dia, de eventos em três ou mais intervalos de tempo seguidos no mesmo dia e de apenas um evento no dia.

Através de um conjunto bem conhecido de instâncias de problemas de referência, o algoritmo proposto pelos autores foi testado e produziu excelentes resultados. Outras abordagens utilizando meta-heurísticas podem ser vistas em [15, 16].

Este Trabalho de Conclusão de Curso é uma continuação de um projeto de Iniciação Científica, que inclusive gerou um artigo aceito no Simpósio Brasileiro de Pesquisa Operacional 2016 [17], contendo alguns dos principais aspectos e resultados descritos posteriormente.

3 – Modelagem Matemática

A abordagem escolhida para o problema é através de uma modelagem matemática, classificada como Programação Inteira Mista. Essa escolha deve-se ao fato de existir um grande número de restrições e situações que devem ser contempladas, o que torna o problema de difícil solução e quase impossível de ser tratado manualmente. Dessa forma, essa abordagem tem o intuito de gerar uma melhor distribuição de turmas e professores no quadro de horários.

Neste trabalho será utilizado o *Cplex* da IBM ILOG, que é um software capaz de trabalhar com programação matemática. Uma das maiores funcionalidades do *Cplex* é a otimização, ou seja, a obtenção de uma solução ótima (a melhor possível) para o problema. Os otimizadores, tais como o *Cplex*, têm apresentado excelente desempenho e, ademais, os mesmos garantem a qualidade da solução final.

3.1 – Comparação com outra modelagem matemática

Inicialmente, para que fosse possível atender a restrição de que cada turma tenha seu próprio perfil, ou seja, de que pudesse ser determinado como as aulas de cada turma serão distribuídas ao longo da semana, foi elaborada uma formulação em que eram considerados, por dia, 14 horários de 1 hora cada, ao invés de apenas 7 horários de 2 horas. Desta forma, seria possível alocar a quantidade de horas desejadas para cada dia. Por exemplo, para uma turma com carga horária de 6 horas semanais e com perfil de 2 aulas por semana de 3 horas cada, seriam alocados 3 horários em 2 dias da semana. Consequentemente, novas restrições tornaram-se necessárias para manter esse controle, de forma a evitar que as aulas de uma hora ficassem dispersas ao longo do dia. Além do grande número de restrições, o número de variáveis também aumentou consideravelmente com este modelo.

A formulação inicial funcionava bem para uma amostra muito pequena de turmas. Em contrapartida, a partir de 20 turmas o algoritmo executava em média por 48 horas e consumia uma quantidade muito grande de memória, a ponto de não conseguir concluir a sua execução.

Sendo assim, foi elaborada a formulação nova (atual), em que foram considerados 7 horários de 2 horas cada e que a metade desses horários também pudessem ser alocadas (através de variáveis binárias específicas), sendo possível compor aulas de 3 horas, como será descrito na próxima seção.

3.2 – Formulação proposta

O modelo proposto apresenta os seguintes parâmetros de entrada:

- τ - Conjunto de turmas;
- G - Conjunto de grupos de turmas;
- P - Conjunto de professores, onde p' representa todos os professores de outros departamentos;
- NUM_SALAS - Número de salas;
- $controle^i$ – Número de 0 a 5 que indica o perfil da turma i ;
- $PH_{D,H}^g$ – Penalidade em alocar alguma turma do grupo g no dia D , horário H ;
- $PHP_{D,H}^p$ – Penalidade em alocar o professor p no dia D , horário H ;
- CH_i – Carga horária semanal da turma i ;
- MDS^p – Número máximo de dias por semana nos quais o professor p deseja ter suas aulas distribuídas;
- MAD^p – Número máximo de aulas por dia que o professor p deseja ministrar;
- MA^g – Número máximo de aulas por dia para turmas do grupo g ;
- $PB^{p,n}$ – Penalidade do professor p de ter um intervalo de tempo vago de tamanho n em um dia;
- $PBG^{g,n}$ – Penalidade do grupo g de ter um intervalo de tempo vago de tamanho n em um dia;
- $PBS^{p,n}$ – Penalidade do professor p de ter um intervalo de dias vagos de tamanho n na semana;
- $PEAD^p$ – Penalidade do professor p por excesso de aulas em um dia;
- PEA^g – Penalidade do grupo g por excesso de aulas em um dia;
- $PEDS^p$ – Penalidade do professor p por excesso de dias na semana;
- $PCONFLITO^g$ – Penalidade do grupo g por existir conflito com outros grupos;
- $PERIODO(g)$ – Conjunto de turmas que pertencem ao grupo g ;

- $TURMAS(p)$ - Conjunto de turmas associadas ao professor p ;
- $PROIBIDOS$ - Conjunto de dias e horários em que todos os professores do departamento de Computação não podem ministrar aulas, como por exemplo, um horário fixo para reunião;
- $CENSURADOS(g)$ - Conjunto de dias e horários em que o grupo g não pode ter aula, como por exemplo, horários destinados para a realização de estágio;
- $OBRIGATORIO(i)$ - Conjunto de dias e horários em que a turma i já está alocada. Por exemplo, as turmas de outros departamentos já possuem o seu horário previamente definido;
- $CONFIG$ - Pares de grupos que possuem conflito entre si.

Além disso, tem-se o seguinte conjunto de variáveis:

- $X_{D,H}^i$ (binária): 1 se a turma i tem aula no dia D , horário H , 0 caso contrário;
- $Y_{D,H}^p$ (binária): 1 se o professor p tem aula no dia D , horário H , 0 caso contrário;
- $Z_{D,H}^g$ (binária): 1 se o período g tem aula no dia D , horário H , 0 caso contrário;
- W_D^p (binária): 1 se o professor p tem aula no dia D , 0 caso contrário;
- $B_{D,H}^{p,n}$ (binária): 1 se existe um intervalo de tempo vago de tamanho n no horário do professor p , no dia D , começando pelo horário H , 0 caso contrário;
- $F_D^{p,n}$ (binária): 1 se existe um intervalo de dias vagos de tamanho n na agenda do professor p , começando pelo dia D , 0 caso contrário;
- $L_{D,H}^{g,n}$ (binária): 1 se existe um intervalo de tempo vago de tamanho n no horário do grupo g , no dia D , começando pelo horário H , 0 caso contrário;
- $R_{D,H}^{g,t}$ (binária): 1 se há conflito entre turmas dos grupos g e t no dia D , horário H , 0 caso contrário;
- $U_{D,H}^{i,j}$ (binária): 1 se a turma i possui aula na metade j (primeira metade ou segunda metade) do horário H , no dia D , 0 caso contrário;
- $N_{D,H}^{p,j}$ (binária): 1 se o professor p possui aula na metade j (primeira metade ou segunda metade) do horário H , no dia D , 0 caso contrário;

- $NG_{D,H}^{g,j}$ (binária): 1 se o grupo g possui aula na metade j (primeira metade ou segunda metade) do horário H , no dia D , 0 caso contrário;
- $V_D^{i,k}$ (binária): 1 se existe aula de k horas consecutivas no dia D para a turma i , 0 caso contrário;
- EDS^p (inteira): Excesso de dias por semana do professor p ;
- EAD_D^p (inteira): Excesso de aulas do professor p no dia D ;
- EA_D^g (inteira): Excesso de aulas do grupo g no dia D ;
- $P0$ (inteira): Penalidades em alocar os professores em dias e horários indesejáveis;
- $P1$ (inteira): Penalidades em alocar os grupos em dias e horários indesejáveis;
- $P2^p$ (inteira): Penalidades de intervalos de tempo vago nos horários do professor p ;
- $P3^g$ (inteira): Penalidades de intervalos de tempo vago nos horários do grupo g ;
- $P4^p$ (inteira): Penalidades de intervalo de dias vagos nos dias da semana do professor p ;
- $P5^p$ (inteira): Penalidades por excesso de aulas em um dia do professor p ;
- $P6^g$ (inteira): Penalidades por excesso de aulas em um dia do grupo g ;
- $P7$ (inteira): Penalidades por excesso de dias na semana dos professores;
- $P8$ (inteira): Penalidades por conflito de horários entre grupos conflitantes.

As restrições foram identificadas a partir das necessidades que o curso apresenta e das preferências dos professores e alunos, visando gerar uma grade horária mais satisfatória. Uma solução é viável se todas as aulas forem alocadas de forma que nenhuma das restrições fortes seja violada. Essas restrições, conhecidas como restrições *hard* (R1-R13), são apresentadas a seguir juntamente com as respectivas formulações.

R1 - Aulas de turmas do mesmo grupo não podem estar alocadas no mesmo dia e horário. Um grupo consiste em um conjunto de turmas onde é desejado a não existência de conflitos de horário entre as mesmas. Como exemplo, as turmas de cada período constituem um grupo, uma vez que um aluno deve poder cursar todas as disciplinas de um período no mesmo semestre.

$$Z_{D,H}^g = \sum_{i \in \text{PERÍODO}(g)} X_{D,H}^i \quad \forall D = 1 \dots 5, \forall H = 1 \dots 7, \forall g \in G \quad (1)$$

$$NG_{D,H}^{g,j} = \sum_{i \in \text{PERÍODO}(g)} U_{D,H}^{i,j} \quad \forall D = 1 \dots 5, \forall H = 1 \dots 7, \forall g \in G, \forall j = 1 \dots 2 \quad (2)$$

R2 - Aulas de turmas ministradas pelo mesmo professor não podem estar alocadas no mesmo dia e horário.

$$Y_{D,H}^p = \sum_{i \in \text{TURMAS}(p)} X_{D,H}^i \quad \forall D = 1 \dots 5, \forall H = 1 \dots 7, \forall p \in P \quad (3)$$

$$N_{D,H}^{p,j} = \sum_{i \in \text{TURMAS}(p)} U_{D,H}^{i,j} \quad \forall D = 1 \dots 5, \forall H = 1 \dots 7, \forall p \in P, \forall j = 1 \dots 2 \quad (4)$$

R3 – Uma turma não pode estar alocada em um horário que não esteja disponível. O conjunto PROIBIDOS é utilizado para reservar um dia e horário específico da semana para determinado evento. Dessa forma, nenhuma turma do Departamento de Computação poderá ser alocada para os horários que constam neste conjunto.

$$X_{D,H}^i = 0 \quad \forall (D,H) \in \text{PROIBIDOS}, \forall i \in \text{TURMAS}(p), p \neq p' \quad (5)$$

$$U_{D,H}^{i,j} = 0 \quad \forall (D,H,j) \in \text{PROIBIDOS}, \forall i \in \text{TURMAS}(p), p \neq p' \quad (6)$$

R4 – Todas as turmas devem ter seu número de aulas respeitado. Em outras palavras, é necessário que o horário de cada turma seja alocado de maneira que a mesma possua sua carga horária semanal satisfeita.

$$2 * \sum_{H=1}^7 \sum_{D=1}^5 X_{D,H}^i + \sum_{H=1}^7 \sum_{D=1}^5 \sum_{j=1}^2 U_{D,H}^{i,j} = CH_i \quad \forall i \in \tau \quad (7)$$

