

UNIVERSIDADE FEDERAL FLUMINENSE
CENTRO UNIVERSITÁRIO DE RIO DAS OSTRAS
INSTITUTO DE CIÊNCIA E TECNOLOGIA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Ricardo Vilela Machado

Estudo de método para solução do problema de alocação de
horários no curso de Ciência da Computação em Rio das Ostras

Rio das Ostras - RJ

2017

RICARDO VILELA MACHADO

ESTUDO DE MÉTODO PARA SOLUÇÃO DO PROBLEMA DE ALOCAÇÃO DE HORÁRIOS NO
CURSO DE CIÊNCIA DA COMPUTAÇÃO EM RIO DAS OSTRAS

Monografia apresentada ao Curso de Bacharelado em Ciência 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

RICARDO VILELA MACHADO

ESTUDO DE MÉTODO PARA SOLUÇÃO DO PROBLEMA DE ALOCAÇÃO DE HORÁRIOS NO
CURSO DE CIÊNCIA DA COMPUTAÇÃO EM RIO DAS OSTRAS

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

Aprovada em Janeiro de 2017.

BANCA EXAMINADORA

Prof. Dr. ANDRÉ RENATO VILLELA DA SILVA - Orientador
UNIVERSIDADE FEDERAL FLUMINENSE

Prof.^a Dr.^a FLÁVIA CRISTINA BERNARDINI
UNIVERSIDADE FEDERAL FLUMINENSE

Prof. Dr. ALESSANDRO COPETTI
UNIVERSIDADE FEDERAL FLUMINENSE

Rio das Ostras-RJ

2017

Ao meu avô, Domingos Alves Machado, meu melhor amigo! E quem sempre me incentivou e me deu forças para realização dos meus sonhos. Com muito carinho, dedico este trabalho a ele.

Agradecimentos

Agradeço primeiramente a Deus por ter me ajudado na realização desse sonho, pois sem Ele não teria conseguido.

Também agradeço a minha família por estar ao meu lado em todos os momentos, sempre me ajudando, me dando forças e me incentivando. Em especial agradeço meu avô e amigo Domingos Machado. Muito obrigado mesmo por todo carinho e amor.

Agradecimento especial para a minha namorada Lais Leal, por todo amor, carinho, ajuda, incentivo e paciência em todos os momentos.

Agradeço imensamente ao meu orientador André Renato por todo ensinamento, dedicação e paciência em todos os momentos. Sempre disposto em ajudar no que foi preciso. O admiro muito.

Agradeço a esta Universidade e aos professores que contribuíram na minha formação.

Muito obrigado mesmo!

Lista de Figuras

3.1	Diagrama de Casos de Uso	10
3.2	Diagrama Entidade Relacionamento(ER)	12
3.3	Diagrama de classe de análise	22
3.4	Diagrama de Classe de Projeto	24
3.5	Exemplo de buracos nos horários	27
3.6	Tela login	28
3.7	Tela de Cadastrar departamento	28
3.8	Tela de Cadastrar professor	29
3.9	Tela de Cadastrar turma	29
3.10	Tela de Atualizar turma	30
3.11	Tela de Cadastrar grupo	30
3.12	Tela de Cadastrar penalidades de alocar turmas do grupo	31
3.13	Tela de Cadastrar horários censurados do grupo	31
3.14	Tela de Cadastrar preferência penalidades	32
3.15	Tela de Cadastrar preferência horários	32
3.16	Tela de Consultar solução	33
3.17	Tela de Listar soluções	33
4.1	Custos das soluções - Cenário 1	34
4.2	Custos das soluções - Cenário 2	35
4.3	Comparações dos custos - Cenário 3	36
4.4	Ilustração da solução exportada	39

Lista de Tabelas

3.1	Tabela usuario	13
3.2	Tabela departamento	13
3.3	Tabela departamento_proibidos	14
3.4	Tabela professor	14
3.5	Tabela professor_penalidadehorario	14
3.6	Tabela professor_penalidadesburacodias	15
3.7	Tabela professor_penalidadesburacosemanas	15
3.8	Tabela grupo	15
3.9	Tabela grupo_penalidadeburacosdias	16
3.10	Tabela grupo_censurados	16
3.11	Tabela grupo_penalidadehorario	16
3.12	Tabela turma	17
3.13	Tabela turma_obrigatorios	17
3.14	Tabela solucao	17
3.15	Tabela solucao_horarios	18
4.1	Resultados	36
4.2	Penalidades contabilizadas nos cenários	37

Sumário

Agradecimentos	v
Lista de Figuras	vi
Lista de Tabelas	vii
Resumo	x
Abstract	xi
1 Introdução	1
1.1 Motivação e Contribuição	1
1.2 Objetivos	1
1.2.1 Objetivo Principal	1
1.2.2 Objetivos Secundários	2
1.3 Organização do Trabalho	2
2 Problema de Alocação de Horários	3
2.1 Trabalhos Relacionados	4
2.1.1 Algoritmos Genéticos	4
2.1.2 VNS/VND	5
2.1.3 Divisão-E-Conquista	5
3 Metodologia de Desenvolvimento	6
3.1 Listagem de Peculiaridades	6
3.1.1 Ponto de Vista do Aluno	6
3.1.2 Ponto de Vista do Professor	7
3.1.3 Ponto de Vista do Departamento	7
3.2 Modelagem do Sistema	7
3.2.1 Requisitos de Sistema - Casos de Uso	7
3.2.2 Modelo Entidade Relacionamento	11
3.2.3 Diagrama de Classe de Análise	18
3.2.4 Diagrama Classe de Projeto	23

3.3	Solução do Problema	25
3.3.1	Alocação dos Horários Pré-Determinados	25
3.3.2	Alocação dos Horários não Pré-Determinados	26
3.4	Interfaces do Sistema	28
3.4.1	Login	28
3.4.2	Gerenciar Departamento	28
3.4.3	Gerenciar Professor	29
3.4.4	Gerenciar Turma	29
3.4.5	Gerenciar Grupo	30
3.4.6	Gerenciar Preferências Professor	32
3.4.7	Gerenciar Solucao	33
4	Resultados Obtidos	34
4.1	Cenário 1	34
4.2	Cenário 2	35
4.3	Cenário 3	36
4.4	Análise dos Resultados	36
4.4.1	Horários indesejados	37
5	Conclusão	40
6	Trabalhos Futuros	41
	Referências Bibliográficas	42
7	Apêndice	44
7.1	Descrição dos casos de uso	44

Resumo

Este trabalho de conclusão de curso refere-se ao Problema de Alocação de Horários (*Timetabling Problem*) em cursos universitários. Esse problema de Otimização Combinatória consiste em definir os horários das disciplinas que compõem um curso levando em consideração as restrições impostas, como por exemplo, disponibilidade de horários de professores, aulas alocadas sem "janelas", evitar conflitos de horários e entre outras. O estudo do método para solucionar o Problema foi feito pelo fato do Departamento de Computação de Rio das Ostras realizar todo o trabalho de alocação de forma manual, levando dias para se ter uma solução. Dessa forma, a heurística desenvolvida leva em consideração todas as restrições impostas. Também possibilita que o coordenador do curso defina a quantidade de soluções para serem geradas em um curto intervalo de tempo, sendo escolhida a melhor e mais viável para o campus. A metodologia consistiu inicialmente em fazer a alocação dos horários que são pré-determinados, ou seja, que não podem de forma alguma ser alterados. Em seguida, foram alocados os horários que não são pré-determinados. O custo dessa alocação deve ser o menor possível dentre todos os horários disponíveis. Para os diferentes cenários criados, a heurística conseguiu se adaptar bem ao aumento das penalidades e retornou soluções satisfatórias com desvio padrão relativamente baixo, além de uma redução de aproximadamente 80% nos horários ruins em um dos cenários de testes criados.

Palavras-chave: Otimização Combinatória, Problema de Alocação de Horários, Heurísticas Construtivas.

Abstract

This conclusion work refers to the Timetabling Problem on university courses. This Combinatorial Optimization problem consists in defining the time of the classes which compose a course, considering the restrictions like professor's time availability, classes allocated back-to-back, avoiding time conflicts, and others. The study of a method to solve the Problem was made due to the fact that the Computing Department in Rio das Ostras makes all the allocation work manually, taking days to have a solution. In this way, the developed heuristic considers all imposed restrictions. It also enables the course coordinator to define the quantity of solutions to be generated in a short period of time, being chosen the best and most feasible solution for the campus. The methodology initially consists of making the allocation of times that are predetermined, et cetera, that can not be altered in any way. Next, the non-predetermined times are allocated. The cost of this allocation must be the least among all the available times. For the different scenarios created, the heuristic managed to adapt well to the increase of penalties and returned satisfactory solutions with a standard deviation relatively low, as well as a reduction of approximately 80% in the bad times in one of the test scenarios created.

Keywords: Combinatorial Optimization, Timetabling Problem, Constructive Heuristics.

Capítulo 1

Introdução

1.1 Motivação e Contribuição

A alocação da grade de horários em cursos universitários é um trabalho contínuo, repetitivo e de difícil solução. Trata-se do problema combinatório de se distribuir as disciplinas ao longo dos vários horários e dias da semana, de forma a se respeitar as mais variadas restrições impostas, como disponibilidade de professores, de salas de aula e não existir conflitos de horários, por exemplo.

Resolver esse problema de otimização combinatória de forma manual é complexo, uma vez que devem ser levadas em consideração várias restrições, e na maioria das vezes leva-se muito tempo para se ter uma solução. Dependendo do caso, essa solução pode não ser satisfatória, pois alguma restrição pode deixar de ser atendida.

Atualmente, no Departamento de Computação (RCM) da Universidade Federal Fluminense do campus de Rio das Ostras (CURO/UFF), todo o trabalho é feito de forma manual pelo coordenador do curso de Ciência da Computação ao longo de vários dias. A cada final de semestre, as turmas são divididas entre os professores do departamento e inicia-se o processo de alocação de horários. Normalmente, quando um quadro de horários fica pronto, os professores fazem diversas sugestões de melhorias ou adequações aos seus projetos didáticos, o que implica na adaptação do quadro ou mesmo na sua completa reconstrução. Várias inconsistências precisam ser resolvidas a cada nova versão, além das tentativas de proporcionar aos alunos a maior quantidade de disciplinas sem conflitos de horários. Dessa forma, esse estudo tem como finalidade fornecer uma solução para o problema de alocação de horários em cursos universitários, como aqueles oferecidos pela UFF.

1.2 Objetivos

1.2.1 Objetivo Principal

O objetivo principal deste trabalho é desenvolver uma heurística capaz de solucionar o Problema de Alocação de Horários em Cursos Universitários com qualidade em um curto intervalo de tempo.

1.2.2 Objetivos Secundários

- Desenvolvimento de um sistema que permita a comparação e a análise de um pequeno conjunto de soluções distintas;
- Permitir modificações pontuais em uma solução gerada, de forma a guiar o processo de refinamento desta solução.

1.3 Organização do Trabalho

A estrutura do trabalho está organizada da seguinte forma: o capítulo 2 apresenta a definição e as características do Problema de Alocação de Horários. Também são apresentados os trabalhos relacionados ao tema na seção 2.1, destacando as heurísticas adotadas e os resultados obtidos.

O capítulo 3 apresenta a metodologia adotada para o desenvolvimento da solução. Na seção 3.1, é apresentada uma listagem de peculiaridades presentes no campus, sendo os requisitos do sistema, que levam em consideração os pontos de vista do Professor, do Aluno e do Departamento. Na seção 3.2, tem-se a modelagem do sistema, incluindo o diagrama de casos de uso e suas descrições na seção 3.2.1; a modelagem conceitual do sistema com o Modelo Entidade Relacionamento na seção 3.2.2; na seção 3.2.3 tem-se o Diagrama de Classe de Análise ; na seção 3.2.4 o Diagrama de Classe de Projeto, sendo mapeadas as classes e seus relacionamentos. A seção 3.3 possui a descrição de como a solução do problema foi desenvolvida; na seção 3.4 tem-se a representação das principais interfaces do sistema.

No capítulo 4, são apresentados os resultados obtidos com a solução desenvolvida, mostrando o comportamento da heurística para os cenários criados.

Por fim, nos capítulos 5 e 6 são apresentados a conclusão e os trabalhos futuros.

Capítulo 2

Problema de Alocação de Horários

Como definido por [1], o Problema de Alocação de Horários, também conhecido como *Timetabling Problem*, é uma classe de problemas combinatórios que consistem em construir quadros de horários para uma série de atividades atendendo a um determinado conjunto de restrições.

Segundo [10], o Problema é classificado como NP-Difícil, que é quando uma solução ótima é difícil de ser encontrada em um tempo computacional aceitável. De acordo com [11], para definir o Problema, deve-se levar em consideração quatro parâmetros, que são: 1) um conjunto finito de horários; 2) um conjunto de restrições; 3) um conjunto finito de recursos; 4) um conjunto de eventos. Dessa forma, os parâmetros 1) e 3) são associados ao 4) com a finalidade de atender a maior quantidade possível de 2).

Como dito em [12], o conjunto de restrições é específico de cada instituição envolvida. Dependendo das restrições impostas pode necessitar de disponibilidade de recursos, preferências de cada envolvido, janelas ou conflitos de horários ou outras regras. É que a quantidade e a combinação dessas restrições torna o Problema mais complexo, uma vez que mais condições precisam ser tratadas. Esse Problema possui diversas áreas de aplicação. Dentre as quais pode-se citar: transporte [14], esporte [13], escala de trabalhador [15] e educacional [7].

Para a área educacional, em [2], o problema foi classificado em três categorias: Quadro de Horários para Escolas (*School Timetabling*), Quadro de Horários para Universidades (*University Timetabling*) e Quadro de Horários para Exames (*Examination Timetabling*). Porém, esse Problema não é estritamente classificado em apenas uma das três categorias anteriores, sendo em alguns casos uma combinação delas. Diante disso, [16] identificou cinco subclassificações para o Problema de Alocação de Horários: Horários de cursos, Alocação de professores nas turmas, Atribuição de salas de aula, Atribuição de professores e Programação de horários de estudantes.

Esse trabalho de conclusão de curso refere-se à subclassificação *Horários de cursos*, a qual trata do Problema de alocação dos horários das turmas de um curso num período semanal levando em consideração as restrições impostas. Estas podem ser de diferentes categorias, como definido por [17]. A heurística desenvolvida para esse sistema considerou as restrições *hard* e *soft constraint*. Quando uma restrição não pode deixar de ser cumprida, ela é do tipo *hard*, como por exemplo um mesmo professor lecionar duas turmas diferentes no mesmo horário. Restrições do tipo *soft* são aquelas que preferivelmente devem ser

atendidas. Como por exemplo, que não exista janelas de horários. O algoritmo desenvolvido aborda o conceito de método construtivo, ou seja, para gerar o quadro de horários, em cada passo o melhor horário (menor penalidade) é adicionado ao quadro final. Este processo é feito até que todos os horários sejam inseridos.

2.1 Trabalhos Relacionados

Neste semestre (2016.2), foi utilizada uma ferramenta experimental chamada Colaborário. A idéia principal desse sistema é permitir que a elaboração do quadro de horários possa feita de forma colaborada. O aluno indica o plano de estudos com as disciplinas que pretende cursar, em seguida o sistema tentará maximizar a geração do quadro de horários para que o aluno consiga se inscrever na maior quantidade de turmas. Depois do quadro de horários ter sido gerado, o aluno e o professor podem sugerir e opinar mudanças dos horários. A seguir, são apresentados alguns trabalhos relacionados com diferentes técnicas (meta-heurísticas). Dentre as quais, tem-se os Algoritmos Genéticos, VNS/VND e Divisão-E-Conquista.