R5 - Todas as aulas devem ser acomodadas em um determinado número de salas. O Departamento de Computação possui um número específico de salas, de modo que não é possível existir, no mesmo momento, mais aulas do que esse número, visto que não existiriam acomodações suficientes para todas as turmas.

$$\sum_{i=0}^{T-1} X_{D,H}^i + \sum_{i=0}^{T-1} U_{D,H}^{i,j} \leq NUM_SALAS \quad \forall D = 1 \dots 5, \forall H = 1 \dots 7, \forall j = 1 \dots 2 \quad (8)$$

R6 – Todas as aulas de um grupo em dias e horários pertencentes ao conjunto $CENSURADOS(g)$ devem ser proibidas. Esse conjunto é utilizado para impedir horários que não são desejados para um determinado grupo de turmas.

$$Z_{D,H}^g = 0 \quad \forall g \in G, \forall (D, H) \in CENSURADOS(g) \quad (9)$$

$$NG_{D,H}^{g,j} = 0 \quad \forall g \in G, \forall j = 1..2, \forall (D, H) \in CENSURADOS(g) \quad (10)$$

R7 – Algumas turmas já possuem um horário fixo, definido previamente, sendo necessário atribuí-lo compulsoriamente.

$$X_{D,H}^i = 1 \quad \forall i \in \tau, \forall (D, H) \in OBRIGATORIO(i) \quad (11)$$

$$U_{D,H}^{i,j} = 1 \quad \forall i \in \tau, \forall (D, H, j) \in OBRIGATORIO(i) \quad (12)$$

R8 – Essa restrição realiza o controle dos perfis das turmas, de forma que cada turma tenha o seu perfil respeitado, contabilizando a quantidade de aulas com 2, 3 e 4 horas por semana. A restrição se aplica para todas as turmas ($\forall i \in \tau$).

$$\left. \begin{aligned} \sum_{D=1}^5 V_D^{i,2} &= 2 \\ \sum_{D=1}^5 V_D^{i,3} &= 0 \\ \sum_{D=1}^5 V_D^{i,4} &= 0 \end{aligned} \right\} \begin{array}{l} \text{se controle}^i = 0 \\ 2 \text{ aulas de 2 horas} \end{array} \quad (13)$$

$$\left. \begin{aligned} \sum_{D=1}^5 V_D^{i,2} &= 0 \\ \sum_{D=1}^5 V_D^{i,3} &= 2 \\ \sum_{D=1}^5 V_D^{i,4} &= 0 \end{aligned} \right\} \begin{array}{l} \text{se controle}^i = 1 \\ 2 \text{ aulas de 3 horas} \end{array} \quad (14)$$

$$\left. \begin{aligned} \sum_{D=1}^5 V_D^{i,2} &= 1 \\ \sum_{D=1}^5 V_D^{i,3} &= 0 \\ \sum_{D=1}^5 V_D^{i,4} &= 1 \end{aligned} \right\} \begin{array}{l} \text{se controle}^i = 2 \\ 1 \text{ aula de 2 horas e} \\ 1 \text{ aula de 4 horas} \end{array} \quad (15)$$

$$\left. \begin{aligned} \sum_{D=1}^5 V_D^{i,2} &= 3 \\ \sum_{D=1}^5 V_D^{i,3} &= 0 \\ \sum_{D=1}^5 V_D^{i,4} &= 0 \end{aligned} \right\} \begin{array}{l} \text{se controle}^i = 3 \\ 3 \text{ aulas de 2 horas} \end{array} \quad (16)$$

$$\left. \begin{array}{l} \sum_{D=1}^5 V_D^{i,2} = 1 \\ \sum_{D=1}^5 V_D^{i,3} = 0 \\ \sum_{D=1}^5 V_D^{i,4} = 0 \end{array} \right\} \begin{array}{l} \text{se controle}^i = 4 \\ 1 \text{ aula de 2 horas} \end{array} \quad (17)$$

$$\left. \begin{array}{l} \sum_{D=1}^5 V_D^{i,2} = 0 \\ \sum_{D=1}^5 V_D^{i,3} = 0 \\ \sum_{D=1}^5 V_D^{i,4} = 1 \end{array} \right\} \begin{array}{l} \text{se controle}^i = 5 \\ 1 \text{ aula de 4 horas} \end{array} \quad (18)$$

R9 – Essa restrição relaciona a variável que realiza o controle dos perfis ($V_D^{i,k}$) com a quantidade de aulas no dia. Ou seja, as Equações (19) verificam se uma turma possui aulas de 2, 3 ou 4 horas em um dia e as Equações (20) garantem que não existirá mais de um perfil no mesmo dia para uma turma. Vale dizer que o lado direito das Equações (19) poderia ter um valor maior do que o lado esquerdo, porém isso não acontece, pois as Equações (7) controlam a carga horária semanal da turma. Ademais, as Equações (20) não são necessárias para a solução se o sistema obrigar que cada turma tenha um perfil pré-definido, ou seja, um valor de controle^i de 0 a 5. Nesse caso, as Equações (19) em conjunto com as equações da restrição R8 já são suficientes para a geração da solução. Se esse perfil não fosse definido pelo usuário do sistema, as Equações (20) seriam necessárias para garantir que cada turma só possa estar associada a um único perfil. Porém, por questão de otimização, estas equações foram mantidas. Na seção 3.3, para um melhor entendimento, esta relação do perfil de turmas é exemplificada.

$$k * V_D^{i,k} \leq 2 * \sum_{H=1}^7 X_{D,H}^i + \sum_{H=1}^7 \sum_{j=1}^2 U_{D,H}^{i,j} \quad \forall i \in \tau, \forall D = 1 \dots 5, \forall k = 2..4 \quad (19)$$

$$\sum_{k=2}^4 V_D^{i,k} \leq 1 \quad \forall i \in \tau, \forall D = 1 \dots 5 \quad (20)$$

R10 – Essa restrição impede que a metade de um horário seja alocada se não existir um horário inteiro adjacente à mesma. Em outras palavras, se o horário inteiro anterior não estiver alocado, não será possível que a primeira metade do horário seguinte seja alocada. Assim como se o horário inteiro posterior não estiver alocado, não será possível que a segunda metade do horário anterior seja alocada. O mesmo vale para os horários de grupos e professores, como se segue. Além disso, a primeira metade do primeiro horário e a segunda metade do último horário nunca poderão estar alocadas (Equações 23 e 24), já que não haveria um horário inteiro adjacente às mesmas.

$$U_{D,1}^{i,1} = 0 \quad \forall i \in \tau, \forall D = 1 \dots 5 \quad (21)$$

$$U_{D,7}^{i,2} = 0 \quad \forall i \in \tau, \forall D = 1 \dots 5 \quad (22)$$

$$U_{D,H}^{i,1} \leq X_{D,H-1}^i \quad \forall i \in \tau, \forall D = 1 \dots 5, \forall H = 2 \dots 7 \quad (23)$$

$$U_{D,H}^{i,2} \leq X_{D,H+1}^i \quad \forall i \in \tau, \forall D = 1 \dots 5, \forall H = 1 \dots 6 \quad (24)$$

$$N_{D,H}^{p,1} \leq Y_{D,H-1}^p \quad \forall p \in P, \forall D = 1 \dots 5, \forall H = 2 \dots 7 \quad (25)$$

$$N_{D,H}^{p,2} \leq Y_{D,H+1}^p \quad \forall p \in P, \forall D = 1 \dots 5, \forall H = 1 \dots 6 \quad (26)$$

$$NG_{D,H}^{g,1} \leq Z_{D,H-1}^g \quad \forall g \in G, \forall D = 1 \dots 5, \forall H = 2 \dots 7 \quad (27)$$

$$NG_{D,H}^{g,2} \leq Z_{D,H+1}^g \quad \forall g \in G, \forall D = 1 \dots 5, \forall H = 1 \dots 6 \quad (28)$$

R11 – Essa restrição impede que a metade de um horário seja alocada se o horário inteiro já tiver sido alocado para uma turma, uma vez que não faria sentido ter mais de uma alocação no mesmo horário para uma mesma turma. O mesmo vale para professores e grupos.

$$X_{D,H}^i + U_{D,H}^{i,j} \leq 1 \quad \forall i \in \tau, \forall D = 1 \dots 5, \forall H = 1 \dots 7, \forall j = 1 \dots 2 \quad (29)$$

$$Y_{D,H}^p + N_{D,H}^{p,j} \leq 1 \quad \forall p \in P, \forall D = 1 \dots 5, \forall H = 1 \dots 7, \forall j = 1 \dots 2 \quad (30)$$

$$Z_{D,H}^g + NG_{D,H}^{g,2} \leq 1 \quad \forall g \in G, \forall D = 1 \dots 5, \forall H = 1 \dots 7, \forall j = 1 \dots 2 \quad (31)$$

R12 – Para permitir a alocação de aulas de 4 horas em um dia, é necessário impedir que existam intervalos de tempo vago entre dois horários inteiros ($X_{D,H}^i$) alocados em um mesmo dia, fazendo com que sejam 4 horas seguidas e não duas aulas de 2 horas separadas.

$$X_{D,H}^i - \sum_{k=1}^n X_{D,H+k}^i + X_{D,H+n+1}^i \leq 2 - V_D^{i,4} \quad (32)$$

$$\forall i \in \tau, \forall D = 1 \dots 5, \forall n = 1 \dots 5, \forall H = 1 \dots (6 - n)$$

R13 – Essa restrição impede que aulas de 3 horas sejam alocadas de forma dividida entre antes e depois do horário de almoço.

$$U_{D,3}^{i,2} + X_{D,4}^i \leq 1 \quad \forall i \in \tau, \forall D = 1 \dots 5 \quad (33)$$

$$U_{D,4}^{i,1} + X_{D,3}^i \leq 1 \quad \forall i \in \tau, \forall D = 1 \dots 5 \quad (34)$$

A qualidade de uma solução viável, por sua vez, depende da satisfação das restrições fracas, conhecidas como restrições *soft* (R14-R22), que são apresentadas a seguir. Se uma restrição fraca for violada, uma penalidade é somada ao valor da solução.

R14 - Penalizar aulas de um professor alocadas em dias e horários indesejáveis, através do parâmetro de entrada fornecido ($PHP_{D,H}^p$). Esse parâmetro indica um valor de penalidade de 0 a 10 para cada par (dia, horário). Desse modo, a solução encontrada para a grade de horários será mais satisfatória para os professores, uma vez que o objetivo é minimizar as penalidades em geral.

$$\sum_{H=1}^7 \sum_{D=1}^5 \left[(Y_{D,H}^p * PPHP_{D,H}^p) + (N_{D,H}^{p,1} + N_{D,H}^{p,2}) * \frac{PHPP_{D,H}^p}{2} \right] = P0 \quad \forall p \in P \quad (35)$$

R15 - Penalizar aulas de um período alocadas em dias e horários indesejáveis, através do parâmetro de entrada fornecido ($PH_{D,H}^g$). Esse parâmetro de entrada e essa restrição ocorrem e se justificam pela mesma forma e razão da restrição R14.

$$\sum_{H=1}^7 \sum_{D=1}^5 \left[(Z_{D,H}^g * PH_{D,H}^g) + (NG_{D,H}^{g,1} + NG_{D,H}^{g,2}) * \frac{PH_{D,H}^g}{2} \right] = P1 \quad \forall g \in G \quad (36)$$

R16 – Penalizar os intervalos de tempo vagos entre horários de aulas de um professor em um dia, através do parâmetro de entrada $PB^{p,n}$. Esse parâmetro de entrada indica um valor de penalidade de 0 a 10 para cada tamanho n possível de tempo vago. Dessa maneira, é possível evitar que existam intervalos de tempo vago no dia do professor. Essa restrição se justifica devido ao fato de que alguns professores preferem ministrar aulas consecutivas, ao invés de ter horários vagos entre suas aulas.

$$Y_{D,H}^p - \sum_{i=1}^n Y_{D,H+i}^p + Y_{D,H+n+1}^p - B_{D,H}^{p,n} \leq 1 \quad (37)$$

$$\forall p \in P, \forall D = 1 \dots 5, \forall n = 1 \dots 5, \forall H = 1 \dots (6 - n)$$

$$P2^p \sum_{D=1}^5 \sum_{n=1}^5 \sum_{H=1}^{6-n} \left[(B_{D,H}^{p,n} * PB^{p,n}) - PB^{p,n} * \frac{N_{D,H}^{p,1} + N_{D,H}^{p,2}}{2 * n} \right] \quad \forall p \in P \quad (38)$$

R17 – Penalizar os intervalos de tempo vagos entre aulas de disciplinas do mesmo período em um dia, através do parâmetro de entrada $PBG^{g,n}$. Sendo assim, é possível evitar que existam intervalos de tempo vago no dia do grupo, onde n é o tamanho do intervalo. Essa restrição e esse parâmetro de entrada ocorrem e se justificam pela mesma razão que a restrição R16.

$$Z_{D,H}^g - \sum_{i=1}^n Z_{D,H+i}^g + Z_{D,H+n+1}^g - L_{D,H}^{g,n} \leq 1 \quad (39)$$

$$\forall g \in G, \forall D = 1 \dots 5, \forall n = 1 \dots 5, \forall H = 1 \dots (6 - n)$$

$$P3^g = \sum_{D=1}^5 \sum_{n=1}^{12} \sum_{H=1}^{13-n} \left[(L_{D,H}^{g,n} * PBG^{g,n}) - PBG^{g,n} * \frac{NG_{D,H}^{g,1} + NG_{D,H}^{g,2}}{2 * n} \right] \quad \forall g \in G \quad (40)$$

R18 – Penalizar os intervalos de dias vagos entre dias de aulas de um professor, baseado no parâmetro de entrada $PBS^{p,n}$, que indica um valor de penalidade de 0 a 10 para cada tamanho n do intervalo de dias vagos. Devido ao fato de que alguns professores não residem em Rio das Ostras, é interessante que os mesmos possam dar aulas em dias consecutivos, sem que existam dias vagos entre os dias de trabalho. Por exemplo, se um professor que não mora na cidade estiver com aulas alocadas no início da semana e outras só na sexta-feira, ele terá maiores gastos com transporte ou hospedagem.

$$W_D^p - \sum_{i=1}^n W_{D+i}^p + W_{D+n+1}^p - F_D^{p,n} \leq 1 \quad \forall p \in P, \forall n = 1 \dots 3, \forall D = 1 \dots (4 - n) \quad (41)$$

$$P4^p = \sum_{n=1}^3 \sum_{D=1}^{4-n} (F_D^{p,n} * PBS^{p,n}) \quad \forall p \in P \quad (42)$$

R19 - Penalizar o excesso de aulas de um professor em um dia, baseado no número máximo de aulas que o mesmo deseja ministrar por dia (MAD^p). Essa restrição é justificável, uma vez que muitas aulas em um mesmo dia pode se tornar exaustivo, reduzindo, inclusive, o rendimento das aulas.

$$2 * \sum_{H=1}^7 Y_{D,H}^p + \sum_{H=1}^7 \sum_{j=1}^2 N_{D,H}^{p,j} - MAD^p \leq EAD_D^p \quad \forall p \in P, \forall D = 1 \dots 5 \quad (43)$$

$$EAD_D^p \geq 0 \quad \forall p \in P, \forall D = 1 \dots 5 \quad (44)$$

$$P5^p = \sum_{D=1}^5 (EAD_D^p * PEAD^p) \quad \forall p \in P \quad (45)$$

R20 - Penalizar o excesso de aulas de turmas de um grupo em um dia, baseado no número máximo de aulas desejado por dia do grupo (MA^g). Essa restrição é plausível pelo mesmo motivo da restrição R19.

$$2 * \sum_{H=1}^7 Z_{D,H}^g + \sum_{H=1}^7 \sum_{j=1}^2 NG_{D,H}^{g,j} - MA^g \leq EA_D^g \quad \forall g \in G, \forall D = 1 \dots 5 \quad (46)$$

$$EA_D^g \geq 0 \quad \forall g \in G, \forall D = 1 \dots 5 \quad (47)$$

$$P6^g = \sum_{D=1}^5 (EA_D^g * PEA^g) \quad \forall g \in G \quad (48)$$

R21 - Penalizar o excesso de dias nos quais um professor tem suas aulas distribuídas, baseado no número máximo de dias desejado por semana do mesmo (MDS^p). As Equações (49) e (50) trabalham em conjunto para verificar se cada professor ministra aula ou não em um determinado dia, as Equações (51) verificam se existe o excesso de dias e qual é o valor desse excesso e as Equações (53) realizam a contabilização dessa penalidade de todos os professores. Essa restrição se justifica pela mesma razão que a restrição R18.

$$Y_{D,H}^p \leq W_D^p \quad \forall p \in P, \forall D = 1 \dots 5, \forall H = 1 \dots 14 \quad (49)$$

$$\sum_{H=1}^{14} Y_{D,H}^p \geq W_D^p \quad \forall p \in P, \forall D = 1 \dots 5 \quad (50)$$

$$\sum_{D=1}^5 W_D^p - MDS^p \leq EDS^p \quad \forall p \in P \quad (51)$$

$$EDS^p \geq 0 \quad \forall p \in P \quad (52)$$

$$P7 = \sum_{P=1}^{|\rho|} (EDS^p * PENEDS^p) \quad (53)$$

R22 – Penalizar o conflito de horário entre turmas de grupos específicos, através do parâmetro de entrada $PCONFLITO^g$, que especifica um valor de penalidade de 0 a 10 para o caso de haver conflito. Grupos conflitantes são grupos que pertencem ao conjunto $CONFIG$.

$$Z_{D,H}^g + Z_{D,H}^t - R_{D,H}^{g,t} \leq 1 \quad \forall D = 1 \dots 5, \forall H = 1 \dots 7, \forall (g,t) \in CONFIG \quad (54)$$

$$P8 = \sum_{D=1}^5 \sum_{H=1}^{14} \sum_{(g,t) \in CONFIG} (R_{D,H}^{g,t} * PCONFLITO^g) \quad (55)$$

A função objetivo busca **minimizar** o total das penalidades adquiridas pelos professores e grupos e é apresentada nas Equações (56).

$$\begin{aligned} P0 + P1 + \sum_{p=1}^{|P|} P2^p + \sum_{g=1}^{|G|} P3^g + \sum_{p=1}^{|P|} P4^p + \sum_{p=1}^{|P|} P5^p + \sum_{g=1}^{|G|} P6^g \\ + P7 + P8 + 5 * \sum_{i=0}^{T-1} \sum_{H=1}^7 \sum_{D=1}^5 \sum_{j=1}^2 U_{D,H}^{i,j} \end{aligned} \quad (56)$$

3.3 – Exemplo relacionado ao perfil de turmas

Esta seção tem o intuito de melhor explicar o conceito do perfil de turmas através do uso das variáveis $V_D^{i,k}$, $X_{D,H}^i$ e $U_{D,H}^{i,j}$. Para isto, serão exemplificadas situações práticas para o caso de turmas com carga horária de 6 horas semanais.

A Tabela 1 considera uma turma com perfil de 2 aulas de 3 horas de duração cada. Supondo que os horários e dias alocados para esta turma tenham sido conforme exibidos na tabela, as variáveis possuiriam os valores exibidos nas respectivas linhas. A variável $V_1^{i,3}$, quando é igual a 1, indica que a turma i , no dia 1 (segunda-feira) possui uma aula de 3 horas de duração. A variável $X_{1,3}^i$, quando igual a 1, indica que a turma i possui aula no dia 1 (segunda-feira), no horário 3 (11h-13h). A variável $U_{1,2}^{i,2}$, por sua vez, indica que a turma i , no dia 1 (segunda-feira) terá a segunda metade do horário 2 (9h-11h) com aula, ou seja, terá aula no horário de (10h-11). Sendo assim, é formada uma aula com duração de 3 horas na

segunda-feira (10h-13h). Em todos os casos apresentados nas tabelas a seguir (Tabela 1, 2 e 3), o valor informado das variáveis segue esta mesma lógica.

| Dia | Horário | Variável de controle | Variáveis de horário |
|---------------|----------------|---|--|
| Segunda-feira | 10h-13h | $V_1^{i,2} = 0$ $V_1^{i,3} = 1$ $V_1^{i,4} = 0$ | $X_{1,3}^i = 1$ $U_{1,2}^{i,2} = 1$ |
| Quarta-feira | 14h-17h | $V_3^{i,2} = 0$ $V_3^{i,3} = 1$ $V_3^{i,4} = 0$ | $X_{3,4}^i = 1$ $U_{3,5}^{i,1} = 1$ |

Tabela 1. Exemplo de turma com perfil de duas aulas de 3 horas

A Tabela 2 considera uma turma com perfil de 3 aulas de 2 horas de duração cada. Supondo que os horários e dias alocados para esta turma tenham sido conforme exibidos na tabela, as variáveis possuiriam os valores exibidos nas respectivas linhas.

| Dia | Horário | Variável de controle | Variáveis de horário |
|--------------|----------------|---|-----------------------------|
| Terça-feira | 9h-11h | $V_2^{i,2} = 1$ $V_2^{i,3} = 0$ $V_2^{i,4} = 0$ | $X_{2,2}^i = 1$ |
| Quinta-feira | 14h-16h | $V_4^{i,2} = 1$ $V_4^{i,3} = 0$ $V_4^{i,4} = 0$ | $X_{4,4}^i = 1$ |
| Sexta-feira | 16h-18h | $V_5^{i,2} = 1$ $V_5^{i,3} = 0$ $V_5^{i,4} = 0$ | $X_{5,5}^i = 1$ |

Tabela 2. Exemplo de turma com perfil de três aulas de 2 horas

A Tabela 3 considera uma turma com perfil de uma aula de 2 horas e uma aula de 4 horas de duração. Supondo que os horários e dias alocados para esta turma tenham sido conforme exibidos na tabela, as variáveis possuiriam os valores exibidos nas respectivas linhas.

| Dia | Horário | Variável de controle | Variáveis de horário |
|---------------|----------------|---|------------------------------------|
| Segunda-feira | 14h-18h | $V_1^{i,2} = 0$ $V_1^{i,3} = 0$ $V_1^{i,4} = 1$ | $X_{1,4}^i = 1$ $X_{1,5}^i = 1$ |
| Quinta-feira | 11h-13h | $V_4^{i,2} = 1$ $V_4^{i,3} = 0$ $V_4^{i,4} = 0$ | $X_{4,3}^i = 1$ |

Tabela 3. Exemplo de turma com perfil de aula de 2 horas e de 4 horas

4 – Modelagem do sistema

4.1 – Requisitos

Este projeto visa disponibilizar um sistema completo e de fácil utilização para a geração automática do quadro de horários em um período letivo. O sistema tem os seguintes requisitos funcionais:

1. O sistema deve ser capaz de gerenciar informações dos usuários (professores) do sistema (inserção, atualização e exclusão de dados do usuário).
2. O sistema deve ser capaz de gerenciar informações das turmas, grupos de turmas e departamento.
3. O sistema deve ser capaz de gerar um quadro de horários a partir das informações cadastradas no sistema e respeitando as restrições descritas no capítulo anterior.