2.1.1 Algoritmos Genéticos

Em [5], foi desenvolvida uma solução para o problema de alocação de grades de horários para instituições de ensino superior através de Algoritmos Genéticos. As restrições e preferências envolvidas na heurística procedem da Faculdade de Engenharia Elétrica da UFU - Universidade Federal de Uberlândia - MG. Possui como foco reduzir os conflitos, potencializar as preferências e compactar os horários dos professores e alunos. Para isso, foi feita uma avaliação da solução gerada contabilizando todas as violações e restrições impostas, sendo realizada a verificação de quanto a solução está em conformidade com a disponibilidade do professor.

Em [8], foram desenvolvidas combinações de heurísticas para solucionar o problema de construção de grades horárias escolares. O trabalho levou em consideração as Metaheurísticas Evolutivas (Algoritmos Genéticos) e de Busca Local (Busca Tabu, Grasp) além de Path Relinking. Foram desenvolvidas 5 metaheurísticas (Busca Tabu, Busca Tabu + Path Relinking, Grasp, Grasp + Path Relinking e Algoritmos Genéticos) e ao final uma comparação entre qual obteve melhor resultado foi feita. Para que essas soluções fossem avaliadas, foram definidas 5 classes que calculam a quantidade de violações que a solução teve e também foi calculado o tempo médio da execução. Para esse caso específico, as metaheurísticas que tiveram melhores resultados e menor tempo de execução foram as que usaram Busca Local. Sendo que a Busca Tabu obteve soluções de boa qualidade em 90% dos casos.

Em [6], é feito um estudo de caso no Departamento de Computação da Universidade Federal de Sergipe para o problema de alocação de horários. É proposta uma solução baseada em Algoritmos Genéticos que visa melhorar e agilizar o processo de geração de quadro de horários de forma manual. A metodologia é formulada em uma busca local em espaço de estados e segue a analogia da seleção natural e evolução para que tenha um melhor resultado. Para este trabalho, foram levados em consideração dois tipos de restrições, *hard* e *soft*. Quando uma determinada restrição é atendida na solução, uma

bonificação é feita e quando não é atendida uma penalidade é gerada. Com isso, a partir da melhor solução gerada através do Algoritmo Genético é feita uma análise para verificar se atende as restrições estabelecidas.

2.1.2 VNS/VND

A proposta de [7] é sobre uma meta-heurística VNS/VND em Java para o problema de alocação de horários aos cursos de Sistema de informação e Análise e Desenvolvimento de Sistemas de um centro Universitário do Espírito Santo. O objetivo foi mostrar que o método desenvolvido por uma solução computacional obteve uma solução com melhor qualidade e um intervalo de tempo bem menor que o desenvolvimento de forma manual. Foram utilizados 2 cursos com 7 turmas, 36 disciplinas e 20 professores como entradas para o trabalho. A heurística buscou maximizar as preferências de cada docente por um determinado dia e hora para as turmas que lecionará. Para se comparar a solução gerada com a forma manual, foram feitas pontuações sobre as preferências de horários de professores e o agrupamento de horários. No método manual, levou-se 20 horas para geração do quadro de horários e teve uma pontuação de 671. Já a solução computacional, o tempo foi de 30 minutos para entrada dos dados, sendo que 9 minutos para execução do algoritmo e teve como pontuação um total de 714. O computador utilizado para teste foi um Dell OPTIPLEX 790, processador Intel Core i5 e memória RAM de 8GB.

2.1.3 Divisão-E-Conquista

Uma proposta baseada na estratégia de divisão-e-conquista foi desenvolvida por [9] para resolver o problema de alocação de forma eficiente, referente ao sistema educacional brasileiro, onde foram utilizados dados reais de sete escolas. A ideia do algoritmo é fazer a divisão do problema inicial em outros problemas menores e solucioná-los. Segundo o autor, quando o quadro de horários é feito de forma manual, intuitivamente é feito através de divisão-e-conquista. O foco do trabalho consistiu em atender as principais preocupações dos gestores, que são as aulas duplas serem atendidas. Mas as exigências principais eram diminuir o número de dias de trabalho e redução da quantidade de janelas nos horários dos professores. Para que a solução encontrada fosse classificada como viável, foi necessário que esta atendesse as exigências das escolas, para que nenhuma das restrições fossem violadas. O algoritmo mostrou-se eficiente para instâncias médias e grandes.

Capítulo 3

Metodologia de Desenvolvimento

3.1 Listagem de Peculiaridades

Para fazer o levantamento de requisitos do sistema, foi feito um questionário usando o formulário do Google para que assim os alunos do campus pudessem expor como eles imaginam um quadro de horários ideal. A partir das respostas obtidas neste formulário, foram definidas restrições que deveriam ser atendidas pelo algoritmo do sistema ao gerar a solução para o problema de alocação de horários, levando em consideração o ponto de vista dos principais interessados, que são o aluno, o professor e o departamento. A seguir, temos os requisitos.

3.1.1 Ponto de Vista do Aluno

Do ponto de vista dos alunos, temos as seguintes restrições levantadas:

- Preferencialmente as aulas devem começar a partir das 09:00 hs;
- Preferencialmente as aulas devem terminar às 18:00 hs;
- Os horários de disciplinas do mesmo período não devem ter muita diferença de horário;
- Horário de almoço do aluno deve ser respeitado;
- Preferencialmente as aulas sejam em sequência, sem 'janelas' de horários;
- Como o campus de Rio das Ostras possui muitos alunos de outras cidades, seria proveitoso que os horários não fossem na segunda cedo e na sexta à tarde;
- Horários preferenciais ao longo da semana. Por exemplo: horário de estágio para turmas de sexto período em diante;
- Tentar evitar conflitos entre disciplinas de períodos próximos.

3.1.2 Ponto de Vista do Professor

Do ponto de vista dos professores, as restrições e preferências foram as seguintes:

- Um professor não pode estar alocado em duas disciplinas no mesmo horário;
- Horário de almoço do professor deve ser respeitado;
- Cada professor deve ter o menor número possível de janelas;
- Professores que moram em outras cidades, buscar alocar as aulas em dias consecutivos;
- Um professor não deve dar 6 horas de aula em um dia.

3.1.3 Ponto de Vista do Departamento

Para o departamento, as restrições definidas foram:

- Disciplinas do mesmo período não podem ter conflito de horário;
- Não pode haver aula no horário da reunião departamental;
- A quantidade de salas ocupadas de um horário não pode ser maior que o total de salas disponíveis;
- As aulas semanais de cada disciplina devem ser atendidas.

3.2 Modelagem do Sistema

Uma etapa muito importante no desenvolvimento de um sistema é fazer a modelagem, para que o seu comportamento seja especificado e facilite a compreensão do mesmo como um todo. Para a realização da etapa de modelagem, a linguagem definida foi a *UML* - Linguagem de Modelagem Unificada (*Unified Modeling Language*), que como dito em [3], é uma linguagem visual para modelar sistemas orientados a objetos. Isso quer dizer que a *UML* é uma linguagem constituída de elementos gráficos utilizados na modelagem que permitem representar os conceitos do paradigma da orientação a objetos. Através dos elementos gráficos definidos nesta linguagem pode-se construir diagramas que representam diversas perspectivas do sistema.

Em [3], é descrita uma vantagem da *UML*, que é independente tanto de linguagens de programação quanto de processos de desenvolvimento. Isso quer dizer que a *UML* pode ser utilizada para a modelagem de sistemas, não importa qual a linguagem de programação será utilizada na implementação, ou qual a forma (processo) de desenvolvimento adotada.

3.2.1 Requisitos de Sistema - Casos de Uso

Como definido em [3], o modelo de caso de uso é uma representação das funcionalidades externamente observáveis do sistema e dos elementos externos ao sistema que interagem com

ele. Esse modelo está relacionado com a especificação dos requisitos do sistema, mais especificamente os requisitos funcionais. Sendo que para fazer a modelagem dos casos de uso, o diagrama adotado da *UML* é o Diagrama de Casos de Uso [3].

Como dito em [3], por ter uma notação gráfica simples e usar uma linguagem natural para descrever requisitos, a *UML* tem sido cada vez mais utilizada para representar requisitos funcionais de um sistema, uma vez que torna a comunicação entre as pessoas que estão desenvolvendo e as que vão usar mais clara, simples e padronizada. A modelagem de casos de uso de uma aplicação consiste dos casos de uso, dos atores envolvidos e dos relacionamentos que estes possuem.

- Segundo [3], caso de uso é a especificação de uma sequência de interações entre um sistema e os agentes externos que utilizam esse sistema. Um caso de uso deve definir o uso de uma parte da funcionalidade de um sistema, sem revelar a estrutura e o comportamento internos desse sistema.
- Ainda segundo [3], qualquer elemento externo que interage com o sistema é denominado ator. O termo externo significa que um ator troca (envia e/ou recebe) informações com o sistema.
- O relacionamento é a forma como os casos de uso estão associados com outros casos de uso ou atores.

Diagrama de Casos de Uso

No diagrama 3.1, existem 2 perfis de atores presentes no diagrama de casos de uso, que são os responsáveis por executar as ações.

1. **Coordenador do curso:** Principal ator do sistema que tem como função o gerenciamento (Cadastrar, Atualizar e Remover) dos dados no sistema referentes aos professores, grupos, turmas e departamentos. Além de ser o responsável por gerar e fazer a exportação do quadro de horários.
2. **Professor:** Ator que tem como função gerenciar (Cadastrar e Atualizar) as suas informações de acordo com suas preferências para serem levadas em consideração na criação da solução.

O diagrama de Casos de Uso está estruturado em pacotes, onde este possui um conjunto de casos de uso que estão relacionados, organizando assim as unidades do sistema.

Temos os seguintes pacotes no diagrama:

- **Gerenciar Professor:** Nesse pacote estão os casos de uso relacionados com o professor. O coordenador do curso pode cadastrar, remover e listar os dados do professor no sistema.
- **Gerenciar Turma:** Possui os casos de uso relacionados com a turma. O coordenador do curso pode cadastrar, remover, atualizar e listar os dados da turma no sistema.

- **Gerenciar Grupo:** Inclui os casos de uso referente ao grupo. O coordenador do curso pode cadastrar, atualizar, remover e listar as informações de um grupo no sistema.
- **Gerenciar Departamento:** Possui os casos de uso relacionados ao departamento. O coordenador do curso pode cadastrar , atualizar, remover e listar os dados de um departamento.
- **Gerenciar Preferencias:** Inclui os casos de uso relacionados com as preferências de um professor. Este pode cadastrar e atualizar as suas preferências.
- **Gerenciar Coordenador:** Nesse pacote estão os casos de uso relacionados com o coordenador do curso. Este pode cadastrar, remover e atualizar a sua senha.
- **Gerenciar Solucao:** Possui os casos de uso relacionados com a Solução do sistema. O coordenador do curso pode gerar, consultar, listar, remover e exportar uma solução do sistema.

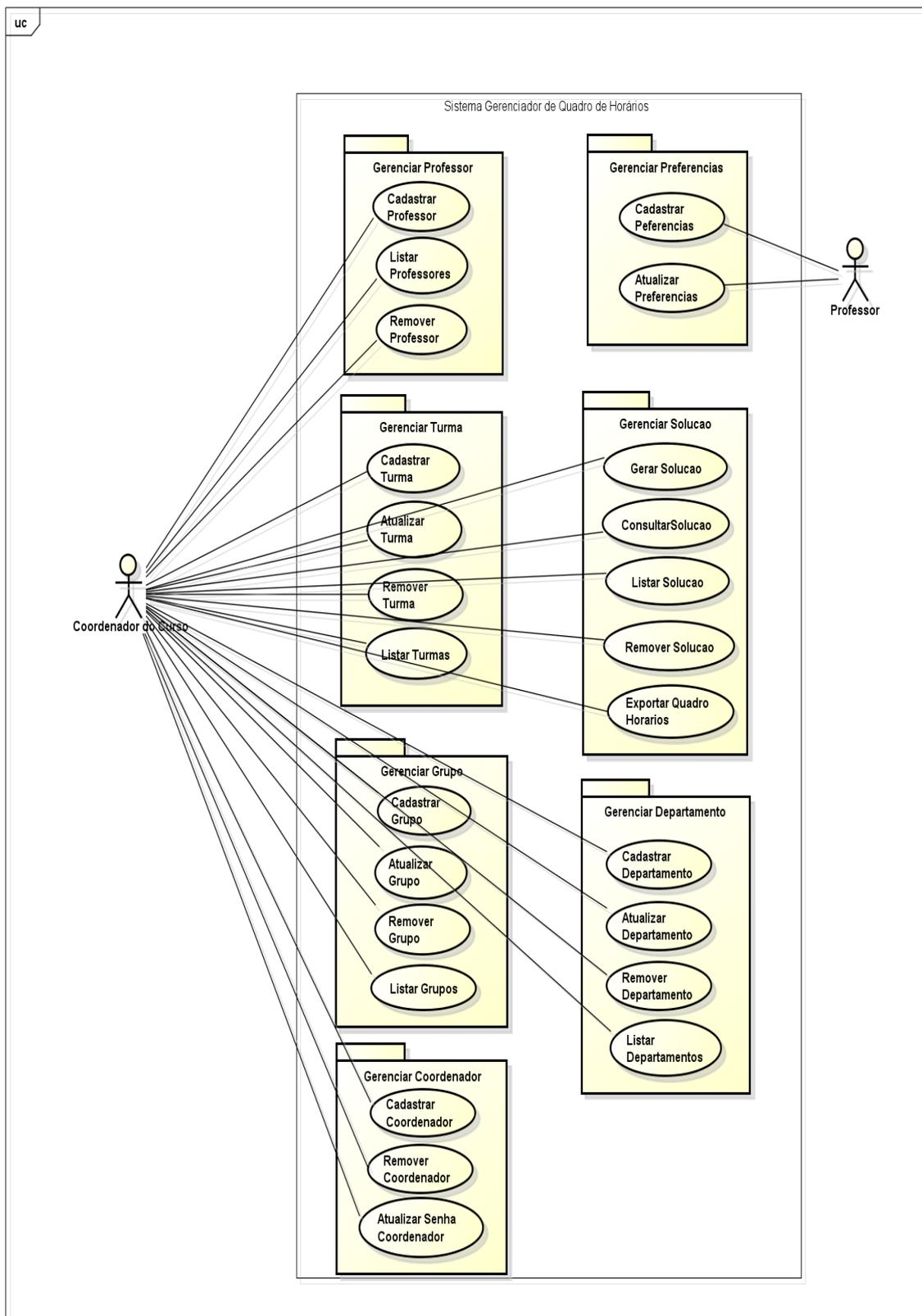


Figura 3.1: Diagrama de Casos de Uso

Descrição dos Casos de Uso

O diagrama anterior 3.1 é uma forma de representar graficamente os casos de uso do sistema, temos ainda como descrever as interações de cada um desses casos de uso de forma textual. As descrições dos casos de uso presentes no sistema encontram-se no apêndice.

3.2.2 Modelo Entidade Relacionamento

Para criar a modelagem conceitual do banco de dados do sistema, foi utilizado o diagrama de Modelo Entidade Relacionamento (ER). De acordo com [4], esse diagrama é caracterizado como um modelo de dados conceitual popular de alto nível, que descreve os dados como entidades, relacionamentos e atributos. A seguir é descrito o mini-mundo contendo as informações persistidas no banco de dados do sistema.