4.2 – Casos de uso

O diagrama de caso de uso é apresentado na Figura 1.

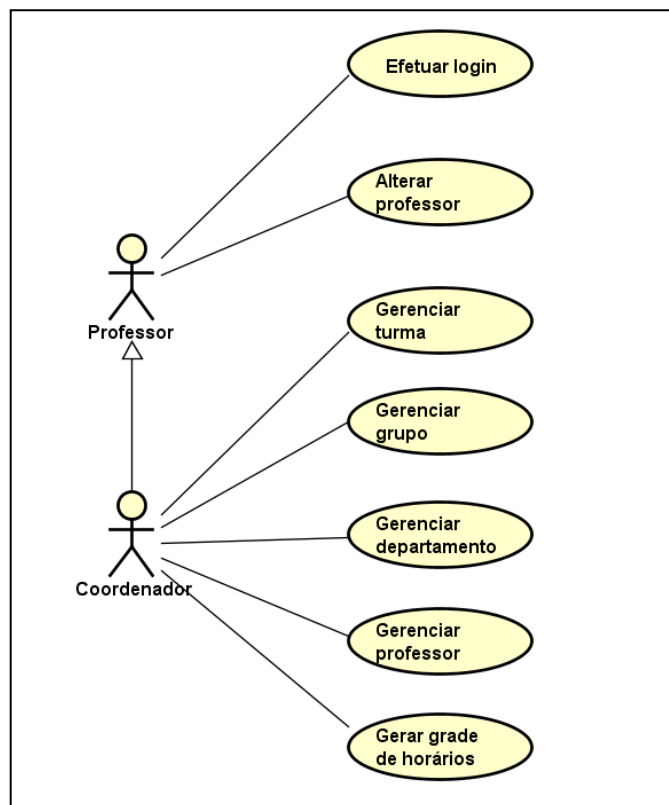


Figura 1. Diagrama de caso de uso

4.2.1 – Efetuar Login (CSU01)

Ator primário: Professor

Descrição: Professor realiza a sua autenticação no sistema.

Fluxo Principal:

1. Professor solicita efetuar sua autenticação.
2. Sistema requisita o login e a senha.
3. Professor fornece login e senha.
4. Sistema verifica que os dados de acesso estão corretos e realiza a autenticação.

Fluxo de Exceção 1 (passo 4 do fluxo principal): professor não cadastrado

1. Sistema verifica que o usuário informado não existe, informa o erro e o caso de uso continua a partir do passo 2 do Fluxo Principal.

Fluxo de Exceção 2 (passo 4 do fluxo principal): professor não autenticado

1. Sistema verifica que os dados de acesso não estão corretos, informa o erro e o caso de uso continua a partir do passo 2 do Fluxo Principal.

Pós-condições: Professor está autenticado no sistema.

4.2.2 – Alterar professor (CSU02)

Ator primário: Professor

Descrição: Professor realiza a alteração de seu cadastro.

Pré-condições: Professor precisa estar autenticado no sistema.

Fluxo Principal:

1. Professor solicita a consulta de seus dados cadastrais.
2. Sistema exibe os dados armazenados sobre o professor.
3. Professor solicita a alteração dos seus dados cadastrais.
4. Sistema exibe um formulário com os campos de todos os dados armazenados sobre o professor.
5. Professor atualiza os campos que deseja alterar.
6. Professor confere os dados e confirma a alteração.
7. Sistema registra a alteração dos dados do professor.

Fluxo de Exceção 1 (passo 7 do fluxo principal): erro nos dados fornecidos

1. Sistema detecta que algum dado não foi preenchido corretamente ou que algum campo está em branco e informa o erro.
2. Professor corrige a informação que foi lançada erroneamente.

3. Sistema aceita a correção e o caso de uso continua a partir do passo original.

Fluxo de Exceção 2 (passo 7 do fluxo principal): erro ao completar ação

1. Sistema detecta que houve algum problema ao completar a ação, informa o erro e o caso de uso continuar a partir do passo original.

Pós-condições: Professor tem seus dados alterados no sistema.

4.2.3 – Gerenciar turma (CSU03)

Ator primário: Coordenador

Descrição: Coordenador realiza a inclusão, exclusão e alteração de turmas com suas respectivas informações.

Pré-condições: Coordenador deve estar autenticado no sistema.

Fluxo Principal:

1. Coordenador solicita a inclusão de turma.
2. Sistema requisita os seguintes dados da turma: código da disciplina, código da turma, carga horária semanal, nome da disciplina, período, perfil de distribuição das aulas, professor, grupo (se a mesma pertencer a algum) e conjunto de horários obrigatórios (se a turma possuir um horário pré-definido).
3. Coordenador fornece os dados solicitados.
4. Coordenador confere os dados e confirma a inclusão da turma.
5. Sistema registra a inclusão da turma.

Fluxo Alternativo 1 (passo 1 do fluxo principal): exclusão de turmas

1. Coordenador solicita a consulta de turmas.
2. Sistema solicita o nome da turma desejada ou a opção de consultar todas as turmas cadastradas.
3. Coordenador informa o nome da turma desejada ou escolhe a opção de consultar todas as turmas.
4. Sistema exibe a lista de turmas correspondente à opção escolhida.
5. Coordenador seleciona a turma que deseja excluir.
6. Sistema retira a turma do sistema.

Fluxo Alternativo 2 (passo 5 do fluxo alternativo 1): alteração de dados das turmas

1. Coordenador seleciona uma turma e solicita a alteração dos dados da mesma.
2. Sistema exibe um formulário com os campos de todos os dados armazenados sobre a turma.

3. Coordenador atualiza os campos que deseja alterar.
4. Coordenador confere os dados e confirma a alteração das características da turma.
5. Sistema registra a alteração dos dados da turma.

Fluxo de Exceção 1 (passo 5 do fluxo principal e passo 5 do fluxo alternativo 2): erro nos dados fornecidos

1. Sistema detecta que algum dado não foi preenchido corretamente ou que algum campo está em branco e informa o erro.
2. Coordenador corrige a informação que foi lançada erroneamente.
3. Sistema aceita a correção e o caso de uso continua a partir do passo original.

Fluxo de Exceção 2 (passo 5 do fluxo principal, passo 6 do fluxo alternativo 1 e passo 5 do fluxo alternativo 2): erro ao completar ação

1. Sistema detecta que houve algum problema ao completar a ação, informa o erro e o caso de uso continuar a partir do passo original.

Pós-condições: Turmas são incluídas, excluídas ou alteradas no sistema.

4.2.4 – Gerenciar grupo (CSU04)

Ator primário: Coordenador

Descrição: Coordenador realiza a inclusão, exclusão e alteração de grupos com suas respectivas características.

Pré-condições: Coordenador deve estar autenticado no sistema.

Fluxo Principal:

1. Coordenador solicita a inclusão de grupo.
2. Sistema requisita os seguintes dados do grupo: nome, número máximo de aulas desejado por dia, conjunto de horários censurados (se existir algum), conjunto de grupos conflitantes (se existir algum), penalidade por excesso de aulas em um dia, penalidade por existir conflito com algum grupo, penalidade para cada um dos 5 tamanhos de horários vagos em um dia e penalidade de ter uma aula alocada em cada um dos horários possíveis da semana.
3. Coordenador fornece os dados solicitados do grupo.
4. Coordenador confere os dados e confirma a inclusão do grupo.
5. Sistema registra a inclusão do grupo.

Fluxo Alternativo 1 (passo 1 do fluxo principal): exclusão de grupos

1. Coordenador solicita a consulta de grupos.
2. Sistema exhibe a lista de grupos cadastrados.

3. Coordenador seleciona qual grupo deseja excluir.
4. Sistema retira o grupo selecionado do sistema.

Fluxo Alternativo 2 (passo 1 do fluxo principal): alteração de dados dos grupos

1. Coordenador solicita a consulta de grupos.
2. Sistema exibe a lista de grupos cadastrados.
3. Coordenador seleciona o grupo que deseja alterar.
4. Sistema exibe um formulário com os campos de todos os dados armazenados sobre o grupo.
5. Coordenador atualiza os campos que deseja alterar.
6. Coordenador confere os dados e confirma a alteração das características do grupo.
7. Sistema registra a alteração dos dados do grupo.

Fluxo de Exceção 1 (passo 5 do fluxo principal e passo 7 do fluxo alternativo 2): erro nos dados fornecidos

1. Sistema detecta que algum dado não foi preenchido corretamente ou que algum campo está em branco e informa o erro.
2. Coordenador corrige a informação que foi lançada erroneamente.
3. Sistema aceita a correção e o caso de uso continua a partir do passo original.

Fluxo de Exceção 2 (passo 5 do fluxo principal, passo 4 do fluxo alternativo 1 e passo 7 do fluxo alternativo 2): erro ao completar ação

1. Sistema detecta que houve algum problema ao completar a ação, informa o erro e o caso de uso continuar a partir do passo original.

Pós-condições: Grupos são incluídos, excluídos ou alterados no sistema.

4.2.5 – Gerenciar departamento (CSU05)

Ator primário: Coordenador

Descrição: Coordenador realiza a inclusão e alteração do departamento com suas respectivas características.

Pré-condições: Coordenador deve estar autenticado no sistema.

Fluxo Principal:

1. Coordenador solicita a inclusão de departamento.
2. Sistema requisita os seguintes dados do departamento: número de salas disponíveis e conjuntos de horários proibidos (se existir algum).
3. Coordenador fornece os dados solicitados.

4. Coordenador confere os dados e confirma a inclusão do departamento.
5. Sistema registra a inclusão do departamento.

Fluxo Alternativo 1 (passo 1 do fluxo principal): alteração de dados do departamento

1. Coordenador solicita a alteração dos dados do departamento.
2. Sistema exibe um formulário com os campos de todos os dados armazenados sobre o departamento.
3. Coordenador atualiza os campos que deseja alterar.
4. Coordenador confere os dados e confirma a alteração das características do departamento.
5. Sistema registra a alteração dos dados da turma.

Fluxo de Exceção 1 (passo 5 do fluxo principal e passo 5 do fluxo alternativo 1): erro nos dados fornecidos

1. Sistema detecta que algum dado não foi preenchido corretamente ou que algum campo está em branco e informa o erro.
2. Coordenador corrige a informação que foi lançada erroneamente.
3. Sistema aceita a correção e o caso de uso continua a partir do passo original.

Fluxo de Exceção 2 (passo 5 do fluxo principal e passo 5 do fluxo alternativo 1): erro ao completar ação

1. Sistema detecta que houve algum problema ao completar a ação, informa o erro e o caso de uso continuar a partir do passo original.

Pós-condições: Departamento é incluído ou alterado no sistema.

4.2.6 – Gerenciar professor (CSU06)

Ator primário: Coordenador

Descrição: Coordenador realiza a troca do cargo de coordenador ou a inclusão e exclusão de professores com suas respectivas características.

Pré-condições: Coordenador precisa estar autenticado no sistema.

Fluxo Principal:

1. Coordenador solicita a inclusão de professor.
2. Sistema requisita os seguintes dados do professor: nome, número máximo de dias por semana e número máximo de aulas por dia que ele deseja ministrar, penalidade para o excesso de aulas em um dia, penalidade para o excesso de dias alocados na semana, penalidade para cada um dos 5 tamanhos de horários vagos em um dia, penalidade para cada um dos 3

tamanhos de horários vagos na semana, penalidade de ter uma aula alocada em cada um dos horários possíveis da semana, login, senha e cargo.

3. Coordenador fornece os dados solicitados.
4. Coordenador confere os dados e confirma a inclusão do professor.
5. Sistema registra a inclusão do professor.

Fluxo Alternativo 1 (passo 1 do fluxo principal): exclusão de professores

1. Coordenador solicita a consulta de professores.
2. Sistema exhibe a lista dos professores cadastrados.
3. Coordenador seleciona qual professor deseja excluir.
4. Sistema retira o professor selecionado do sistema.

Fluxo Alternativo 2 (passo 1 do fluxo principal): troca de cargo de coordenador

1. Coordenador solicita a consulta de professores.
2. Sistema exhibe a lista dos professores cadastrados.
3. Coordenador seleciona qual professor deseja tornar coordenador.
4. Sistema altera o cargo do coordenador para “professor” e do professor selecionado para “coordenador”.

Fluxo de Exceção 1 (passo 5 do fluxo principal): erro nos dados fornecidos

1. Sistema detecta que algum dado não foi preenchido corretamente ou que algum campo está em branco e informa o erro.
2. Coordenador corrige a informação que foi lançada erroneamente.
3. Sistema aceita a correção e o caso de uso continua a partir do passo original.

Fluxo de Exceção 2 (passo 5 do fluxo principal, passo 4 do fluxo alternativo 1 e passo 4 do fluxo alternativo 2): erro ao completar ação

1. Sistema detecta que houve algum problema ao completar a ação, informa o erro e o caso de uso continuar a partir do passo original.

Pós-condições: Professores são incluídos, excluídos ou tem seu cargo alterado no sistema.

4.2.7 – Gerar grade de horários (CSU07)

Ator primário: Coordenador

Descrição: Coordenador gera a grade de horários do semestre.

Pré-condições: Coordenador deve estar autenticado no sistema.

Fluxo Principal:

1. Coordenador solicita a criação de uma nova grade de horários.

2. Sistema solicita o caminho do arquivo que irá conter o quadro de horários gerado.
3. Coordenador informa o caminho do arquivo.
4. Sistema gera um arquivo no formato PDF no caminho indicado com a grade de horários do semestre.

Fluxo de Exceção 1 (passo 4 do fluxo principal): sistema não encontra solução para as restrições fornecidas

1. Sistema informa que não foi possível encontrar uma solução com os dados informados.

Pós-condições: Arquivo no formato PDF é criado com a grade de horários.

4.3 – Diagrama de classes

A Figura 2 apresenta o Diagrama de Classes, que representa a estrutura e relações das classes que servem de modelo para objetos [18]. O tipo de dado “Horário” utilizado no diagrama representa uma combinação de dia, horário e metade do horário. Para fins de simplificação, a classe auxiliar Horário não é exibida no diagrama.

Buscando uma melhor visibilidade, alguns métodos foram ocultados no Diagrama de Classes, como os métodos *get* e *set*, por exemplo. Além disso, os métodos *escreve_formulacao* das classes Professor, Turma, Grupo e Departamento englobam um conjunto de outros métodos, que são listados a seguir.

Classe Professor

+ *escreve_formulacao_prof()*: void inclui os seguintes métodos:

- + *escreve_conf_horario_professor()*: void
- + *escreve_controla_metades_p1()*: void
- + *escreve_controla_metades_p2()*: void
- + *escreve_excessoauladia()*: void
- + *escreve_excessodiasemana()*: void
- + *escreve_minburaco_prof()*: void
- + *escreve_minburacodias_prof()*: void
- + *escreve_ministra_auladia_p1()*: void
- + *escreve_ministra_auladia_p2()*: void
- + *escreve_penburacodia()*: void
- + *escreve_penburacosemana()*: void
- + *escreve_penh_prof()*: void

- + evitar_horarioconsec(): void
- + maxaulasdia_prof(): void
- + minqtddias_prof(): void

Classe Turma

+ *escreve_formulacao_turma()*: void inclui os seguintes métodos:

- + escreve_cargahoraria(): void
- + escreve_controla_metades_t1(): void
- + escreve_controla_metades_t2(): void
- + escreve_controleperfil(): void
- + escreve_horarioalmoco(): void
- + escreve_horariosobrig(): void
- + escreve_perfildia(): void
- + escreve_proibir_buraco_t(): void
- + escreve_relacionaperfil(): void

Classe Grupo

+ *escreve_formulacao_grupo()*: void inclui os seguintes métodos:

- + escreve_censuradosgrupo(): void
- + escreve_conf_horario_grupos(): void
- + escreve_conflitos_grupos(): void
- + escreve_controla_metades_g1(): void
- + escreve_controla_metades_g2(): void
- + escreve_excessoauladia(): void
- + escreve_minburaco_grupo(): void
- + escreve_penburacodia(): void
- + escreve_penconfhorario(): void
- + escreve_penh_grupo(): void
- + maxaulasdia_grupo(): void

Classe Departamento

+ *escreve_formulacao_dpto()*: void inclui os seguintes métodos:

- + escreve_maxsalas(): void
- + escreve_proibidos(): void

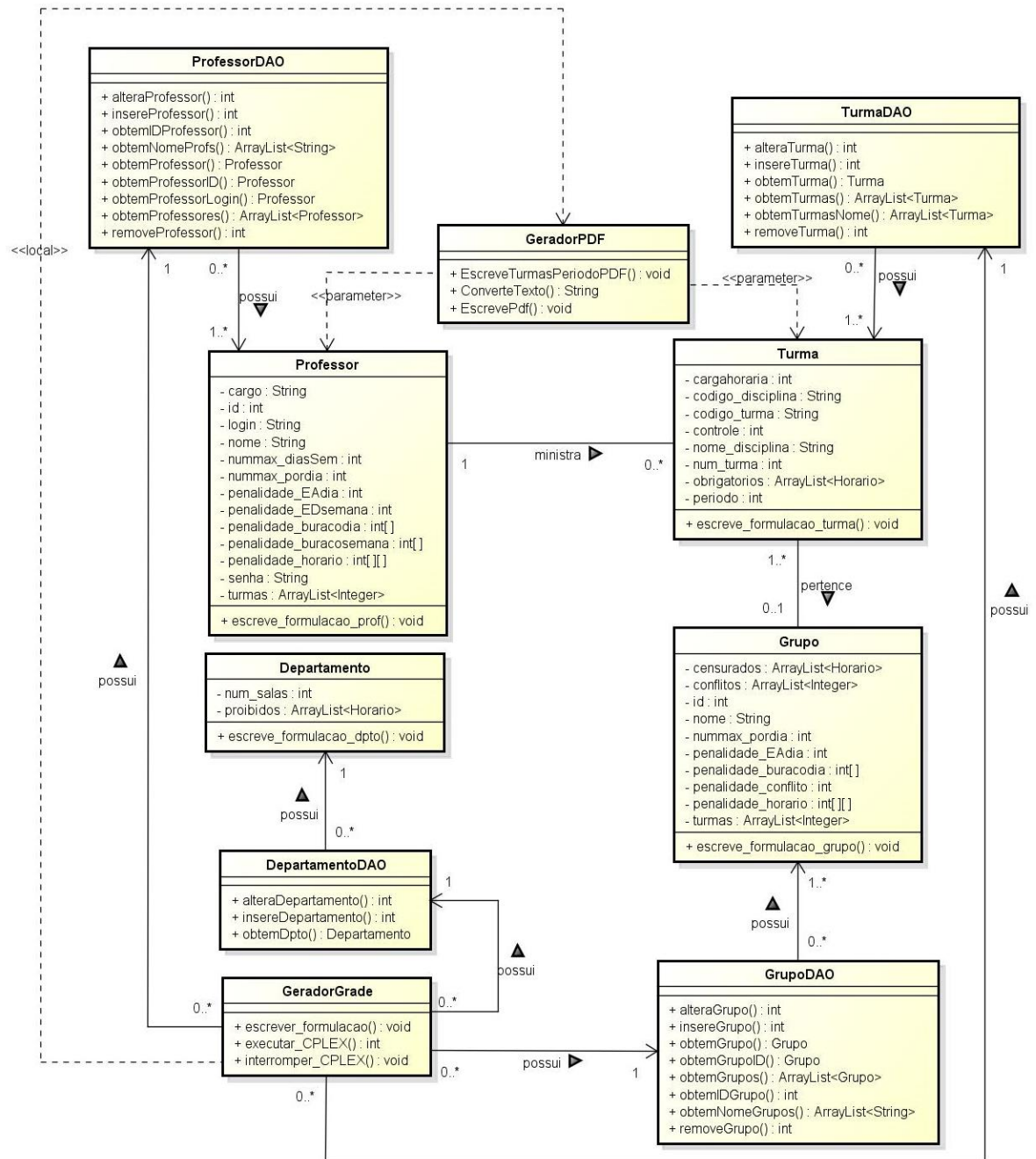


Figura 2. Diagrama de Classes

4.4 – Banco de dados

A Figura 3 apresenta o modelo Entidade Relacionamento, que representa de forma abstrata a estrutura que o banco de dados da aplicação possui. O banco de dados utilizado para a aplicação foi o MySQL Workbench 6.3 e o servidor utilizado foi o MySQL Server 5.7.