- A faculdade é organizada em departamentos. Cada departamento possui nome exclusivo, um número exclusivo, a quantidade de salas e uma lista de horários (dias, horas) em que as turmas desse departamento não podem ter aula.
- Um departamento possui turmas associadas. Essas turmas possuem um código exclusivo (código da turma e código da Disciplina), nome, carga horária, período, máximo de aulas por dia, horários obrigatórios.
- Cada turma, tem no máximo um professor. Este possui um número único, nome, as preferências de quantidade de dias por semana que deseja ministrar aula, penalidade de existir um excesso de aulas por dia, penalidades dos buracos referentes as aulas em um dia e as penalidades dos buracos referentes as aulas na semana.
- Para agrupar as turmas, temos os grupos (Agrupamento das turmas de um período, com preferências de horários similares). Estes possuem um número exclusivo, nome, máximo de aulas por dia, penalidade de existir um excesso de aulas num determinado dia, horários censurados e as penalidades de alocar uma turma num horário.
- Uma Solucao possui um número único, nome, custo total e os horários gerados das turmas.

O esquema do banco de dados do sistema é representado na Figura 3.2, através do Diagrama Entidade Relacionamento.

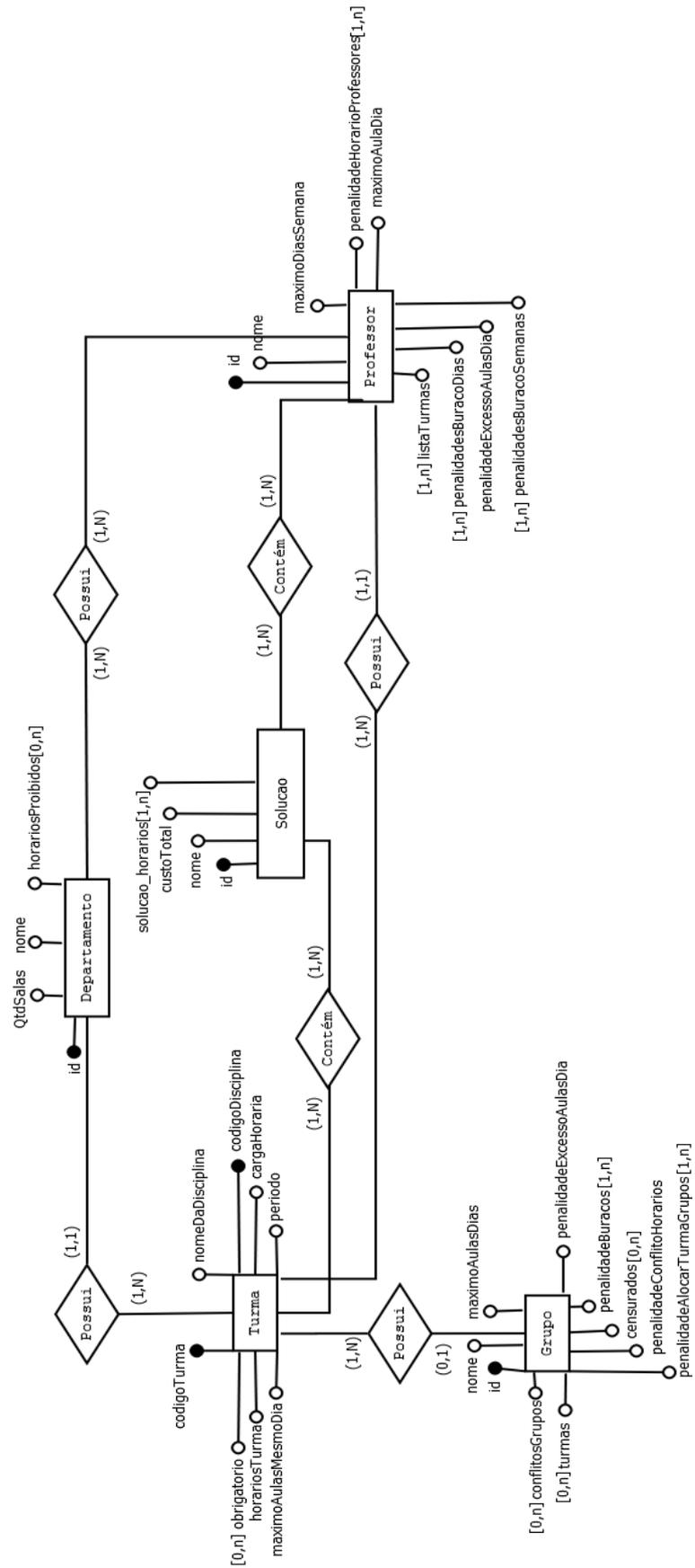


Figura 3.2: Diagrama Entidade Relacionamento(ER)

Com base no diagrama apresentado na Figura 3.2, foram criadas as seguintes tabelas no banco de dados para persistir as informações do sistema.

- A Tabela 3.1 persiste os dados dos usuários (professor e coordenador) cadastrados no sistema. Possui como identificador um número, nome, login, senha e o tipo do usuário (se é ou não coordenador). Quando um professor é criado no sistema pelo coordenador, esse professor é cadastrado na Tabela **usuario**. Assim que o professor realizar o cadastro das preferências ele será também cadastrado na Tabela **professor** 3.4 com o mesmo identificador da Tabela **usuario**. Foi necessário criar a Tabela **usuario** pelo fato do sistema possuir dois perfis de atores com funções e acessos aos dados restritos por perfil.

usuario	
Campo	Tipo
id	int
login	varchar
senha	varchar
nome	varchar
coordenador	boolean

Tabela 3.1: Tabela usuario

A Tabela 3.2 armazena os dados referentes a um departamento na base de dados. Cada departamento é identificado por um número, nome e a quantidade de salas disponíveis.

departamento	
Campo	Tipo
id	int
qtdSalas	int
nome	varchar

Tabela 3.2: Tabela departamento

Cada departamento pode possuir horários em que não seja possível uma turma ser alocada. Esses horários são classificados como proibidos (reunião de departamento, colegiados, etc). Com isso, a Tabela 3.3 foi criada para armazenar esses horários. Possui um número que identifica o departamento.

departamento_proibidos	
Campo	Tipo
id	int
hora	int
dia	int

Tabela 3.3: Tabela departamento_proibidos

A Tabela 3.4 armazena os dados das preferências referentes aos professores cadastrados no sistema.

professor	
Campo	Tipo
id	int
nome	varchar
maximoDiasSemana	int
maximoAulasDias	int
penalidadesExcessoAulasDia	int
penalidadesExcessoAulasSemana	int

Tabela 3.4: Tabela professor

Na Tabela 3.5 são persistidas as preferências de cada professor referentes aos horários de alocação das turmas. Para $0 \leq \text{penalidade} \leq 10$. Possui um número que identifica o professor.

professor_penalidadehorario	
Campo	Tipo
id	int
hora	int
dia	int
penalidade	int

Tabela 3.5: Tabela professor_penalidadehorario

A Tabela 3.6 foi criada para armazenar as preferências dos professores quanto às penalidades de buracos durante o dia na grade. Para $0 \leq \text{penalidade} \leq 10$ e $1 \leq \text{tamanhoBuraco} \leq 5$.

professor_penalidadesburacodias	
Campo	Tipo
id	int
tamanhoBuraco	int
penalidade	int

Tabela 3.6: Tabela professor_penalidadesburacodias

A Tabela 3.7 foi criada para armazenar as preferências dos professores quanto às penalidades de buracos durante a semana na grade. Para $0 \leq \text{penalidade} \leq 10$ e $1 \leq \text{tamanhoBuraco} \leq 3$.

professor_penalidadesburacosemanas	
Campo	Tipo
id	int
tamanhoBuraco	int
penalidade	int

Tabela 3.7: Tabela professor_penalidadesburacosemanas

Para que os dados referentes aos grupos fossem armazenados, foi criada a Tabela 3.8, que possui um número para identificar cada grupo.

grupo	
Campo	Tipo
id	int
nome	varchar
maximoAulasDias	int
penalidadesExcessoAulasDia	int
penalidadesConflitosHorarios	int

Tabela 3.8: Tabela grupo

A Tabela 3.9 foi criada para armazenar as penalidades dos buracos no dia para as turmas pertencentes ao grupo. Para $0 \leq \text{penalidade} \leq 10$ e $1 \leq \text{tamanhoBuraco} \leq 5$.

grupo_penalidadeburacosdias	
Campo	Tipo
id	int
tamanhoBuraco	int
penalidade	int

Tabela 3.9: Tabela grupo_penalidadeburacosdias

Cada grupo pode possuir horários em que não seja possível uma turma ser alocada. Esses horários são classificados como censurados. Com isso, a Tabela 3.10 foi criada para armazenar esses horários. Possui um número que identifica o grupo.

grupo_censurados	
Campo	Tipo
id	int
hora	int
dia	int

Tabela 3.10: Tabela grupo_censurados

Na Tabela 3.11 são persistidas as penalidades de cada grupo referentes aos horários de alocação das turmas. Para $0 \leq \text{penalidade} \leq 10$. Possui um número que identifica o grupo.

grupo_penalidadehorario	
Campo	Tipo
id	int
hora	int
dia	int
penalidade	int

Tabela 3.11: Tabela grupo_penalidadehorario

Os dados referentes as turmas são armazenados na Tabela 3.12. Cada turma é identificada pelo `codigoTurma` e o `codigoDisciplina`.

turma	
Campo	Tipo
codigoTurma	varchar
codigoDisciplina	varchar
nomeDaDisciplina	varchar
periodo	int
cargaHoraria	int
maximoAulasMesmoDia	int
professor_id	int
grupo_id	int
departamento_id	int

Tabela 3.12: Tabela turma

Algumas turmas possuem horários pré-determinados, principalmente quando são de outro departamento. Para armazenar esses horários, foi criada a Tabela 3.13.

turma_obrigatorios	
Campo	Tipo
codigoTurma	varchar
codigoDisciplina	varchar
hora	int
dia	int

Tabela 3.13: Tabela turma_obrigatorios

Para que os dados das Soluções geradas pudessem ser armazenados, foi criada a Tabela 3.14.

solucao	
Campo	Tipo
id	int
custoTotal	int
nome	varchar

Tabela 3.14: Tabela solucao

Cada solução possui um conjunto de horários para as turmas. Para armazenar esses horários foi criada a Tabela 3.15 . Possui os identificadores da turma e da solução a que se refere.

solucao_horarios	
Campo	Tipo
id	int
codigoDisciplina	int
codigoTurma	varchar
hora	int
dia	int

Tabela 3.15: Tabela solucao_horarios

3.2.3 Diagrama de Classe de Análise

O Diagrama de Classe de Análise faz a representação das classes que envolvem o domínio da aplicação. Como dito em [3], é um diagrama que para solucionar um problema, e não considera as tecnologias que serão utilizadas. Mas são importantes para descrever como as classes vão estar estruturadas e a relação que vão ter entre si. No Diagrama da Figura 3.3, temos a forma visual de como as classes do domínio estão representadas e os seus relacionamentos.

Classes

As classes representam os objetos de um domínio que possuem características e comportamentos semelhantes. Nelas, são declarados os métodos e os atributos referentes a cada objeto. Os métodos são os comportamentos e os atributos são as propriedades que um determinado objeto possui.

As classes presentes no sistema são:

- **Horario**
 - Atributos
 - * dia
 - * hora
 - Métodos
 - * Esta classe possui apenas os métodos básicos de recuperar e definir um valor para os atributos (GET e SET), com isso, não precisam ser definidos no diagrama da classe.
- **Departamento**
 - Atributos
 - * identificador
 - * nome
 - * QuantidadeSalas
 - * proibidos
 - Métodos

- * verificaAulaDiaHora

- Professor

- Atributos

- * identificador
 - * nome
 - * MaximoDiasSemana
 - * MaximoAulaDia
 - * PenalidadeExcessoAulasDia
 - * PenalidadeExcessoDiasSemana
 - * PenalidadesBuracosSemanas
 - * PenalidadeBuracoDias
 - * PenalidadeHorarioProfessores

- Métodos

- * insereMatriz

- Grupo

- Atributos

- * identificador
 - * nome
 - * MaximoAulasDia
 - * penalidadeExcessoAulasDias
 - * penalidadeConflitoHorarios
 - * censurados
 - * penalidadeBuracos
 - * PenalidadeAlocarTurmaGrupo

- Métodos

- * verificarHorariosCensurados
 - * verificarTurmaDia
 - * calculaBuracoMatriz
 - * insereMatriz

- Turma

- Atributos

- * NomeDaDisciplina
 - * codigoDisciplina

- * codigoTurma
- * periodo
- * cargaHoraria
- * maximoAulasMesmoDia
- * obrigatorios
- * grupo
- * professor
- * departamento

- Solução

- Atributos

- * quadroProfessores
 - * quadroGrupos
 - * aulas
 - * valor
 - * horarios

- Métodos

- * insereHorariosNaoObrigatorios
 - * CalculaTotalPenalidadesBuracoDia
 - * CalculaTotalPenalidadesBuracoSemana
 - * CalculaTotalPenalidadesExcessoDia
 - * CalculaTotalPenalidadeExcessoAulas
 - * CalculaTotalPenalidadeExcessoGrupo
 - * CalculaValorFinal
 - * salvarSolucao
 - * inserirSolucaoHorarios
 - * inserirSolucao

- Quadro

- Atributo

- * quadroHorarios

- Métodos

- * qtdAulasTurmaDia
 - * qtdBuracosDia
 - * buscarTurmaHorario
 - * atualizarDisciplina

- * quantidadeAulasSemanaOcupadas
- * totalDiasSemana
- * quantidadeAulasSemanaOcupadas
- * totalDiasSemana
- * quantidadeAulasSemanaVagas
- * quantidadeAulasDia
- * inicializarQuadro
- * qtdAulasAlocadasHora
- * existeAulaDia
- * qtdBuracoSemana

Relacionamentos entre Classes

Segundo[3], a associação é a forma de se fazer a representação dos relacionamentos que as instâncias de uma classe possuem no momento que o sistema é executado.

No Diagrama da Figura 3.3, temos os seguintes relacionamentos:

- A classe Solucao tem associação com turma e professor. São necessários esses relacionamentos pelo fato da classe Solucao precisar das informações dessas outras classes temporariamente em alguns dos seus métodos no momento de fazer a alocação dos horários.
- Um grupo é composto por turmas, dessa forma, faz-se necessário a associação entre essas classes. Cada grupo deve ter no mínimo uma turma. Pode acontecer de um grupo ser formado por optativas e nenhuma delas serem oferecidas no semestre, mas nesse caso é mais lógico que esse grupo não seja criado. Essa associação também é importante pelo fato das turmas estarem sujeitas a algumas restrições dos grupos que pertencem, como por exemplo, o máximo de aulas por dia e os horários em que são proibidos de serem alocados.
- Tem-se que toda turma possui um professor, com isso, é indispensável esse relacionamento entre eles. Não é possível que uma turma seja alocada no quadro de horários sem saber quem a lecionará. E cada professor possui algumas preferências particulares para dar aula. Uns preferem aulas sem intervalos, outros com menos dias na semana e também o máximo de aulas por dia pode ser diferente para cada docente. Com isso, para se alocar uma turma, além de suas restrições, existem ainda as preferências do professor encarregado por ministrar as aulas.
- Como o professor pertence a um departamento, existe a associação entre eles. E como cada departamento possui restrições de horários, ou seja, horários em que as turmas e professores não podem estar alocados, deve-se levar em consideração esses horários restritos no momento da alocação dos horários do professor.