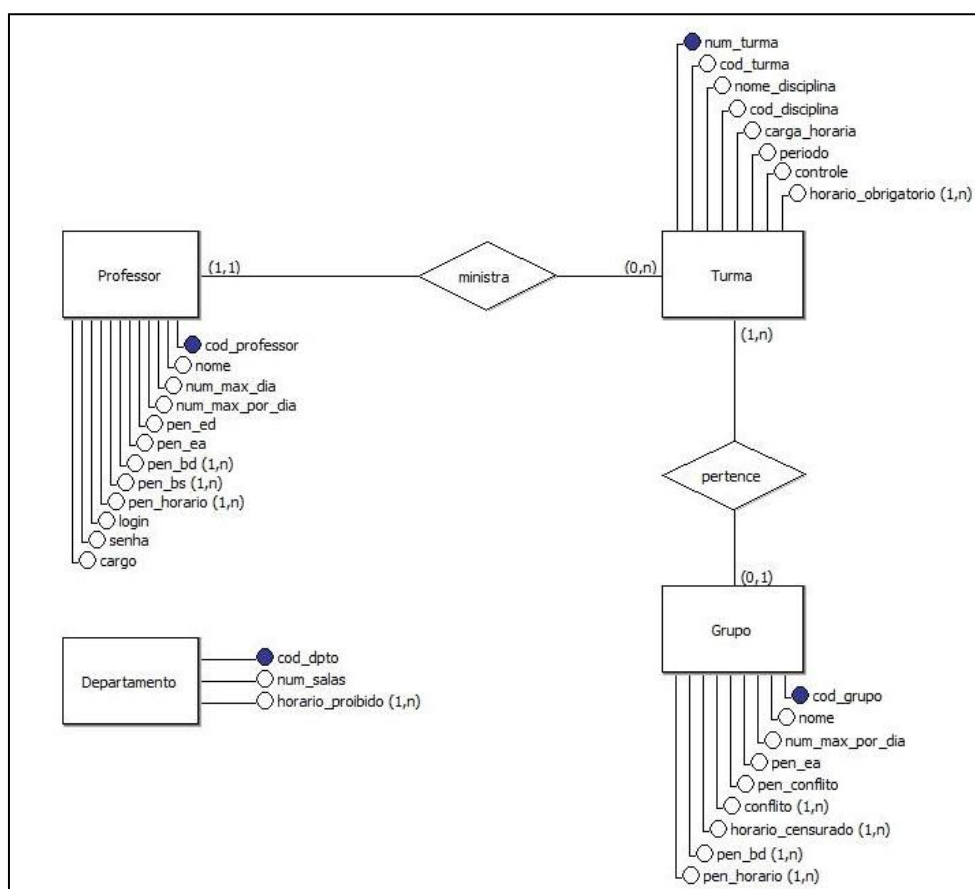


Figura 3. Modelo Entidade Relacionamento

Seguindo a ordem apresentada na figura, os atributos de Professor representam, respectivamente, o código do professor, nome, número máximo de dias que o mesmo deseja ir à instituição, número máximo de aulas por dia que o mesmo deseja ministrar, penalidade por excesso de dias na semana, penalidade por excesso de aulas em um dia, penalidade para cada um dos 5 possíveis tamanhos de intervalo de tempo vago em um dia, penalidade para cada um dos 3 possíveis tamanhos de intervalo de tempo vago na semana, penalidade para a alocação em cada um dos dias e horários possíveis, login, senha e cargo.

Os atributos de Turma representam, respectivamente, o número da turma, código da turma, nome da disciplina, código da disciplina, carga horária, período, controle (perfil da turma) e o conjunto de horários obrigatórios (pré-fixados).

Os atributos de Grupo representam, respectivamente, o código do grupo, nome do grupo, número máximo de aulas por dia desejado para as turmas do grupo, penalidade por excesso de aulas em um dia, penalidade por conflito com outro grupo, conjunto de grupos que são conflitantes, conjunto de horários censurados, penalidade para cada um dos 5 possíveis

tamanhos de intervalo de tempo vago em um dia e penalidade para a alocação em cada um dos dias e horários possíveis.

Os atributos de Departamento representam, respectivamente, o código do departamento, o número de salas disponíveis para a alocação e o conjunto de horários proibidos.

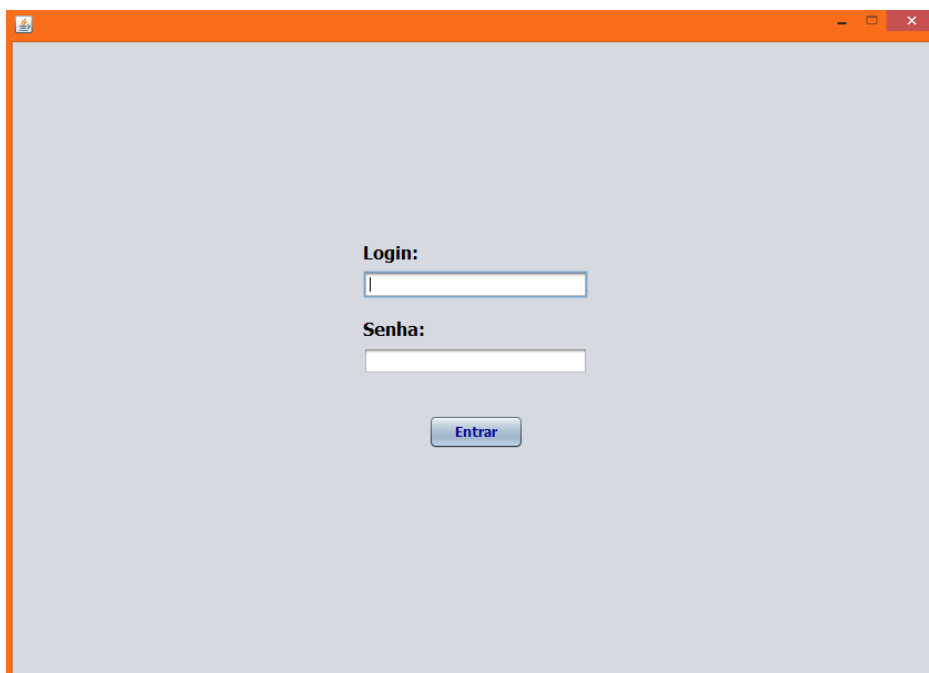
O banco de dados contém outras tabelas intermediárias além das apresentadas no modelo da Figura 3, sendo uma para cada atributo multivalorado. Tais entidades não são relevantes para esta representação gráfica, uma vez que esta é apenas uma visão geral da estrutura do banco de dados.

4.5 – Apresentação do sistema

O sistema foi desenvolvido utilizando linguagem Java. Através do mesmo, o coordenador e os professores do curso podem inserir todos os dados necessários para a construção da grade de horários. A partir desses dados, a formulação pode ser criada e então processada pelo Cplex. A solução gerada é formatada e exibida em um arquivo no formato PDF. Para a geração desse arquivo, foi utilizada a biblioteca Itext 5.4.3 [19].

O sistema está dividido em: Tela de autenticação com login e senha; Tela principal contendo o menu de acesso à todas as outras telas; Telas de cadastros, tais como cadastro dos professores, turmas, grupos e departamento; Telas de consulta e alteração dos dados cadastrados e Tela de geração do quadro de horários. As telas e o arquivo com o quadro de horários são apresentados nas seções a seguir.

4.5.1 – Tela de autenticação



The screenshot shows a web browser window with an orange border. The main content area is light gray. In the center, there is a login form with the following elements:

- Login:** A text label followed by a white input field.
- Senha:** A text label followed by a white input field.
- Entrar:** A blue button with white text, positioned below the password field.

Figura 4. Tela de login

4.5.2 – Tela principal



Figura 5. Tela inicial

4.5.3 – Telas relacionadas ao professor

The screenshot shows a web application window titled 'Cadastro de Professor' with two tabs: 'Informações Gerais' and 'Penalidades de horário'. The 'Informações Gerais' tab is active. The form contains the following fields:

- Cargo:** dropdown menu with 'professor' selected.
- Nome:** text input field.
- Login:** text input field.
- Senha:** text input field.
- Número máximo de aulas por dia:** dropdown menu with '1' selected.
- Penalidade p/ excesso de aulas:** dropdown menu with '0' selected.
- Número máx de dias por semana:** dropdown menu with '1' selected.
- Penalidade p/ excesso de dias:** dropdown menu with '0' selected.
- Penalidade p/ quantidade de dias vagos:** three dropdown menus labeled '1:', '2:', and '3:', all with '0' selected.
- Penalidade p/ quantidade de aulas vagas:** five dropdown menus labeled '1:', '2:', '3:', '4:', and '5:', all with '0' selected.

At the bottom, there is a note: '(0 - sem penalidade, 10 - penalidade máxima)'. Two buttons are present: 'Cancelar' and 'Avançar'.

Figura 6. Tela de cadastro de professor (1)

The screenshot shows the same 'Cadastro de Professor' window, but with the 'Penalidades de horário' tab active. The form displays a grid of penalty settings for each day of the week and time slot:

| Day | 7h - 9h: | 9h - 11h: | 11h - 13h: | 14h - 16h: | 16h - 18h: | 18h - 20h: | 20h - 22h: |
|----------------|----------|-----------|------------|------------|------------|------------|------------|
| Segunda-feira: | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Terça-feira: | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Quarta-feira: | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Quinta-feira: | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Sexta-feira: | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

At the bottom, there is a note: '(0 - sem penalidade, 10 - penalidade máxima)'. Two buttons are present: 'Cancelar' and 'Cadastrar'.

Figura 7. Tela de cadastro de professor (2)

Consultar Professor

| Nome | Login | Cargo |
|--------------|-------------|-------------|
| Danilo | Danilo | coordenador |
| P0 | Prof 0 | professor |
| Leandro | Leandro | professor |
| Marcos | Marcos | professor |
| André Renato | Andre | professor |
| Sergio | Sergio | professor |
| Copetti | Copetti | professor |
| Patrick | Patrick | professor |
| Marcilene | Marcilene | professor |
| Dalessandro | Dalessandro | professor |
| Bazilio | Bazilio | professor |
| Luciano | Luciano | professor |

Tornar coord. Remover Ver mais Voltar

Figura 8. Tela de consulta de professores

4.5.4 – Telas relacionadas à turma

Cadastro de Turma

Nome da disciplina: Carga horária: 2

Código da disciplina: Grupo: Sem grupo

Código da turma: Período: 1º

Professor: André Renato Perfil: 1 aula de 2h

Selecione a lista de horários para as aulas (opcional):

Dia: Segunda Horário: 7h - 8h

Adicionar horário Remover horário

Lista de horários pré-definidos:

| Dia | Horário |
|-----|---------|
|-----|---------|

Cancelar Cadastrar

Figura 9. Tela de cadastro de turma

Consultar Turma

Nome da disciplina:

| Nome | Codigo Turma | Codigo Disciplina | Professor |
|---|--------------|-------------------|-----------|
| Geometria Analítica e Cálculo Vetorial | V1 | RCT0021 | P0 |
| Geometria Analítica e Cálculo Vetorial | K1 | RCT0021 | P0 |
| Geometria Analítica e Cálculo Vetorial | R1 | RCT0021 | P0 |
| Cálculo I | V1 | RCT0016 | P0 |
| Cálculo I | R1 | RCT0016 | P0 |
| Introdução à Ciência da Computação | V1 | RCT0001 | Leandro |
| Fundamentos de Arquitetura e Computado... | V1 | RCT00013 | Marcos |

Figura 10. Tela de consulta de turmas

4.5.5 – Telas relacionadas ao grupo

Cadastro de Grupo

Nome:

Número máximo de aulas por dia:

Penalidade p/ excesso de aulas:

Horários Censurados

7h - 9h 9h - 11h 11h - 13h

14h - 16h 16h - 18h 18h - 20h

20h - 22h

Penalidade p/ quantidade de aulas vagas:

1: 2: 3: 4: 5:

Grupos Conflitantes:

Grupo

Penalidade p/ grupos conflitantes:

(0 - sem penalidade, 10 - penalidade máxima)

Figura 11. Tela de cadastro de grupo (1)

Informações Gerais | Penalidades de horário

Cadastro de Grupo

Segunda-feira:

| | | | | | | |
|----------|-----------|------------|------------|------------|------------|------------|
| 7h - 9h: | 9h - 11h: | 11h - 13h: | 14h - 16h: | 16h - 18h: | 18h - 20h: | 20h - 22h: |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Terça-feira:

| | | | | | | |
|----------|-----------|------------|------------|------------|------------|------------|
| 7h - 9h: | 9h - 11h: | 11h - 13h: | 14h - 16h: | 16h - 18h: | 18h - 20h: | 20h - 22h: |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Quarta-feira:

| | | | | | | |
|----------|-----------|------------|------------|------------|------------|------------|
| 7h - 9h: | 9h - 11h: | 11h - 13h: | 14h - 16h: | 16h - 18h: | 18h - 20h: | 20h - 22h: |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Quinta-feira:

| | | | | | | |
|----------|-----------|------------|------------|------------|------------|------------|
| 7h - 9h: | 9h - 11h: | 11h - 13h: | 14h - 16h: | 16h - 18h: | 18h - 20h: | 20h - 22h: |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Sexta-feira:

| | | | | | | |
|----------|-----------|------------|------------|------------|------------|------------|
| 7h - 9h: | 9h - 11h: | 11h - 13h: | 14h - 16h: | 16h - 18h: | 18h - 20h: | 20h - 22h: |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

(0 - sem penalidade, 10 - penalidade máxima)

Cancelar Cadastrar

Figura 12. Tela de cadastro de grupo (2)

Consultar Grupo

| Nome |
|-----------|
| 1 periodo |
| 2 periodo |
| 3 periodo |

Remover Ver mais Voltar

Figura 13. Tela de consulta de grupos

4.5.6 – Tela de departamento

Dados Departamento

Número de salas:

Selecione a lista de horários proibidos (opcional):

Dia: Horário:

Lista de horários proibidos:

| Dia | Horário |
|--------|-----------|
| Quinta | 16h - 17h |
| Quinta | 17h - 18h |

Figura 14. Tela de alteração de departamento

4.5.7 – Tela de geração da grade

Gerar quadro de horários

Escolha o caminho para o arquivo de saída:

Figura 15. Tela de geração de grade

4.5.8 – Quadro de horários gerado

A Figura 16 apresenta um exemplo do quadro de horários gerado em PDF pelo software desenvolvido, possuindo as seguintes informações: código da disciplina, nome da disciplina, código da turma, nome do professor e o horário das aulas nos dias da semana. As cores são usadas para fazer a separação das disciplinas por períodos, iniciando no 1º período e seguindo em ordem crescente até o 8º período.

Quadro de Horários

| Código | Disciplina | Turma | Professor | Seg | Ter | Qua | Qui | Sex | Sab |
|----------|---|-------|-----------------|---------|--------------------|---------|---------|---------|-----|
| RCT00021 | Geometria Analítica e Cálculo Vetorial | V1 | P0 | | 9h-11h | | 9h-11h | | |
| RCT00021 | Geometria Analítica e Cálculo Vetorial | R1 | P0 | | 11h-13h | | 16h-18h | | |
| RCT00016 | Cálculo I | V1 | P0 | | 11h-13h | | 11h-13h | | |
| RCT00016 | Cálculo I | R1 | P0 | | 16h-18h | 16h-18h | | | |
| RCT00001 | Introdução à Ciência da Computação | V1 | Leandro | | | | | 9h-11h | |
| RCT00013 | Fundamentos de Arquitetura e | V1 | Marcos | 9h-11h | | 14h-16h | | | |
| RCT00002 | Programação de Computadores I | V1 | Sergio | | | 9h-12h | | 14h-17h | |
| RCN00024 | Algebra Linear | C1 | P0 | | | 9h-11h | 9h-11h | | |
| RFM00002 | Calculo II | C1 | P0 | | | 14h-16h | 7h-9h | | |
| RCT00003 | Programação de Computadores II | C1 | Copetti | 9h-12h | 9h-12h | | | | |
| RCT00023 | Lógica para Ciência da Computação | C1 | Marcilene | 14h-16h | 14h-16h | | | | |
| RFM00007 | Fisica I | C1 | P0 | | | 11h-13h | | 11h-13h | |
| RFM00010 | Equações Diferenciais | R1 | P0 | | 9h-11h | | 9h-11h | | |
| RCN00027 | Fisica II | C1 | P0 | 11h-13h | | 11h-13h | | | |
| RCM00004 | Estrutura de Dados | C1 | Dalessandro | | | 9h-11h | | 14h-16h | |
| RCM00006 | Programação com arquivos | C1 | Marcos | | | | | 11h-13h | |
| RCM00011 | Circuitos Digitais | C1 | Luciano | 9h-11h | | | 11h-13h | | |
| RCT00032 | Técnicas de Programação Avançada | C1 | Eduardo Marques | 7h-9h | | 9h-11h | | | |
| RCM00009 | Análise de Projetos de Algoritmos | C1 | Danilo | | 9h-11h | | | 16h-18h | |
| RCM00021 | Banco de Dados | C1 | Leila | 14h-16h | 7h-9h | | | | |
| RCM00023 | Engenharia de Software | C1 | Adriana | | 16h-18h | | | 9h-11h | |
| RCM00010 | Linguagens Formais e teoria da computação | C1 | Dalessandro | | | | 20h-22h | 11h-13h | |
| RCM00022 | Projeto de banco de dados | C1 | Bazilio | 14h-16h | | 16h-18h | | | |
| RCM00026 | Computação Gráfica | C1 | Eduardo Marques | 11h-13h | | | | 14h-16h | |
| RCM00025 | Inteligência Artificial | C1 | Eduardo Marques | 16h-18h | | | 7h-9h | | |
| RCM00019 | Rede de Computadores II | C1 | Leandro | | 9h-11h | 9h-11h | | | |
| RCM00031 | Computação e Sociedade | C1 | Sergio | 11h-13h | | | | | |
| REG00001 | Empreendedorismo | C1 | P0 | | | | 16h-18h | | |
| RCM00058 | Engenharia de Software II | C1 | Adriana | 9h-11h | | 20h-22h | | | |
| RCT00058 | Programação para internet | C1 | Bazilio | | 11h-13h 14h-16h | | | | |

Figura 16. Exemplo do PDF gerado com a grade de horário

5 – Resultados computacionais

A partir da formulação descrita no Capítulo 3, foram realizados diversos testes com o objetivo de verificar quais restrições causam maior impacto na qualidade da solução. A amostra utilizada para os testes é composta por 10 grupos, 27 professores e 60 turmas. Cada professor é responsável por 2 ou 3 turmas, o que equivale a uma média de 8 horas semanais em aulas para cada professor. Os grupos são constituídos, em média, por 6 turmas. Dentre as turmas, 12 possuem carga horária de 2 horas semanais, 37 possuem carga horária de 4 horas semanais e 11 possuem carga horária de 6 horas semanais. Além disso, tais turmas possuem diferentes perfis de distribuição das aulas e apenas 3 delas possuem horário previamente fixado.

O tamanho da amostra utilizada é válido para projeções futuras de crescimento do departamento, visto que atualmente os números são relativamente menores. No semestre atual, o curso de Ciência da Computação teve 59 turmas, sendo 24 delas pertencentes ao Departamento de Matemática e Física. Em outras palavras, se fosse utilizado o sistema para gerar o quadro de horários atual, essas 24 turmas teriam o horário previamente fixado, o que facilita consideravelmente para que o otimizador Cplex encontre a solução. Ademais, o Departamento possui 9 salas disponíveis e 16 professores, além do professor p' , representando todos os professores de outro departamento.

Para o processamento da formulação é utilizado o otimizador Cplex versão 12.5. A resolução através do otimizador foi feita em um computador com processador Intel I5 2.2 Ghz, 4 Gbytes de memória RAM, com sistema operacional Windows 8.

Os testes foram baseados em diferentes critérios, tais como o número de salas disponíveis para a alocação, a proibição de dias da semana, entre outros, e são detalhados a seguir.

5.1 – Teste baseado no número de salas

O número de salas disponíveis para a alocação variou de 12 a 7, buscando encontrar o número ideal de salas para a amostra utilizada. Os resultados são apresentados a seguir na Tabela 4.

| Número de salas | Tempo (s) | Solução |
|------------------------|------------------|----------------|
| 12 | 368,44 | 735 |
| 11 | 524,02 | 735 |
| 10 | 929,31 | 735 |
| 9 | 692,92 | 735 |
| 8 | 3195,48 | 738 |
| 7 | 3761,53* | 769* |

Tabela 4. Resultado do teste baseado no número de salas

Conforme os resultados, é possível perceber que o número de 9 salas é suficiente para a alocação, já que o tempo para gerar a solução é razoável e a qualidade da solução não piora comparado a um número maior de salas. Dessa forma, para esta instância, não é necessário um número maior de salas. Por outro lado, um número menor de salas aumenta consideravelmente o tempo para gerar a solução, além de piorar a qualidade da mesma. Para o cenário de 7 salas, o programa executou por mais de uma hora e não concluiu sua execução. Até esse momento, a diferença entre o limite dual e primal era de 0,77% e o valor da função objetivo era de 769.

5.2 – Teste baseado em evitar e proibir dias da semana

Nesse teste, as amostras de entrada foram utilizadas de forma que nenhuma aula pudesse ser alocada ou que as mesmas fossem evitadas em um dia da semana específico, por meio de penalidades muito altas nesse dia. O resultado é apresentado na Tabela 5. A coluna “Aulas no dia” indica quantas aulas permaneceram alocadas em um dia, mesmo com a tentativa de evitar essa alocação.

| Dia | Salas | Evitar | | | Proibir | |
|------------|--------------|------------------|----------------|---------------------|------------------|----------------|
| | | Tempo (s) | Solução | Aulas no dia | Tempo (s) | Solução |
| Segunda | 10 | 68,23 | 873 | 16 | 71,84 | 1020 |
| Terça | 10 | 172,58 | 851 | 12 | 119,92 | 1085 |
| Quarta | 10 | 154,34 | 849 | 14 | 132,95 | 1121 |
| Quinta | 10 | 27,02 | 854 | 16 | 85,98 | 1093 |
| Sexta | 10 | 449,14 | 849 | 15 | 133,83 | 1000 |

Tabela 5. Resultado do teste de evitar ou proibir um dia da semana

Comparando o resultado da proibição de determinados dias com o apresentado na Tabela 4, percebe-se que, apesar do tempo ser significativamente menor, a qualidade da solução piora muito com a inclusão desta restrição. Porém, é possível perceber que a qualidade da solução e o tempo variam bastante para cada um dos dias, o que mostra que o impacto dessa restrição na qualidade da solução diverge dependendo do dia restrito e da amostra de entrada em questão.