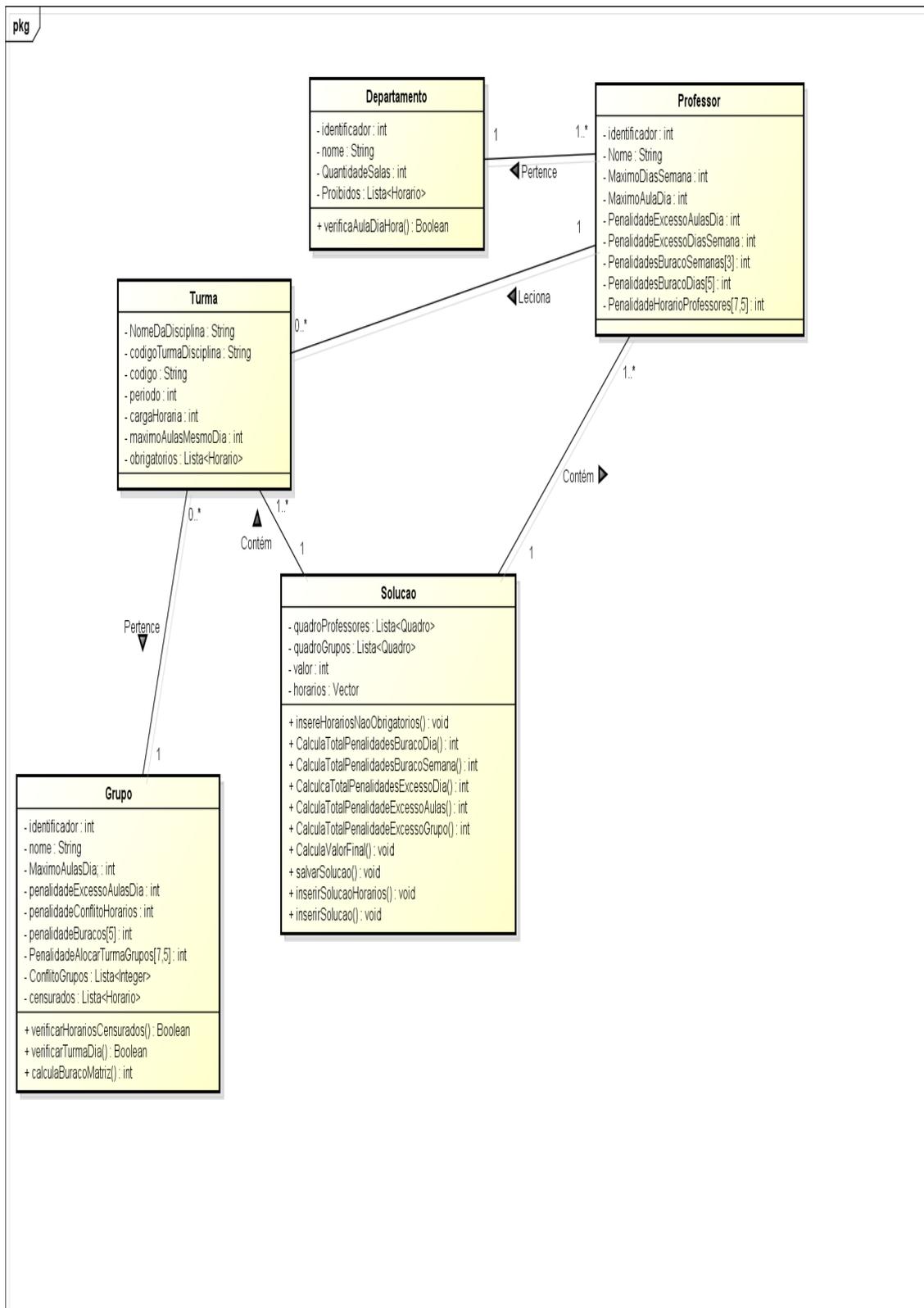


Figura 3.3: Diagrama de classe de análise

3.2.4 Diagrama Classe de Projeto

O Diagrama de Classe de Projeto é uma continuação do Diagrama de Classe de Análise. São adicionados detalhes referentes às tecnologias que serão utilizadas. Novas classes que não estavam definidas no Diagrama de Análise mas que são necessárias para o desenvolvimento da aplicação são inseridas. É onde se define especificamente o tipo de relacionamento que cada classe possui (associação, agregação ou composição).

No Diagrama Classe de Projeto 3.4, a Classe Horário foi adicionada. Ela foi estruturada para armazenar o dia e a hora de uma turma ou Reunião. Se o sistema estivesse sendo desenvolvido numa linguagem não orientada a objeto, seria possível representar Horário como um *struct*, ao invés de uma classe.

Agregação

Em [3], é definido que Agregação é um tipo específico de Associação. Em uma Associação, um objeto não pode estar contido em outro, mas em uma Agregação pode estar. Esse tipo de relacionamento é útil para fazer a representação das relações dos objetos que estão vinculados todo-parte entre si. Vale ressaltar que na Agregação todo objeto parte existe na ausência do objeto todo.

Para saber se existe uma Agregação entre duas classes, [3] propõe a seguinte regra:

- Caso duas classes que estão associadas, como por exemplo A e B tiverem como resposta *sim* para uma das perguntas a seguir, então é provável que exista uma relação de Agregação entre elas, onde a classe A é o todo e a B é parte.

1. *A tem um ou mais B?*
2. *B é parte de A?*

Dessa forma, seguindo a regra acima, foram definidas as agregações presentes no diagrama do projeto:

- Como um departamento possui mais de um professor e como este pertence a no máximo um departamento, então existe a relação de agregação entre eles. De modo que o departamento é o todo e o professor é a parte.
- Outra agregação presente no Diagrama de Classe de Projeto é entre turma e professor. Este pode possuir uma ou mais turmas para o período. Com isso, o professor é o todo e a turma é a parte.
- Há ainda o relacionamento de agregação para turma e grupo. De modo que cada grupo contém uma ou mais turmas. Logo, o grupo é o todo e a turma é a parte.

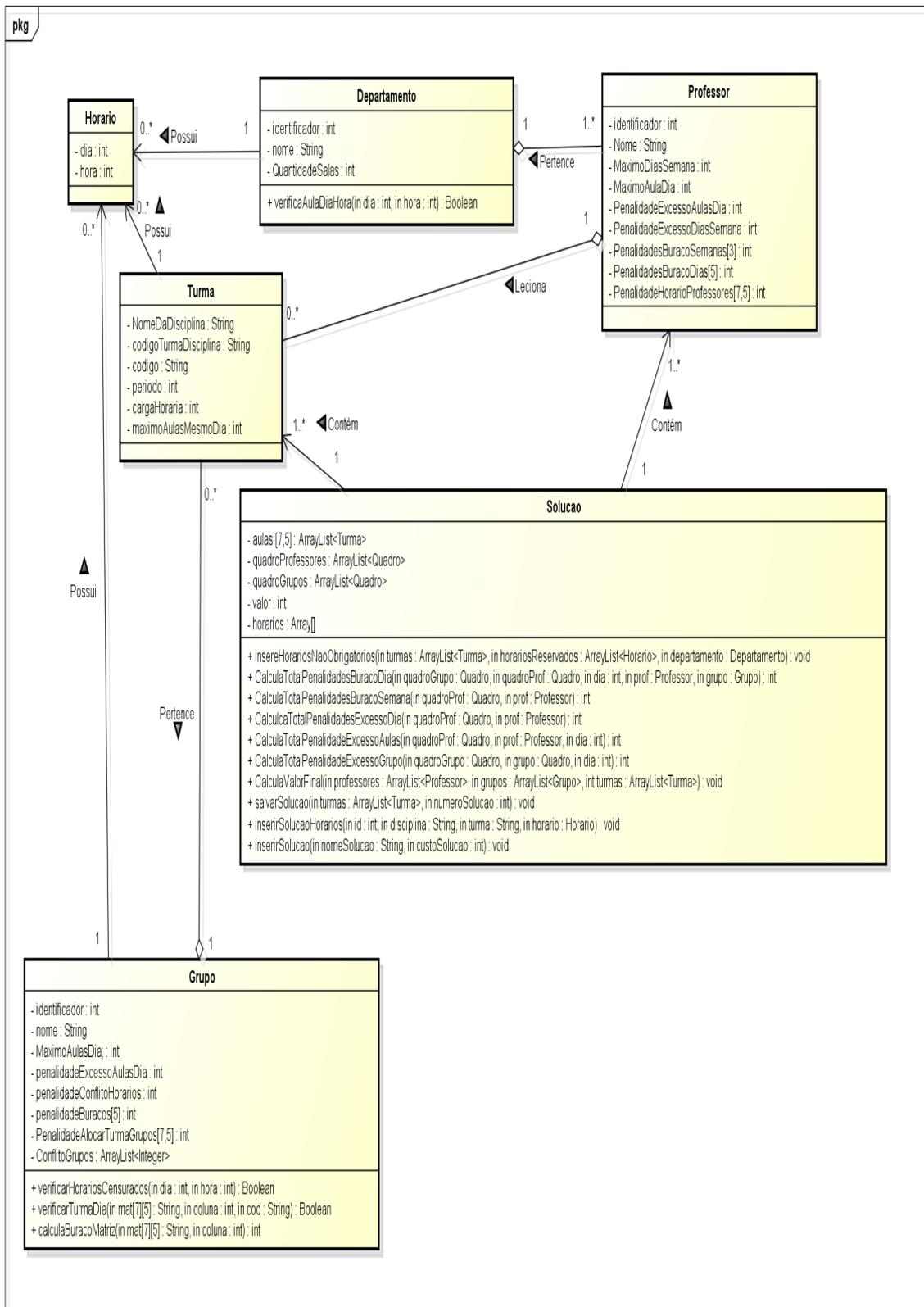


Figura 3.4: Diagrama de Classe de Projeto

3.3 Solução do Problema

A heurística desenvolvida para solucionar o Problema de Alocação de Horários, cujo quadro de horários foi dimensionado para comportar os 5 dias letivos com 7 possíveis horários em cada um, possibilitou a geração de soluções diferentes a cada execução, sendo escolhida pelo coordenador do curso essa quantidade de soluções. Para o algoritmo montar o quadro de horários, foram informadas todas as turmas com os horários que são e os que não são pré-determinados. Dessa forma, para que as soluções fossem diferentes, a estratégia adotada foi alterar a ordem como as turmas estavam organizadas.

O algoritmo foi dividido em duas etapas. A primeira consistiu na alocação dos horários classificados como pré-determinados e a segunda dos horários que não são pré-determinados. A seguir, tem-se a representação da heurística utilizada no sistema em Pseudo-código:

Algoritmo 1 Pseudo-código do Algoritmo de Geração do Quadro de Horários

Input:

```

     $L_{obg}$                                 ▷ Lista de turmas com horários obrigatórios
     $L_{nobj}$                                 ▷ Lista de turmas com horários não-obrigatórios
     $L_{pref}$                                 ▷ Lista das preferências de professores, grupos e departamento
     $L_{hor}$                                 ▷ Lista de horários disponíveis
1:  $S \leftarrow \emptyset$ 
2: for each  $t \in L_{obg}$  do
3:    $S \leftarrow S + \text{horario}(t)$           ▷ a turma  $t$  com seu horário obrigatório
4: end for
5: for each  $t \in L_{nobj}$  do
6:   melhorValor  $\leftarrow \infty$ 
7:   for each  $h \in L_{hor}$  do
8:      $S' \leftarrow S + \{t, h\}$           ▷ a turma  $t$  associada ao horário  $h$ 
9:     valor'  $\leftarrow \text{avalia}(S', L_{pref})$ 
10:    if valor' < melhorValor then
11:      melhorHorario  $\leftarrow h$ 
12:      melhorValor  $\leftarrow \text{valor}'$ 
13:    end if
14:  end for
15:   $S \leftarrow S + \{t, \text{melhorHorario}\}$   ▷ a turma  $t$  associada ao melhor horário disponível
16: end for
    return S

```

3.3.1 Alocação dos Horários Pré-Determinados

Os horários pré-determinados são aqueles que não podem ser alterados no momento da alocação no quadro de horários. Independente da solução gerada, esses horários devem se manter inalterados quanto ao dia e hora nos quadros de horários. Por exemplo, imagine um horário pré-determinado nos dias terça-feira e quinta-feira às 09:00-11:00. O sistema, ao gerar o quadro de horários, não pode de forma alguma alterar o dia ou a hora desse horário. Os horários pré-determinados são classificados de duas formas: Fixos e os Reservados.

Fixos

Em uma grade curricular de um ensino superior é comum que tenham disciplinas de outros departamentos que complementem a ementa do curso. Por exemplo, no campus de Rio das Ostras, o curso de Ciência da Computação possui disciplinas do Departamento de Matemática e Física, como Cálculo I, Cálculo II, Álgebra Linear, Física, entre outras. No mesmo campus, o curso de Psicologia possui a disciplina de Estatística, assim como o curso de Engenharia de Produção possui na sua grade disciplinas de programação. Dessa forma, essas disciplinas são de responsabilidade do departamento ao qual pertence, podendo este fixar o dia e a hora em que essas turmas terão aula. Com isso, não é possível alterar essa fixação inicial no momento da criação de um quadro de horários.

Dessa forma, a heurística realizou primeiramente a inserção no quadro de horários dessas turmas classificadas como horários Fixos, para que não ocorresse conflitos ou indisponibilidade de alocar no mesmo horário.

Reservados

Além dos horários Fixos, foi classificado um outro tipo para os horários pré-determinados, que são os Reservados. Em cursos universitários é comum que cada Departamento, Colegiado ou Unidades realizem reuniões para discutir sobre algum tema, definir alguma pauta, fazer votação para alguma eleição e outras coisas. E em muitos casos, essas reuniões são totalmente ignoradas no momento de se fazer o quadro de horários, fazendo com que professores e alunos tenham indisponibilidade de participar e contribuir na reunião. Pensando nisso, foram classificados como Reservados os horários das reuniões, para que dessa forma professores e alunos tenham total disponibilidade de participar.

O algoritmo após inserir os horários Fixos fez a inserção dos horários Reservados no quadro. Até aqui, foram inseridos apenas os horários pré-determinados, incluindo os horários Fixos e Reservados nos seus respectivos dias e horas. A seguir será abordada a forma como foi feito a alocação dos horários que não são pré-determinados.

3.3.2 Alocação dos Horários não Pré-Determinados

Os horários que não são pré-determinados são aqueles que no momento de gerar o quadro de horários não precisam ser alocados especificamente em um dia e hora, como no caso dos pré-determinados. Após serem alocados os horários Fixos e os Reservados é iniciado o processo de alocação dos horários que não são pré-definidos.

Nessa etapa, cada turma só é inserida em um determinado horário do quadro quando o custo associado a esta alocação for o menor possível dentre todos os horários disponíveis. Este custo é calculado levando em consideração as preferências do professor que leciona a turma. Por exemplo, se tem preferência em dar aula à tarde ou se há excesso de aulas em um dia. Outros critérios também são levados em consideração, como a formação ou não de buracos (janelas) nos horários dos professores e dos grupos e as preferências do grupo ao qual a turma está associada (por exemplo, para o 1º e 2º período as aulas serem na parte da manhã). O buraco é caracterizado como o intervalo de horários vagos entre as aulas

de uma turma. Na segunda-feira, existe uma janela de tamanho 2 das 09:00 horas até às 13:00 horas. Na terça-feira, de 09:00 horas até 20:00 horas está vago, caracterizando um buraco de tamanho 5 no horário de uma turma de um grupo ou professor. Esses buracos são representados na Figura 3.5.

	Segunda-feira	Terça-feira
07:00-09:00	●	●
09:00-11:00	●	●
11:00-13:00	●	●
14:00-16:00	●	●
16:00-18:00	●	●
18:00-20:00	●	●
20:00-22:00	●	●

Figura 3.5: Exemplo de buracos nos horários

Caso ocorra alguma situação indesejável, é somada uma penalidade ao custo total de se alocar a turma no horário específico. Vale ressaltar que o horário da turma, uma vez inserido no quadro de horários, não pode ser alterado. O algoritmo repete essa etapa de alocação das turmas diversas vezes. Em cada vez, a lista de turmas a serem alocadas é ordenada aleatoriamente, o que permite que as soluções finais sejam distintas.

Dessa forma, espera-se que algumas delas tenham qualidade melhor e que tenham mais chance de serem escolhidas pelo coordenador do curso como solução final do sistema. Esta também pode passar por um processo de “refinamento”, onde o coordenador fixa os horários de algumas turmas que ficaram bons e pede que o sistema gere novos quadros de horários com os horários daquelas turmas fixados. Com isso, intensifica-se a busca por boas soluções em um espaço de soluções reduzido. Este processo pode ser repetido até que haja um quadro de horários que seja satisfatório para o coordenador. Essa abordagem caracteriza a heurística como sendo um algoritmo do tipo *multi-start*.

3.4 Interfaces do Sistema

As principais interfaces do sistema gerenciador de quadro de horários são as seguintes:

3.4.1 Login

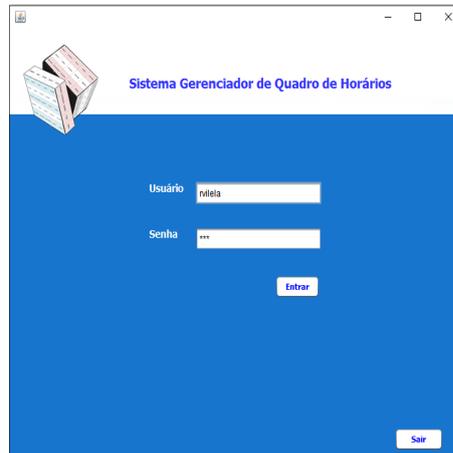


Figura 3.6: Tela login

A interface mostrada na Figura 3.6 é onde o usuário faz o login no sistema. Existem dois tipos de usuários: professor e coordenador.

3.4.2 Gerenciar Departamento

Interface relacionada ao cadastro de um departamento no sistema.

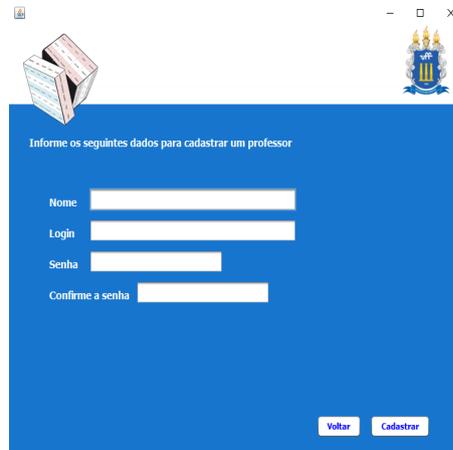


Figura 3.7: Tela de Cadastrar departamento

A interface na Figura 3.7 é onde o coordenador do curso realiza o cadastro de um departamento. São informados o nome, quantidade de salas e os horários reservados do departamento.

3.4.3 Gerenciar Professor

Interface relacionada ao cadastro de um professor no sistema.



The screenshot shows a web browser window with a blue header. On the left is a logo of a building, and on the right is a crest. The main content area has a blue background and contains the text 'Informe os seguintes dados para cadastrar um professor'. Below this text are four input fields: 'Nome', 'Login', 'Senha', and 'Confirme a senha'. At the bottom right of the form are two buttons: 'Voltar' and 'Cadastrar'.

Figura 3.8: Tela de Cadastrar professor

Um professor é cadastrado no sistema pelo coordenador do curso através da interface mostrada na Figura 3.8. São informados o nome, login e a senha do novo professor.

3.4.4 Gerenciar Turma

Interfaces referentes ao gerenciamento de uma turma no sistema.



The screenshot shows a web browser window with a blue header. On the left is a logo of a building, and on the right is a crest. The main content area has a blue background and contains the text 'Informe os dados a seguir para cadastrar uma Turma'. Below this text are several input fields and dropdown menus: 'Nome da disciplina', 'Código Disciplina', 'Carga Horária' (with a spinner set to 2), 'Período' (with a spinner set to 0), 'Código Turma', and 'Máximo de aulas por dia' (with a spinner set to 0). There are three dropdown menus: 'Selecione o Professor' (set to André Renato), 'Selecione o Grupo' (set to 1 período), and 'Selecione o Departamento' (set to ICT). At the bottom left, there is a label 'Informe os horários obrigatórios' and a button 'Obrigatórios'. At the bottom right are two buttons: 'Voltar' and 'Cadastrar'.

Figura 3.9: Tela de Cadastrar turma

A interface na Figura 3.9 é para cadastrar uma turma no sistema pelo coordenador do curso. São informados o nome, código da disciplina, código da turma, carga horária, período, máximo de aulas por dia permitido, professor, grupo, departamento e os horários obrigatórios (caso seja de outro departamento).

Selecione a Turma que deseja atualizar: Matemática Discreta C1 RCT0003

Carga Horária: 4 | Período: 1 | Máximo de aulas por dia: 1

Selecione o Professor: Fernando N.

Selecione o Grupo: 1 período

Selecione o Departamento: ICT

Atualize os horários obrigatórios: Obrigatórios

Voltar | Atualizar

Figura 3.10: Tela de Atualizar turma

Para os dados de uma turma serem atualizados no sistema, o coordenador do curso utiliza a interface na Figura 3.10. Através de um seletor, a turma é escolhida para ser atualizada. São fornecidas a carga horária, período, máximo de aulas por dia, professor, grupo, departamento e os horários obrigatórios caso possua.

3.4.5 Gerenciar Grupo

Interfaces relacionadas ao cadastro dos dados dos grupos no sistema.

Informe os dados a seguir para cadastrar um Grupo

Nome do Grupo: | Máximo de aulas por dia: 0

Penalidade de excesso aulas por dia: 0

Penalidades de conflitos de horários: 0

Informe a penalidade de tamanho de buraco nos dias:

Tamanho 1: 0 | Tamanho 2: 0 | Tamanho 3: 0 | Tamanho 4: 0 | Tamanho 5: 0

Cancelar | Próximo passo

Figura 3.11: Tela de Cadastrar grupo

A interface mostrada na Figura 3.11 é onde o coordenador do curso informa os dados do grupo para o cadastro. São informados o nome, máximo de aulas por dia, penalidade de excesso de aula por dia, penalidade de conflito de horários e as penalidades de buracos nos dias.

	Segunda-feira	Terça-feira	Quarta-feira	Quinta-feira	Sexta-feira
07:00-09:00	0	0	0	0	0
09:00-11:00	0	0	0	0	0
11:00-13:00	0	0	0	0	0
14:00-16:00	0	0	0	0	0
16:00-18:00	0	0	0	0	0
18:00-20:00	0	0	0	0	0
20:00-22:00	0	0	0	0	0

Figura 3.12: Tela de Cadastrar penalidades de alocar turmas do grupo

Na interface na Figura 3.12 o coordenador do curso informa as penalidades de alocar as turmas pertencentes ao grupo para cada horário na semana.

	Segunda-feira	Terça-feira	Quarta-feira	Quinta-feira	Sexta-feira
07:00-09:00	<input type="radio"/>				
09:00-11:00	<input type="radio"/>				
11:00-13:00	<input type="radio"/>				
14:00-16:00	<input type="radio"/>				
16:00-18:00	<input type="radio"/>				
18:00-20:00	<input type="radio"/>				
20:00-22:00	<input type="radio"/>				

Figura 3.13: Tela de Cadastrar horários censurados do grupo

Na interface na Figura 3.13 o coordenador do curso informa os horários censurados do grupo que está sendo cadastrado na semana.

3.4.6 Gerenciar Preferências Professor

Interfaces relacionadas ao cadastro das preferências de um professor no sistema.

Figura 3.14: Tela de Cadastrar preferência penalidades

A interface da Figura 3.14 é onde o professor informa as suas preferências para serem cadastradas. São informadas as penalidades de excesso de aulas nos dias, penalidade de excesso de aulas na semana, máximo de aulas por dia, máximo de aulas na semana, penalidades de tamanho de buraco nos dias e na semana.

Figura 3.15: Tela de Cadastrar preferência horários

Na interface da Figura 3.15 o professor informa as suas preferências de horários para lecionar uma turma na semana.

3.4.7 Gerenciar Solucao

Interfaces relacionadas ao gerenciamento da Solucao no sistema.

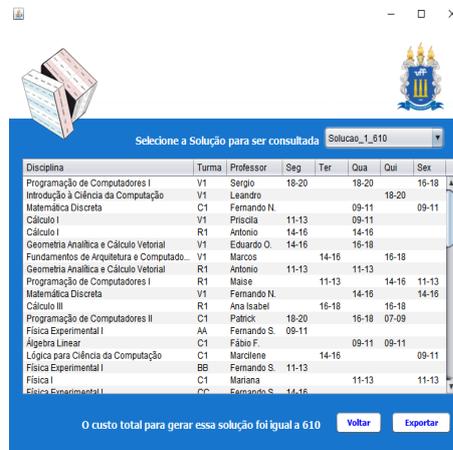


Figura 3.16: Tela de Consultar solução

Na interface da Figura 3.16 o coordenador do curso pode consultar e exportar as soluções geradas pelo sistema. Através de um seletor, a solução é escolhida pelo nome e é exibido o quadro de horários contendo as turmas, professor e os horários na semana.

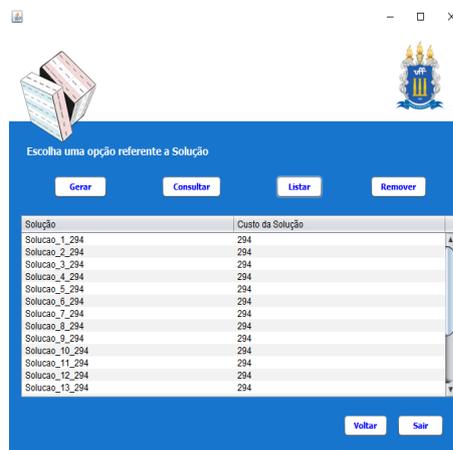


Figura 3.17: Tela de Listar soluções

O coordenador do curso utiliza a interface da Figura 3.17 para listar as soluções geradas pelo sistema. Mostrando os nomes e o custos associados.

Capítulo 4

Resultados Obtidos

Para gerar os quadros de horários, foi necessário realizar o cadastro das informações no banco de dados através do sistema. As turmas e os professores cadastrados foram baseados no quadro de horários **201601-v2**. Para analisar o comportamento da heurística, o sistema foi submetido a 3 cenários em que os dados cadastrados foram modificados com o objetivo de verificar se o algoritmo conseguiu se adaptar e retornar boas soluções. Uma solução é boa quando o custo associado é o menor possível. Em cada cenário foram geradas 50 soluções.

4.1 Cenário 1

No Cenário 1, as penalidades dos professores e dos grupos foram cadastradas com o valor 5 em média. Para este caso, a melhor solução retornada pelo sistema teve custo de 1221 e média igual a 1248.16. Todas as soluções foram geradas em aproximadamente 13 segundos.

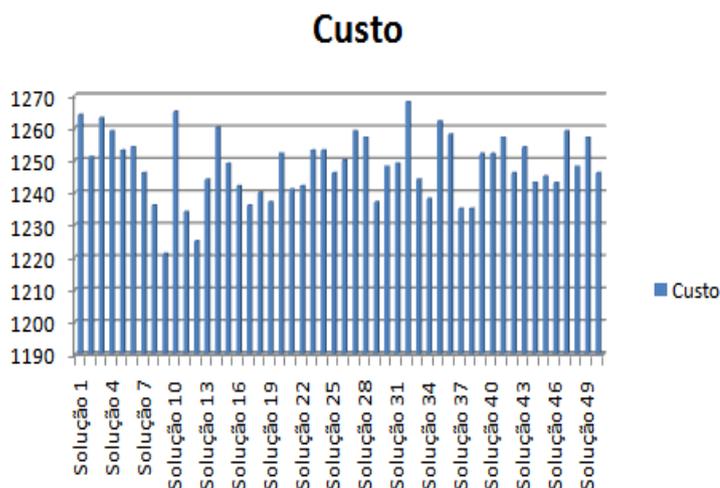


Figura 4.1: Custos das soluções - Cenário 1

Na figura 4.1 é mostrado um comparativo dos custos das soluções. Pode-se notar que apenas 5 casos tiveram custos maiores ou iguais a 1260, que foram as piores soluções.

4.2 Cenário 2

Para o segundo Cenário, 5 dos 9 grupos cadastrados tiveram as penalidades modificadas. Para escolher os grupos que tiveram as penalidades ajustadas foram levados em consideração os alunos recém-chegados a faculdade (1º e 2º período) e os que estão no 6º período em diante. Para que os calouros tivessem aulas até 16 horas, foram aumentadas as penalidades de alocar uma turma pertencente aos grupos (1º e 2º período) nos horários de 16 horas em diante. Para os alunos veteranos que procuram estágio, as penalidades dos grupos de alocar as turmas do 6º ao 8º período foram incrementadas nos horários na parte da manhã. Em média, as penalidades dos grupos aumentaram 5 vezes mais. Neste cenário, a melhor solução foi 1338 e a média de custo foi 1361.86. O tempo para gerar todas soluções foi de aproximadamente 9 segundos.

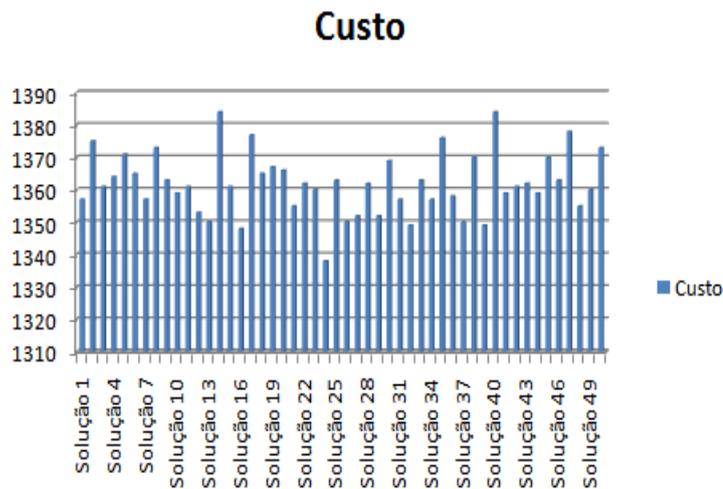


Figura 4.2: Custos das soluções - Cenário 2

Na Figura 4.2 nota-se que a maioria das soluções geradas ficou com um nível de qualidade próximo. Apesar do aumento das penalidades dos grupos, o algoritmo conseguiu se adaptar ao novo cenário.

4.3 Cenário 3

Neste cenário, além das penalidades dos grupos ajustadas no caso anterior, foram agora modificadas as preferências e penalidades de 10 professores aleatórios dos 15 cadastrados. Em média, foram aumentadas em 4 vezes mais. Vale ressaltar que os professores que lecionam as turmas cujos os horários são pré-determinados não tiveram as preferências cadastradas. O sistema retornou como melhor solução um quadro de horários com custo 1650 e média igual a 1685.96. O tempo para gerar todas soluções foi de aproximadamente 9 segundos.

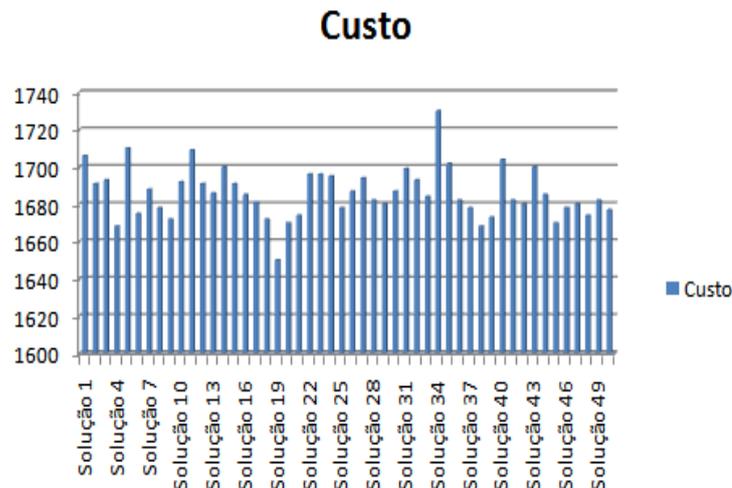


Figura 4.3: Comparações dos custos - Cenário 3

De acordo com a Figura 4.3, nota-se que a pior solução gerada para este cenário possui qualidade bem inferior às demais, uma vez que o custo associado foi bem maior.