Para todos os casos de evitar algum dia da semana, a qualidade da solução piorou quando comparada ao resultado exibido na Tabela 4, mas não substancialmente. Por outro lado, o tempo necessário para gerar a solução referente a evitar a sexta-feira foi bem maior que o da quinta-feira, por exemplo, apesar de ainda ser consideravelmente menor que o apresentado na Tabela 4. Assim como no caso de proibição de dias da semana, o impacto que essa restrição causa na qualidade da solução depende da amostra de entrada e do dia que se deseja evitar.

5.3 – Teste baseado na proibição de horários específicos

Para este teste, foram proibidos intervalos de horários específicos e os resultados são exibidos na Tabela 6. Os horários de 7 e 20 horas foram escolhidos para o teste, visto que muitos professores e alunos não ficam satisfeitos com aulas nesses horários. É possível perceber que a proibição de ambos os horários não afetou a qualidade da solução, quando comparado ao resultado apresentado na Tabela 4. Em contrapartida, para o caso da proibição do horário de 20 horas, o tempo para encontrar a solução foi consideravelmente maior que o apresentado na Tabela 4.

| Horário | Número de salas | Tempo (s) | Solução |
|-----------|-----------------|-----------|---------|
| 7 às 9h | 10 | 366,67 | 735 |
| 20 às 22h | 10 | 5242,17 | 735 |

Tabela 6. Resultado do teste baseado na proibição de horários

5.4 – Teste baseado na remoção do máximo de aulas por dia

Para este teste, foram removidas as restrições que penalizavam o excesso de aulas em um dia para grupos e professores. Em outras palavras, considerou-se a não existência de qualquer limite de número de aulas por dia, tanto para os professores, quanto para os grupos. O resultado é exibido na Tabela 7.

| Número de salas | Tempo (s) | Solução |
|------------------------|------------------|----------------|
| 10 | 1128,06 | 161 |

Tabela 7. Resultado do teste baseado na remoção do máximo de aulas por dia

Sem essas penalidades o valor da função objetivo foi consideravelmente reduzido se comparado ao valor apresentado na Tabela 4. Porém, isso não significa que a qualidade prática da solução tenha sido melhorada. Com a remoção dessa restrição, um professor e sete grupos passaram a ter mais de 7 horas de aulas alocadas em um mesmo dia e essa situação, algumas vezes, ocorreu em mais de um dia para um mesmo grupo, o que causa uma sobrecarga de atividades nesses dias.

5.5 – Teste baseado na flexibilização do modelo

A flexibilização do modelo consiste no poder de escolha do professor sobre como será a distribuição das aulas de suas turmas. Para realizar esse teste, as amostras de entrada foram utilizadas de maneira que pudesse ser obtida uma comparação entre as soluções para os modelos com e sem a flexibilização dos perfis de turmas.

Primeiramente, todas as turmas com carga horária semanal de 6 horas foram consideradas com o perfil de 3 aulas semanais de 2 horas cada, ou seja, é simulado um modelo em que as aulas não possam ser distribuídas de maneira diferente da convencional (perfil original). Em seguida, as mesmas turmas foram consideradas com o perfil de 2 aulas semanais de 3 horas cada, ou seja, utilizando a flexibilização do modelo (perfil modificado). O resultado é exibido na Tabela 8.

| Número de salas | Perfil | Tempo (s) | Solução |
|------------------------|---------------|------------------|----------------|
| 10 | Original | 262,75 | 734 |
| 10 | Modificado | 180,02 | 769 |

Tabela 8. Resultado do teste baseado na flexibilização do modelo

No caso do perfil modificado, o tempo para encontrar a solução foi menor que o tempo do perfil original, porém, a qualidade da solução encontrada foi pior. Essa piora, no entanto, não foi tão elevada (aproximadamente 4,8%). Além disso, é necessário analisar outros aspectos para definir a qualidade de uma solução. Ao mesmo tempo em que se obtém um aumento razoavelmente pequeno nas penalidades, obtém-se a possibilidade do professor

escolher como deseja que as aulas de suas turmas sejam distribuídas, o que pode compensar subjetivamente. Em outras palavras, a flexibilização do modelo não piora substancialmente a solução, mas deve ser utilizado com cautela e em situações específicas.

6 – Conclusão

Neste trabalho foi proposta uma abordagem para o problema de alocação de horários em uma universidade, tendo como estudo de caso o curso de Ciência da Computação da Universidade Federal Fluminense de Rio das Ostras, com o intuito de tornar mais rápido e satisfatório o processo de criação da grade de horários. Após a análise das necessidades de todos os envolvidos, diversas restrições foram modeladas, considerando também um conjunto de penalidades para as situações indesejadas.

A implementação do modelo e sua execução sobre dados reais do curso propiciam a confirmação de sua eficiência na resolução do problema proposto, validando a sua aplicabilidade. Os resultados solidificam que é possível a geração de uma grade de horários automatizada consistente, reduzindo o tempo de criação e atendendo ao máximo as preferências dos professores.

O modelo flexibilizado proposto neste trabalho se adaptou muito bem aos diversos cenários testados, mostrando pouca variação na qualidade da solução produzida. A única exceção se deu quando foi testada a tentativa de evitar aulas e, principalmente, a proibição de aulas em algum dia da semana, o que é um cenário de ocorrência muito improvável. Em relação à mudança do perfil das turmas, pode-se dizer que a mesma não torna a solução final necessariamente melhor com os dados de entrada que foram utilizados como teste. Portanto, a flexibilização dos perfis das turmas deve ser utilizada com cautela pelo usuário do sistema, pois pode causar uma piora no quadro final em relação à qualidade da solução, embora permita aos professores maior controle sobre a organização de suas aulas.

6.1 – Trabalhos futuros

Existem diversas possibilidades de trabalhos futuros para dar continuidade a este trabalho. Uma delas é a identificação exata do erro que impediu que o otimizador Cplex encontrasse uma solução. Por exemplo, se duas turmas que pertencem ao mesmo grupo possuírem algum horário pré-definido conflitante, não vai ser possível encontrar qualquer solução que satisfaça todas as restrições. O sistema atual apenas informa a ocorrência de erro na geração do quadro, porém não especifica o que ocasionou, e isto facilitaria a correção pelo usuário.

Outro aperfeiçoamento é sobre as situações de compartilhamento de salas de aula e laboratórios, que poderiam ser resolvidas automaticamente. Como exemplo, atualmente o

acordo entre os departamentos é de que apenas metade das turmas do Departamento de Matemática tenha aula nas salas do Departamento de Computação. O sistema proposto não prevê isso e, como trabalho futuro, pode ser permitido especificar o tipo de sala que uma turma deve ter, permitindo assim o compartilhamento dos recursos.

Outro passo poderia ser o desenvolvimento de uma versão web do sistema, para que os professores pudessem fornecer seus dados de forma totalmente descentralizada. Além disso, o sistema foi desenvolvido utilizando como base o curso de Ciência da Computação. Em contrapartida, poderia ser feita uma adaptação do mesmo com o intuito de ampliar o seu uso, possibilitando que os outros cursos do Centro Universitário de Rio das Ostras também possam gerar seus quadros de horários de forma automatizada.

Referências Bibliográficas

- [1] VIEIRA, Felipe; MACEDO, Hendrik. Sistema de alocação de horários de cursos universitários: um estudo de caso no departamento de computação da Universidade Federal de Sergipe. *Scientia Plena*, 2011, 7.3.
- [2] SIMÃO, Thiago Dias. Utilização de algoritmos genéticos para otimização de soluções para o timetabling escolar. Tese apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras, 2013.
- [3] PAIM, Alexandre Da Silva; GREIS, Ivone Chassot. Abordagens para Elaboração Automatizada de Tabela de Horários Acadêmicos. *Artigo (Acadêmicos do curso de Administração da Universidade Luterano do Brasil)*, 2008.
- [4] MURRAY, Keith; MÜLLER, Tomáš; RUDOVÁ, Hana. Modeling and solution of a complex university course timetabling problem. In: *International Conference on the Practice and Theory of Automated Timetabling*. Springer Berlin Heidelberg, 2006. p. 189-209.
- [5] INTERNATIONAL BUSINESS MACHINES, CPLEX Optimizer. Disponível em: <<https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer>>. Acesso em novembro de 2016.
- [6] DE LIMA, Thiago Jéffery Barisão. Colaborário – Sistema Colaborativo para Geração de Quadro de Horários de Curso Universitário. Dissertação apresentada ao Mestrado em Engenharia de Produção e Sistemas Computacionais da Universidade Federal Fluminense, 2016.
- [7] LACH, Gerald; LÜBBECKE, Marco E. Curriculum based course timetabling: new solutions to Udine benchmark instances. *Annals of Operations Research*, 2012, 194.1: 255-272.

- [8] VAN DEN BROEK, John; HURKENS, Cor; WOEGINGER, Gerhard. Timetabling problems at the TU Eindhoven. *European Journal of Operational Research*, 2009, 196.3: 877-885.
- [9] MIRHASSANI, S. A. A computational approach to enhancing course timetabling with integer programming. *Applied Mathematics and Computation*, 2006, 175.1: 814-822.
- [10] MUSHI, Allen R.; MARWA, Yohana. Late Acceptance Heuristic for University Course Timetabling Problem. *International Journal of Advanced Research in Computer Science*, 2013, 4.2.
- [11] HEAD, Christopher; SHABAN, Sami. A heuristic approach to simultaneous course/student timetabling. *Computers & Operations Research*, 2007, 34.4: 919-933.
- [12] LÜ, Zhipeng; HAO, Jin-Kao. Adaptive tabu search for course timetabling. *European Journal of Operational Research*, 2010, 200.1: 235-244.
- [13] GLOVER, Fred; LAGUNA, Manuel. *Tabu Search**. Springer New York, 2013.
- [14] LEWIS, Rhyd; THOMPSON, Jonathan. Analysing the effects of solution space connectivity with an effective metaheuristic for the course timetabling problem. *European Journal of Operational Research*, 2015, 240.3: 637-648.
- [15] CESCHIA, Sara; DI GASPERO, Luca; SCHAERF, Andrea. Design, engineering, and experimental analysis of a simulated annealing approach to the post-enrolment course timetabling problem. *Computers & Operations Research*, 2012, 39.7: 1615-1624.
- [16] ABUHAMDAH, Anmar, et al. Population based Local Search for university course timetabling problems. *Applied intelligence*, 2014, 40.1: 44-53.
- [17] GALL, Ana Flávia Neves; DA SILVA, André Renato Villela. Alocação de Horários de Professores e Turmas em um Curso Universitário. In: *Simpósio Brasileiro de Pesquisa Operacional. Anais do SBPO*, 2016.

[18] BEZERRA, Eduardo. *Princípios De Análise E Projeto De Sistemas Com Uml-3ª Edição*. Elsevier Brasil, 2006.

[19] ITEXT. Disponível em: < <http://itextpdf.com>>. Acesso em novembro de 2016.