4.4 Análise dos Resultados

A tabela 4.1 mostra o resultado do comportamento da heurística para os cenários propostos. Um indicador que define se um algoritmo é robusto é o desvio-padrão. Se um algoritmo sempre produz resultados com um nível de qualidade próximo, ele é classificado como robusto e o desvio-padrão não varia muito. A robustez indica que a qualidade final da solução não oscila muito, mas está próxima para cada cenário.

Cenário	Melhor solução	Média custo	Desvio padrão custo
1	1221	1248,16	10,24
2	1338	1361,86	9,53
3	1650	1685,96	13,63

Tabela 4.1: Resultados

Com o menor desvio-padrão e a média das soluções próxima ao melhor custo encontrado, o algoritmo teve melhor comportamento no cenário 2. Indicando que as soluções geradas tiveram um nível de qualidade mais próximo. Apesar do aumento das penalidades dos professores no cenário 3, o desvio-padrão não aumentou muito. Mostrando que a heurística conseguiu adaptar-se bem as novas restrições impostas.

Cenário	Média total de penalidades	Média porcentagem grupos	Média porcentagem professores
1	1248,16	44	56
2	1361,86	47	53
3	1685,96	40	60

Tabela 4.2: Penalidades contabilizadas nos cenários

A tabela 4.2 mostra para cada cenário a média do total de penalidades contabilizado nas soluções geradas considerando os professores e grupos. No cenário 1, a média do total de penalidades foi 1248.16, onde em média 44% desse valor corresponde as penalidades dos grupos e 56% à média do total das penalidades dos professores. Com o aumento das penalidades para os grupos, uma mesma solução do cenário 1 teria aumento de 176% se fosse produzida no cenário 2. Isto não ocorreu, uma vez que as soluções pioraram em média 9.11%. Em relação ao aumento das penalidades e preferências dos professores, uma mesma solução do cenário 2 teria aumento de 159% se fosse produzida no cenário 3. O que também não ocorreu, pois as soluções pioraram em média 23.8%.

Dessa forma, o algoritmo conseguiu se adaptar bem aos diferentes cenários, absorvendo bem o crescimento das penalidades e gerando soluções com boa qualidade num curto intervalo de tempo.

4.4.1 Horários indesejados

No cenário 2, foram modificadas as penalidades dos grupos para evitar horários indesejados para calouros e veteranos, como dito na seção 4.2. Afim de mostrar quantas aulas tiveram horários ruins para os períodos escolhidos, foi feita uma contagem na base de dados para as 50 soluções geradas levando em consideração as turmas que pertencem apenas ao Departamento de Computação. Para o primeiro cenário houve 540 horários indesejados, dos quais 336 foram na alocação das turmas do 1º e 2º período e 204 para as turmas do 6º ao 8º período. Para o cenário 2, tiveram 114 horários indesejados. Onde 113 foram para as turmas do 1º e 2º período e apenas 1 para as turmas do 6º ao 8º período.

Uma razão pela qual as turmas do 1º e 2º período tiveram mais horários indesejados é o fato de nesses períodos existirem mais horários pré-determinados, o que implica em menos escolhas de horários para o algoritmo alocar as turmas. Dessa forma, pode-se concluir que a heurística atendeu bem as restrições impostas, uma vez que após o aumento das penalidades no cenário 2, a quantidade de aulas em horários ruins que era de 540 no cenário 1 reduziu para 114 no segundo cenário, correspondendo a uma redução de 80% dos horários indesejados.

O quadro de horários da Figura 4.4 é a melhor solução gerada no cenário 2. Ilustra como é a visualização da solução exportada após ter sido gerada pelo sistema. É semelhante ao que é feito de forma manual pelo coordenador do curso a cada início de período.

Disciplina	Turma	Professor	Seg	Ter	Qua	Qui	Sex
Matemática Discreta	C1	Fernando N.			09-11		09-11
Cálculo I	V1	Priscila	11-13		09-11		
Geometria Analítica e Cálculo Vetorial	R1	Antonio	11-13		11-13		
Introdução à Ciência da Computação	V1	Leandro		11-13			
Cálculo I	R1	Antonio	14-16		14-16		
Geometria Analítica e Cálculo Vetorial	V1	Eduardo O.	14-16		16-18		
Matemática Discreta	V1	Fernando N.			14-16		14-16
Fundamentos de Arquitetura e Computadores	V1	Marcos		07-09			11-13
Programação de Computadores I	V1	Sergio		09-11		14-16	07-09
Programação de Computadores I	R1	Maise	09-11	14-16		16-18	
Física Experimental I	BB	Fernando S.	11-13				
Álgebra Linear	C1	Fábio F.			09-11	09-11	
Física I	C1	Mariana			11-13		11-13
Física Experimental I	CC	Fernando S.	14-16				
Cálculo II	C1	Eduardo O.	16-18		14-16		
Cálculo III	R1	Ana Isabel		16-18		16-18	
Programação de Computadores II	C1	Patrick		14-16		07-09	09-11
Lógica para Ciência da Computação	C1	Marcilene		11-13			07-09
Física Experimental I	AA	Fernando S.	09-11				
Física II	C1	Carlos Magno	11-13		11-13		
Equações Diferenciais	R1	Fábio G.		09-11		09-11	
Linguagens de Programação	C1	Flavia		11-13			16-18
Física Experimental II	AA	Robson			11-13		
Análise Combinatória	C1	Fernando N.		14-16			
Física Experimental II	DD	Robson			14-16		
Laboratório de Circuitos Digitais	C1	Luciano				14-16	
Circuitos Digitais	C1	Luciano	16-18			16-18	
Programação com Arquivos	C1	André Renato		07-09			
Estrutura de Dados	C1	Dalessandro	09-11				18-20
Técnicas de Programação Avançada	C1	Eduardo M.	11-13		16-18		
Metodos numéricos	R1	Fábio G.		11-13		11-13	
Análise de Projetos de Algoritmos	C1	Maise	16-18				11-13
Probabilidade e Estatística	C1	Ana Paula	14-16	16-18			
Arquitetura de Computadores	C1	Luciano		14-16			18-20
Banco de dados	C1	Leila				14-16	14-16
Engenharia de Software	C1	Adriana	09-11		14-16		
Rede de Computadores I	C1	Leandro				14-16	18-20
Linguagens formais	C1	Dalessandro	11-13	07-09			
Algoritmos em grafos	C1	Danilo	09-11			16-18	
Sistemas Operacionais	C1	Marcos		09-11	11-13		
Interface Humano-Computador	C1	Sergio	14-16				14-16
Inteligência Artificial	C1	Flavia		14-16		14-16	
Compiladores	C1	Eduardo M.	16-18				16-18
Empreendedorismo	C1	Mateus		09-11			
Projeto de Aplicação I	C1	Eduardo M.				14-16	
Projeto de Aplicação II	C1	Danilo					18-20
Estágio Supervisionado	C1	Maise		16-18			
Ética	C1	Maria Helena		18-20			
Tópicos Especiais em Computação III	C1	Patrick				09-11	18-20
Tópicos Especiais em ES	C1	Adriana	11-13		11-13		
Introdução à Computação Móvel	C1	Copetti		11-13			14-16
Tópicos Especiais em Algoritmos	C1	Danilo	14-16			18-20	
Tópicos Especiais em Redes e Sists. Dist.	C1	Leandro		14-16	14-16		
Lambda-Cálculo	C1	Eduardo O.		16-18			
Tópicos Especiais em Banco de Dados	C1	Leila		18-20			16-18
Métodos Computacionais em Física	C1	Fernando S.	18-20				
Tópicos Especiais em Combinatória II	C1	Marcilene	09-11			14-16	

Figura 4.4: Ilustração da solução exportada

Capítulo 5

Conclusão

O desenvolvimento da heurística para solucionar o Problema de Alocação de Horários em Cursos Universitários proporcionou um conhecimento satisfatório sobre o tema. Além da aprendizagem adquirida ao criar as interfaces para interação do usuário com o sistema.

Resolver esse problema de forma manual é complexo, uma vez que devem ser levadas em consideração muitas restrições impostas. E dependendo do caso alguma restrição pode não ser considerada, tornando a solução não satisfatória. No Departamento de Computação de Rio das Ostras esse trabalho ainda é feito de forma manual, necessitando de bastante tempo para ser elaborado. A heurística desenvolvida se mostrou eficiente para resolver Problemas dessa categoria, com soluções de boa qualidade num curto intervalo de tempo.

Para os cenários abordados, era esperado que as soluções fossem bastante piores com o aumento das penalidades, porém o algoritmo conseguiu absorver bem as modificações dos dados e gerou soluções satisfatórias. Foram também reduzidos em aproximadamente 80% os horários indesejados para calouros e veteranos com os ajustes das penalidades dos grupos no cenário 2. Dessa forma, o sistema desenvolvido se mostrou uma boa ferramenta para alocação das turmas no quadro de horários em cursos universitários, onde são levadas em consideração as preferências dos professores, do departamento e dos alunos.

Capítulo 6

Trabalhos Futuros

Como trabalhos futuros deseja-se realizar um estudo de novas restrições, penalidades e preferências dos envolvidos no problema com o objetivo de proporcionar a estes uma solução mais satisfatória. Evoluir o sistema para uma plataforma Web, de maneira a descentralizar o cadastro e a atualização das preferências dos professores. Além disso, desenvolver métodos mais robustos (meta-heurísticas), que possuem resultados mais eficientes para Problemas de Otimização Combinatória em relação à heurística desenvolvida. Dentre os quais pode-se citar: GRASP (*Greedy Randomized Adaptive Search Procedures*) [18], VNS (*Variable Neighborhood Search*) [19], VND (*Variable Neighborhood Descent*) [19] e Busca Tabu (*Tabu Search*) [20]. Deseja-se também definir as disciplinas de outros departamentos como não pré-determinadas, para que os horários destas possam ser geradas pelo sistema, de maneira que as preferências dos envolvidos sejam atendidas de forma mais igualitária.

Referências Bibliográficas

- [1] Cordenonsi, A. Z. ; Aramburu, L. G. C. ; Almanca, L. "Resolução do Problema de Quadro de Horários Através de um Algoritmo de Satisfação de Restrições. In: VIII Simpósio de Informática e III Mostra de Software Acadêmico, 2003, Uruguaina - RS, 2003."
- [2] Schaerf, A. "A survey of automated timetabling." *Artificial intelligence review* 13.2 (1999). Dipartimento di Informatica e Sistemistica Università di Roma La Sapienza.
- [3] Bezerra, E. "Princípios de Análise e Projeto de Sistemas com UML". Brasil. São Paulo. Campus. Um guia prático para modelagem de sistemas orientados a objetos através da Linguagem de Modelagem Unificada.
- [4] Elmasri. N. "Sistema de banco de dados". 6ª edição. São Paulo. 2011.
- [5] Hamawaki, C. D. L., Yamanaka, K., Oliveira, R. C., & Osvaldo, T. (2005) "Alocação de Grade Horária em Instituições de Ensino Superior Utilizando Algoritmos Genéticos. IV Simpósio de Informática da Região Centro do RS - SIRC/RS 2005 - Santa Maria."
- [6] Vieira, F., & Macedo, H. (2011). "Sistema de alocação de horários de cursos universitários: um estudo de caso no departamento de computação da Universidade Federal de Sergipe. *Scientia Plena*, 7(3)."
- [7] Xavier, B. M., da Silva, A. D., Vianna, D. S., Costa, H. G., & Coelho, W. B. (2013). "Proposta de Alocação de Horários de Professores e Turmas em Instituições de Ensino Superior Utilizando uma Heurística VNS/VND". *Simpósio Brasileiro de Pesquisa Operacional - A Pesquisa Operacional na busca de eficiência nos serviços públicos e/ou privados*
- [8] Moura, A.; Scaraficci, R.; Silveira, R.; dos Santos, V. (2004). "Técnicas Metaheurísticas Aplicadas à Construção de Grades de Horárias Escolares". *XXXVI SBPO - O impacto da Pesquisa Operacional nas novas tendências multidisciplinares*.
- [9] Poulsen, C.J.B.; Bandeira, D.L. (2013). "Uma eficiente heurística baseada na estratégia de divisão-e-conquista para o School Timetabling Problem". *Simpósio Brasileiro de Pesquisa Operacional - A Pesquisa Operacional na busca de eficiência nos serviços públicos e/ou privados*.
- [10] Even, S.; Itai, A.; Shamir, A. "On the complexity of timetabling and multicommodity flow problems. *SIAM Journal of Computation*, n. 5, p. 691-703, 1976."

- [11] Burke, E.; Werra, D. D.; Kingston, J. *"Applications to timetabling. Handbook of Graph Theory. p.445-474. CRC Press."*
- [12] Navarro, F.; Coelho, F. *"Estudo de Coloração Aplicado ao Problema de Alocação de Horário de Professores"*
- [13] Schönberger, J.; Mattfeld, D. C. e Kopfer, H. *"Algorithm timetabling for non-commercial sport leagues. European Journal of Operational Research, v. 153, n. 1, p. 102 - 116. (2004)"*
- [14] Schoenauer, M.; Semet, Y. *"An efficient memetic, permutation based evolutionary algorithm for real-world train timetabling. p. 2752 - 2759, Congress on Evolutionary Computation, IEEE Press. (2005)"*
- [15] Burke, E.; Cowling, P.; Causmaecker, P.; Bergue, G. *"A memetic approach to the nurse rostering problem. Applied Intelligence, v. 15, n. 3, p. 199 - 214. (2001)"*
- [16] Carter, M.; Laporte, G. *"Recent developments in practical course timetabling. In: Burke and Carter. pp. 3 - 19. (1998)"*
- [17] Fang, H. *"Genetic Algorithms in Timetabling and Scheduling. 1994. Tese - Department of Artificial Intelligence, University of Edinburgh. (1994)"*
- [18] Feo T. A.; Resende M. G. C. *"Greedy Randomized Adaptive Search Procedures. Journal of Global Optimization 6, p. 109-133. (1995)"*
- [19] Mladenovic N.; Hansen. P. *"Variable neighborhood search. Computers & Operations Research, 24:1097-1100. (1997)"*
- [20] Glover F.; Laguna M. *"Tabu Search. Kluwer Academic Publishers, 1a Edição. (1997)"*

Capítulo 7

Apêndice

7.1 Descrição dos casos de uso

Atualizar Departamento (CSU01)

Ator Primário: Coordenador do curso.

Descrição: Coordenador do curso atualiza os dados de um departamento no sistema.

Precondição: O Coordenador do curso está autenticado e o departamento se encontra cadastrado no sistema.

Fluxo Principal:

1. Coordenador do curso solicita atualizar os dados de um departamento.
2. Sistema solicita o nome do departamento.
3. Coordenador do curso escolhe o nome do departamento.
4. Sistema exibe os dados (nome, quantidade de salas e os horários proibidos) para serem modificados.
5. Coordenador do curso modifica os dados do departamento.
6. Sistema exibe dados atualizados.
7. Coordenador do curso clica em "Atualizar".
8. Sistema pede confirmação do coordenador do curso para atualizar os dados.
9. Coordenador do curso confirma atualização.
10. Sistema atualiza os dados do departamento.
11. Sistema informa ao coordenador do curso que o departamento foi atualizado com sucesso.
12. Caso de Uso termina.

Fluxo Alternativo (Passo : 9) Confirmação da atualização

1. Coordenador do curso não confirma a atualização do departamento.
2. Sistema não atualiza os dados do departamento.
3. Caso de Uso continua a partir do passo 2.

Fluxo de Exceção (passo 3): Inserção de dado errada

1. Coordenador do curso detecta que escolheu o nome do departamento de forma incorreta.
2. Sistema possibilita que o coordenador do curso escolha o nome de outro departamento.
3. Coordenador do curso escolhe o nome do departamento correto.
4. Sistema aceita a correção e o Caso de Uso continua a partir do passo 2.

Fluxo de Exceção (passo 5): Modificação de dados errada

1. Coordenador do curso detecta que fez a modificação errada.
2. Coordenador do curso corrige a informação que foi modificada de forma incorreta.
3. Sistema aceita a correção e o Caso de Uso continua a partir do passo 5.

Fluxo de Exceção (passo 8): Não atualização do departamento

1. Coordenador do curso não confirma atualização dos dados do departamento.
2. Sistema aceita a não atualização do departamento e o Caso de Uso continua a partir do passo 2.

Pós-Condição: Os dados do departamento foram atualizados no sistema.

Cadastrar Departamento (CSU02)

Ator primário: Coordenador do curso.

Descrição: Coordenador do curso cadastra os dados de um novo departamento no sistema.

Precondições: O coordenador do curso está autenticado no sistema.

Fluxo Principal:

1. Coordenador do curso seleciona a opção de cadastrar um novo departamento.
2. Sistema solicita os dados do departamento (nome, quantidade de salas e os horários proibidos).
3. O coordenador do curso fornece os dados solicitados e clica em "Cadastrar".
4. Sistema registra os dados fornecidos.
5. Sistema informa ao coordenador do curso que o departamento foi cadastrado com sucesso.
6. Caso de Uso termina.

Fluxo Exceção (Passo 4): Inserção de dado errada

1. Coordenador do curso detecta que forneceu algum dado errado do departamento.
2. Sistema permite que o dado seja corrigido.
3. Coordenador do curso corrige o dado que foi fornecido de forma incorreta.
4. Sistema aceita a correção e o Caso de Uso continua a partir do passo 3.

Pós-Condição: Os dados do departamento foram cadastrados no sistema.

Remover Departamento (CSU03)

Ator Primário: Coordenador do curso.

Descrição: Coordenador do curso exclui um departamento do sistema.

Precondição: O coordenador do curso está autenticado e o departamento se encontra cadastrado no sistema.

Fluxo Principal:

1. Coordenador do curso solicita a remoção de um departamento.
2. Sistema solicita o nome do departamento.
3. Coordenador do curso fornece o nome do departamento
4. Coordenador do curso clica em "Remover".
5. Sistema pede para coordenador do curso confirmar remoção do departamento.
6. Coodenador confirma remoção do departamento.
7. Sistema verifica se existe turma associada ao departamento.
8. Sistema remove o departamento.
9. Sistema informa ao coordenador que o departamento foi removido.
10. Coordenador do curso fica ciente da remoção.
11. Caso de Uso termina.

Fluxo Alternativo (Passo : 5) Confirmação de remoção

1. Coordenador do curso não confirma a remoção do departamento.
2. Sistema não remove os dados do departamento.
3. Caso de Uso continua a partir do passo 2.

Fluxo de Exceção (Passo 3): Escolha de departamento errado

1. Coordenador do curso detecta que informou o nome do departamento errado.
2. Sistema possibilita que o coordenador forneça o nome do departamento.

3. Coordenador do curso fornece o nome do departamento.
4. Sistema aceita a correção e o Caso de Uso continua a partir do passo 2.

Fluxo de Alternativo (Passo 7): Existe turma associada ao departamento.

1. Sistema informa que existe turma associada ao departamento.
2. Coordenador do curso fica ciente que existe turma associada ao departamento.
3. Sistema não permite a remoção do departamento e o Caso de Uso continua a partir do passo 2.

Fluxo de Alternativo (Passo 7): Nenhuma turma associada ao departamento.

1. Sistema verifica que não existe turma associada ao departamento.
2. Caso de Uso continua a partir do passo 8.

Pós-Condição: Os dados do departamento foram removidos do sistema.

Listar Departamentos (CSU04)

Ator Primário: Coordenador do curso.

Descrição: Coordenador do curso lista os departamentos cadastrados no sistema.

Precondição: O Coordenador do curso está autenticado.

Fluxo Principal:

1. Coordenador do curso solicita listar os departamentos cadastrados no sistema.
2. Sistema exibe o nome e a quantidade de salas de cada departamento cadastrado.
3. Coordenador do curso fica ciente dos departamentos cadastrados.
4. Caso de Uso termina.

Pós-Condição: Os dados do departamento foram listados pelo sistema.

Cadastrar Professor (CSU05)

Ator primário: Coordenador do curso.

Descrição: Coordenador do curso realiza o cadastro de um professor no sistema.

Precondições: O Coordenador do curso está autenticado no sistema.

Fluxo Principal:

1. Coordenador do curso seleciona a opção de cadastrar um novo professor.
2. Sistema solicita os dados do professor (nome, login, senha e confirmação da senha).
3. Coordenador do curso fornece os dados solicitados.
4. Coordenador do curso clica em "cadastrar".

5. Sistema realiza o cadastro do professor.
6. Sistema informa ao coordenador do curso que o professor foi cadastrado com sucesso.
7. Caso de Uso termina.

Fluxo de Exceção (Passo 4): Senhas informadas divergentes

1. Sistema detecta que o coordenador do curso forneceu a senha e a sua confirmação diferentes.
2. Sistema informa ao coordenador do curso que as senhas são diferentes.
3. Coordenador do curso atualiza as senhas e o Caso de Uso continua a partir do passo 4.

Pós-Condição: Os dados do professor foram cadastrados no sistema.

Remover Professor (CSU06)

Ator Primário: Coordenador do curso.

Descrição: Coordenador do curso remove um professor do sistema.

Precondição: O Coordenador do curso está autenticado e o professor se encontra cadastrado no sistema.

Fluxo Principal:

1. Coordenador do curso solicita a remoção de um professor.
2. Sistema solicita o nome do professor.
3. Coordenador do curso fornece o nome do professor.
4. Coordenador do curso clica em "Remover".
5. Sistema pede que coordenador do curso confirme a remoção do professor.
6. Coordenador do curso confirma a remoção.
7. Sistema verifica se existe turma associada ao professor.
8. Sistema remove o professor.
9. Sistema informa ao coordenador do curso que o professor foi removido com sucesso.
10. Coordenador do curso fica ciente da remoção.
11. Caso de Uso termina.

Fluxo Alternativo (Passo : 5) Confirmação da remoção

1. Coordenador do curso não confirma a remoção do professor.
2. Sistema não remove os dados do professor.
3. Caso de Uso continua a partir do passo 2.

Fluxo de Exceção (Passo 3): Escolha do professor errado

1. Coordenador do curso detecta que informou o nome do professor errado.
2. Sistema possibilita que o coordenador do curso informe o nome do professor.
3. Coordenador do curso informa o nome do professor correto.
4. Sistema aceita a correção e o Caso de Uso continua a partir do passo 2.

Fluxo de Alternativo (Passo 7): Existe turma associada ao professor.

1. Sistema informa que existe turma associada ao professor.
2. Coordenador do curso fica ciente que existe turma associada ao professor.
3. Sistema não permite a remoção do professor e o Caso de Uso continua a partir do passo 2.

Fluxo de Alternativo (Passo 7): Nenhuma turma associada ao professor.

1. Sistema verifica que não existe turma associada ao professor.
2. Caso de Uso continua a partir do passo 8.

Pós-Condição: Os dados do professor foram removidos do sistema.

Listar Professores (CSU07)

Ator Primário: Coordenador do curso.

Descrição: Coordenador do curso lista os professores cadastrados no sistema.

Precondição: O Coordenador do curso está autenticado.

Fluxo Principal:

1. Coordenador do curso solicita listar os professores cadastrados no sistema.
2. Sistema exibe o nome de cada professor cadastrado.
3. Coordenador do curso fica ciente dos professores cadastrados.
4. Caso de Uso termina.

Pós-Condição: Os nomes dos professores cadastrados foram listados pelo sistema.

Atualizar Grupo (CSU08)

Ator Primário: Coordenador do curso.

Descrição: Coordenador do curso atualiza os dados de um grupo no sistema.

Precondição: O Coordenador do curso está autenticado e o grupo se encontra cadastrado no sistema.

Fluxo Principal:

1. Coordenador do curso solicita atualizar os dados de um grupo.

2. Sistema solicita o nome do grupo a ser atualizado.
3. Coordenador do curso fornece o nome do grupo.
4. Sistema exibe os dados (máximo aulas dias, penalidade excesso aulas dia, penalidade conflito horários, penalidades de tamanho de buracos nos dias, penalidade de alocar turma num horário e os horários censurados) para serem atualizados.
5. Coordenador do curso atualiza os dados do grupo.
6. Coordenador do curso clica em "Atualizar".
7. Sistema solicita a confirmação ao coordenador do curso para a atualização.
8. Coordenador do curso confirma a atualização.
9. Sistema atualiza os dados do grupo.
10. Sistema informa ao coordenador do curso que o grupo foi atualizado com sucesso.
11. Caso de Uso termina.

Fluxo Alternativo (Passo : 7) Confirmação da atualização

1. Coordenador do curso não confirma a atualização do grupo.
2. Sistema não atualiza os dados do grupo.
3. Caso de Uso continua a partir do passo 2.

Fluxo de Exceção (Passo 3): Escolha do grupo errado

1. Coordenador do curso detecta que informou o nome do grupo errado.
2. Sistema possibilita que o coordenador do curso informe o nome do grupo.
3. Coordenador do curso informa o nome do grupo correto.
4. Sistema aceita a correção e o Caso de Uso continua a partir do passo 2.

Fluxo de Exceção (Passo 5): Modificação de dados errada

1. Coordenador do curso detecta que fez a modificação errada.
2. Coordenador do curso corrige a informação que foi alterada de forma incorreta.
3. Sistema aceita a correção e o Caso de Uso continua a partir do passo 5.

Pós-Condição: Os dados do grupo foram atualizados no sistema.

Cadastrar Grupo (CSU09)

Ator primário: Coordenador do curso.

Descrição: Coordenador do curso cadastra os dados de um novo grupo no sistema.

Precondições: O Coordenador do curso está autenticado no sistema.

Fluxo Principal:

1. Coordenador do curso seleciona a opção de cadastrar um novo grupo.
2. Sistema solicita os dados do grupo (nome, máximo aulas dias, penalidade excesso aulas dia, penalidade conflito horários, penalidades de tamanho de buracos nos dias, penalidade de alocar Turma num horário e os horários censurados).
3. Coordenador do curso fornece os dados solicitados.
4. Coordenador do curso clica em "Cadastrar".
5. Sistema registra os dados fornecidos do grupo.
6. Sistema informa ao coordenador do curso que o grupo foi cadastrado com sucesso.
7. Caso de Uso termina.

Fluxo de Exceção (Passo 3): Inserção do dado errada

1. Coordenador do curso detecta que fez a modificação errada.
2. Coordenador do curso corrige a informação que foi alterada de forma incorreta.
3. Sistema aceita a correção e o Caso de Uso continua a partir do passo 3.

Pós-Condição: Os dados do grupo foram cadastrados no sistema com sucesso.

Remover Grupo (CSU10)

Ator Primário: Coordenador do curso.

Descrição: Coordenador do curso remove um grupo do sistema.

Precondição: O coordenador do curso está autenticado e o grupo se encontra cadastrado no sistema.

Fluxo Principal:

1. Coordenador do curso solicita a remoção de um grupo.
2. Sistema solicita o nome do grupo.
3. Coordenador do curso fornece o nome do grupo.
4. Coordenador do curso clica em "Remover".
5. Sistema solicita a confirmação da remoção do grupo.
6. Coordenador do curso confirma a remoção.
7. Sistema verifica se existe turma associada ao grupo.

8. Sistema remove o grupo.
9. Sistema informa ao coordenador do curso que o grupo foi removido com sucesso.
10. Coordenador do curso fica ciente da remoção.
11. Caso de Uso termina.

Fluxo de Exceção (Passo 3): Escolha do grupo errado

1. Coordenador do curso detecta que informou o nome do grupo errado.
2. Sistema possibilita que o coordenador do curso forneça o nome do grupo.
3. Coordenador do curso informa o nome do grupo correto.
4. Sistema aceita a correção e o Caso de Uso continua a partir do passo 2.

Fluxo de Alternativo (Passo 7): Existe turma associada ao grupo.

1. Sistema informa que existe turma associada ao grupo.
2. Coordenador do curso fica ciente que existe turma associada ao grupo.
3. Sistema não permite a remoção do grupo e o Caso de Uso continua a partir do passo 2.

Fluxo de Alternativo (Passo 7): Nenhuma turma associada ao grupo.

1. Sistema verifica que não existe turma associada ao grupo.
2. Caso de Uso continua a partir do passo 8.

Fluxo Alternativo (Passo : 7) Confirmação de remoção

1. Coordenador do curso não confirma a remoção do grupo.
2. Sistema não remove os dados do grupo.
3. Caso de Uso continua a partir do passo 2.

Pós-Condição: Os dados do grupo foram removidos do sistema.

Listar Grupos (CSU11)

Ator Primário: Coordenador do curso.

Descrição: Coordenador do curso lista os grupos cadastrados no sistema.

Precondição: O coordenador do curso está autenticado.

Fluxo Principal:

1. Coordenador do curso solicita listar os grupos cadastrados no sistema.
2. Sistema exibe o nome e o máximo de aulas por dia de cada grupo cadastrado.

3. Coordenador do curso fica ciente dos grupos cadastrados.
4. Caso de Uso termina.

Pós-Condição: Os dados dos grupos cadastrados foram listados pelo sistema.

Atualizar Turma (CSU12)

Ator Primário: Coordenador do curso.

Descrição: Coordenador do curso atualiza os dados de uma turma no sistema.

Precondição: O Coordenador do curso está autenticado e a turma encontra-se cadastrada no sistema.

Fluxo Principal:

1. Coordenador do curso solicita atualizar os dados de uma turma.
2. Sistema solicita o nome da disciplina, código da turma e o código da disciplina.
3. Coordenador do curso fornece o nome da disciplina, código da turma e o código da disciplina que deseja atualizar.
4. Sistema exibe os dados (período, carga horária, máximo aulas mesmo dia, nome do grupo, nome do professor, nome do grupo, nome do departamento e os horários obrigatórios) para serem modificados.
5. Coordenador do curso modifica os dados da turma.
6. Coordenador do curso clica em "Atualizar".
7. Sistema solicita confirmação da atualização da turma para o coordenador do curso.
8. Coordenador do curso confirma atualização da turma.
9. Sistema atualiza os dados da turma.
10. Sistema informa ao coordenador do curso que a turma foi atualizada com sucesso.
11. Coordenador fica ciente da atualização.
12. Caso de Uso termina.

Fluxo de Exceção (Passo 3): Escolha da turma errada

1. Coordenador do curso detecta que informou o nome da disciplina, código da turma e o código da disciplina errados.
2. Sistema possibilita que o coordenador do curso informe o nome da disciplina, código da turma e o código da disciplina corretos.
3. Coordenador do curso informa o nome da disciplina, código da turma e o código da disciplina.
4. Sistema aceita a correção e o Caso de Uso continua a partir do passo 2.

Fluxo de Exceção (Passo 5): Modificação de dados errada

1. Coordenador do curso detecta que fez a modificação errada.
2. Coordenador do curso corrige a informação que foi modificada de forma incorreta.
3. Sistema aceita a correção e o Caso de Uso continua a partir do passo 6.

Fluxo Alternativo (Passo : 7) Confirmação da atualização

1. Coordenador do curso não confirma a atualização da turma.
2. Sistema não atualiza os dados da turma.
3. Caso de Uso continua a partir do passo 2.

Pós-Condição: Os dados da turma foram atualizados no sistema.

Cadastrar Turma (CSU13)

Ator primário: Coordenador do curso.

Descrição: Coordenador do curso cadastra os dados de uma nova turma no sistema.

Precondições: O coordenador do curso está autenticado no sistema.

Fluxo Principal:

1. Coordenador do curso seleciona a opção de cadastrar uma nova turma.
2. Sistema solicita os dados da turma (nome da disciplina, código da turma, código da disciplina, período, carga horária, máximo aulas mesmo dia, nome do grupo, nome do professor, nome do departamento e os horários obrigatórios).
3. O Coordenador do curso fornece os dados solicitados.
4. O Coordenador do curso clica em "Cadastrar".
5. Sistema registra os dados fornecidos.
6. Sistema informa ao coordenador do curso que a turma foi cadastrada com sucesso.
7. Caso de Uso termina.

Fluxo de Exceção (Passo 4): Inserção de dado errada

1. Coordenador do curso detecta que forneceu algum dado errado da turma.
2. Sistema permite que o dado seja corrigido.
3. Coordenador do curso corrige o dado que foi fornecido de forma incorreta.
4. Sistema aceita a correção e o Caso de Uso continua a partir do passo 3.

Pós-Condição: Os dados da turma foram cadastrados no sistema.

Remover Turma (CSU14)

Ator Primário: Coordenador do curso.

Descrição: Coordenador do curso remove uma turma do sistema.

Precondição: O coordenador do curso está autenticado e a turma se encontra cadastrada no sistema.

Fluxo Principal:

1. Coordenador do curso solicita a remoção de uma turma.
2. Sistema solicita o nome, código da turma e o código da disciplina da turma.
3. Coordenador do curso fornece o nome, código da turma e o código da disciplina.
4. Coordenador do curso clica em "Remover".
5. Sistema solicita confirmação do coordenador do curso para remover a turma.
6. Coordenador do curso confirma a remoção.
7. Sistema remove a turma.
8. Sistema informa ao coordenador do curso que os dados da turma foram removidos com sucesso.
9. Coordenador do curso fica ciente da remoção.
10. Caso de Uso termina.

Fluxo de Exceção (Passo 3): Escolha da turma errada

1. Coordenador do curso detecta que informou o nome, código da turma e o código da disciplina errados.
2. Sistema possibilita que o coordenador do curso corrija o nome, código da turma e o código da disciplina.
3. Coordenador do curso corrige as informações que foram inseridas de forma incorreta.
4. Sistema aceita a correção e o Caso de Uso continua a partir do passo 2.

Fluxo Alternativo (Passo : 5) Confirmação de remoção

1. Coordenador do curso não confirma a remoção da turma.
2. Sistema não remove os dados da turma.
3. Caso de Uso continua a partir do passo 2.

Pós-Condição: Os dados da turma foram removidos do sistema.

Listar Turmas (CSU15)

Ator Primário: Coordenador do curso.

Descrição: Coordenador do curso lista as turmas cadastradas no sistema.

Precondição: O Coordenador do curso está autenticado.

Fluxo Principal:

1. Coordenador do curso solicita listar as turmas cadastradas no sistema.
2. Sistema exibe o nome e o código da turma de cada turma cadastrada.
3. Coordenador do curso fica ciente das turmas cadastradas.
4. Caso de Uso termina.

Pós-Condição: Os dados das turmas cadastradas foram listados pelo sistema.

Cadastrar Coordenador (CSU16)

Ator primário: Coordenador do curso.

Descrição: Coordenador do curso cadastra os dados de um novo coordenador no sistema.

Precondições: O coordenador do curso está autenticado no sistema.

Fluxo Principal:

1. Coordenador do curso seleciona a opção de cadastrar um novo coordenador.
2. Sistema solicita os dados do coordenador (nome, login, senha e a confirmação da senha).
3. O coordenador do curso fornece os dados solicitados.
4. O coordenador do curso clica em "Cadastrar".
5. Sistema registra os dados fornecidos.
6. Sistema informa ao coordenador do curso que o coordenador foi cadastrado com sucesso.
7. Caso de Uso termina.

Fluxo de Exceção (Passo 4): Inserção de dado errada

1. Coordenador do curso detecta que forneceu algum dado errado do novo coordenador.
2. Sistema permite que o dado seja corrigido.
3. Coordenador do curso corrige o dado que foi fornecido de forma incorreta.
4. Sistema aceita a correção e o Caso de Uso continua a partir do passo 3.

Pós-Condição: Os dados do novo coordenador foram cadastrados no sistema.

Remover Coordenador (CSU17)

Ator Primário: Coordenador do curso.

Descrição: Coordenador do curso remove um coordenador do sistema.

Precondição: O coordenador do curso está autenticado e o coordenador se encontra cadastrado no sistema.

Fluxo Principal:

1. Coordenador do curso solicita a remoção de um coordenador.
2. Sistema solicita o nome e o login do coordenador para ser removido.
3. Coordenador do curso fornece o nome e o login.
4. Coordenador do curso clica em "Remover".
5. Sistema solicita confirmação ao coordenador do curso para remover o coordenador.
6. Coordenador do curso confirma a remoção.
7. Sistema remove o coordenador.
8. Sistema informa ao coordenador do curso que os dados do coordenador foram removidos com sucesso.
9. Coordenador do curso fica ciente da remoção.
10. Caso de Uso termina.

Fluxo de Exceção (Passo 3): Escolha do coordenador errado

1. Coordenador do curso detecta que informou o nome e o login errados.
2. Sistema possibilita que o coordenador do curso corrija o nome e o login.
3. Coordenador do curso corrige as informações que foram inseridas de forma incorreta.
4. Sistema aceita a correção e o Caso de Uso continua a partir do passo 2.

Fluxo Alternativo (Passo : 5) Confirmação de remoção

1. Coordenador do curso não confirma a remoção do coordenador.
2. Sistema não remove os dados do coordenador.
3. Caso de Uso continua a partir do passo 2.

Pós-Condição: Os dados do coordenador foram removidos do sistema.

Atualizar senha Coordenador (CSU18)

Ator Primário: Coordenador do curso.

Descrição: Coordenador do curso atualiza a senha no sistema.

Precondição: O coordenador do curso está autenticado.

Fluxo Principal:

1. Coordenador do curso solicita atualizar a senha.
2. Sistema solicita a senha atual, nova senha e a confirmação da nova senha.
3. Coordenador do curso fornece a senha atual, nova senha e a confirmação da nova senha.
4. Coordenador do curso clica em "Atualizar".
5. Sistema solicita confirmação da atualização da senha.
6. Coordenador do curso confirma a atualização.
7. Sistema verifica se a atual senha foi informada corretamente.
8. Sistema verifica se a nova senha informada e a sua confirmação são iguais.
9. Sistema atualiza a senha do coordenador do curso.
10. Sistema informa ao coordenador do curso que a senha foi atualizada com sucesso.
11. Coordenador do curso fica ciente da atualização.
12. Caso de Uso termina.

Fluxo de Exceção (Passo 5): Inserção de dados errada

1. Coordenador do curso detecta que forneceu a senha errada.
2. Coordenador do curso corrige a informação que foi fornecida de forma incorreta.
3. Sistema aceita a correção e o Caso de Uso continua a partir do passo 6.

Fluxo de Exceção (Passo 8): Verificação de senhas iguais

1. Sistema detecta que o coordenador do curso forneceu a nova senha e a sua confirmação diferentes.
2. Sistema informa ao coordenador do curso que a nova senha e a sua confirmação são diferentes.
3. Coordenador do curso corrige a informação que foi fornecida de forma incorreta.
4. Sistema aceita a correção e o Caso de Uso continua a partir do passo 4.

Fluxo de Exceção (Passo 7): Verificação da atual senha

1. Sistema detecta que o coordenador do curso forneceu a atual senha errada.
2. Sistema informa ao coordenador do curso que a atual senha está errada.
3. Coordenador do curso corrige a informação que foi fornecida de forma incorreta.
4. Sistema aceita a correção e o Caso de Uso continua a partir do passo 4.

Fluxo Alternativo (Passo : 5) Confirmação da atualização

1. Coordenador do curso não confirma a atualização da senha.

2. Sistema não atualiza a senha do coordenador do curso.
3. Caso de Uso continua a partir do passo 2.

Pós-Condição: A senha do coordenador do curso foi atualizada no sistema.

Atualizar Preferências (CSU19)

Ator Primário: Professor.

Descrição: Professor atualiza as preferências no sistema.

Precondição: O professor está autenticado sistema.

Fluxo Principal:

1. Professor solicita atualizar as preferências no sistema.
2. Sistema exibe as preferências (máximo de aulas por dia, máximo de dias na semana, penalidade excesso aulas dia, penalidade excesso dias semana, penalidades buraco semanas, penalidades buraco dias, penalidade horários) para serem atualizadas.
3. Professor atualiza as suas preferências.
4. Professor clica em "Atualizar".
5. Sistema solicita confirmação de atualização ao professor.
6. Professor confirma atualização das preferências.
7. Sistema atualiza as preferências do professor.
8. Sistema informa o professor que as preferências foram atualizadas com sucesso.
9. Professor fica ciente da atualização das preferências.
10. Caso de Uso termina.

Fluxo de Exceção (Passo 3): Atualização de preferência errada

1. Professor detecta que fez a modificação errada.
2. Professor corrige a informação que foi modificada de forma incorreta.
3. Sistema aceita a correção e o Caso de Uso continua a partir do passo 3.

Fluxo Alternativo (Passo 5): Confirmação da atualização

1. Professor não confirma a atualização das preferências.
2. Sistema não atualiza as preferências do professor.
3. Caso de Uso continua a partir do passo 2.

Pós-Condição: As preferências do professor foram atualizadas no sistema.

Cadastrar Preferências (CSU20)

Ator Primário: Professor.

Descrição: Professor cadastra as preferências no sistema.

Precondição: O professor está autenticado sistema.

Fluxo Principal:

1. Professor solicita cadastrar as preferências no sistema.
2. Sistema solicita as preferências (máximo de aulas por dia, máximo de dias na semana, penalidade excesso aulas dia, penalidade excesso dias semana, penalidades buraco semanas, penalidades buraco dias, penalidade horários) do professor para serem cadastradas.
3. O professor fornece as preferências solicitadas.
4. Professor clica em "Cadastrar".
5. Sistema registra as preferências fornecidas.
6. Sistema informa ao professor que as preferências foram cadastradas com sucesso.
7. Professor fica ciente que as preferências foram cadastradas.
8. Caso de Uso termina.

Fluxo Alternativo (Passo 3): Inserção de preferência errada

1. Professor detecta que forneceu alguma preferência errada.
2. Sistema permite que a preferência seja corrigida.
3. Professor corrige a preferência.
4. Sistema aceita a correção e o Caso de Uso continua a partir do passo 3.

Pós-Condição: As preferências do professor foram cadastradas no sistema.

Gerar Solucao (CSU21)

Ator primário: Coordenador do curso.

Descrição: Coordenador do curso gera a solução pelo sistema.

Precondições: O coordenador do curso está autenticado e todos os dados foram cadastrados no sistema.

Fluxo Principal:

1. Coordenador do curso solicita gerar solução.
2. Sistema solicita que o coordenador do curso informe a quantidade de soluções para serem geradas.
3. Coordenador do curso fornece a quantidade de soluções que deseja.
4. Coordenador do curso clica em "OK".

5. Sistema gera a quantidade informada de soluções de Quadro de Horários.
6. Sistema informa ao coordenador do curso que as soluções foram geradas.
7. Coordenador do curso fica ciente.
8. Caso de Uso termina.

Fluxo de Exceção (Passo 3): Inserção de quantidade errada

1. Coordenador do curso detecta que forneceu a quantidade de soluções errada.
2. Sistema permite que a quantidade seja corrigida.
3. Coordenador do curso corrige a quantidade que foi fornecida de forma errada.
4. Sistema aceita a correção e o Caso de Uso continua a partir do passo 3.

Pós-Condição: Os quadro de horários foram gerados.

Consultar Solucao (CSU22)

Ator primário: Coordenador do curso.

Descrição: Coordenador do curso consulta a solução pelo sistema.

Precondições: O coordenador do curso está autenticado e a solução se encontra gerada no sistema.

Fluxo Principal:

1. Coordenador do curso solicita de consultar uma solução.
2. Sistema solicita que o coordenador do curso informe o nome da solução que deseja consultar.
3. Coordenador do curso fornece o nome da solução.
4. Sistema exibe o quadro de horários (nomes das disciplinas, turmas, professores, horários e o custo da solução) para o nome da solução fornecido.
5. Coordenador do curso fica ciente da solução gerada.
6. Caso de Uso termina.

Fluxo de Exceção (Passo 3): Escolha da solução errada

1. Coordenador do curso detecta que forneceu o nome da solução errada.
2. Sistema permite que o nome seja corrigido.
3. Coordenador do curso corrige o nome da solução desejada para consultar.
4. Sistema aceita a correção e o Caso de Uso continua a partir do passo 3.

Pós-Condição: O quadro de horários foi consultado pelo coordenador do curso.

Listar Solucoes (CSU23)

Ator Primário: Coordenador do curso.

Descrição: Coordenador do curso lista as Soluções geradas no sistema.

Precondição: O coordenador do curso está autenticado.

Fluxo Principal:

1. Coordenador do curso solicita listar as Soluções geradas no sistema.
2. Sistema exibe o nome e o custo de cada solução gerada.
3. Coordenador do curso fica ciente das Soluções geradas.
4. Caso de Uso termina.

Pós-Condição: Os dados das Soluções geradas foram listadas pelo sistema.

Remover Solucao (CSU24)

Ator Primário: Coordenador do curso.

Descrição: Coordenador do curso remove uma solução do sistema.

Precondição: O Coordenador do curso está autenticado e a solução se encontra gerada no sistema.

Fluxo Principal:

1. Coordenador do curso solicita a remoção de uma solução.
2. Sistema solicita o nome da solução para ser removida.
3. Coordenador do curso fornece o nome.
4. Coordenador do curso clica em "Remover".
5. Sistema solicita confirmação ao coordenador do curso para remover a solução.
6. Coordenador do curso confirma a remoção.
7. Sistema remove a solução.
8. Sistema informa ao coordenador do curso que os dados da solução foram removidos com sucesso.
9. Coordenador do curso fica ciente da remoção.
10. Caso de Uso termina.

Fluxo de Exceção (Passo 3): Escolha da solução errada

1. Coordenador do curso detecta que informou o nome errado.
2. Sistema possibilita que o coordenador do curso corrija o nome.
3. Coordenador do curso corrige o nome.
4. Sistema aceita a correção e o Caso de Uso continua a partir do passo 2.

Fluxo Alternativo (Passo : 5) Confirmação de remoção

1. Coordenador do curso não confirma a remoção da solução.
2. Sistema não remove os dados da solução.
3. Caso de Uso continua a partir do passo 2.

Pós-Condição: Os dados da solução foram removidos do sistema.

Exportar Solucao (CSU25)

Ator primário: Coordenador do curso.

Descrição: Coordenador do curso exporta a solução gerada pelo sistema.

Precondições: O coordenador do curso está autenticado e a solução foi gerada pelo sistema.

Fluxo Principal:

1. Coordenador do curso solicita exportar a solução.
2. Sistema exporta a solução em PDF.
3. Sistema informa ao coordenador do curso que a solução foi exportada com sucesso.
4. Coordenador do curso fica ciente da exportação da solução.
5. Caso de Uso termina.

Pós-Condição: A solução é exportada pelo coordenador do curso.