



Universidade Federal de Ouro Preto  
Instituto de Ciências Exatas e Aplicadas  
Departamento de Computação e Sistemas

# ALOCAÇÃO DE PROFESSORES ÀS DISCIPLINAS

Relatório Final, referente ao período (Março de 2018 a  
Março de 2019), apresentado à Universidade Federal de  
Ouro Preto, como parte das exigências do programa de  
iniciação científica - FAPEMIG

**Bolsista:** Felipe Augusto Magno Tomaz

**Orientador:** Prof. Dr. George Henrique Godim da  
Fonseca

João Monlevade - MG  
13 de janeiro de 2020

# Resumo

Este trabalho pode ser classificado como um problema de Atribuição de Tarefas, tal que um conjunto de tarefas é atribuído a um conjunto de recursos tal que a compatibilidade entre estes seja máxima. O problema da Atribuição de tarefas, também conhecido na literatura como Assignment Problem, é muito estudado na área de Otimização Combinatória. O projeto busca resolver, em específico, o Problema de Alocação de Disciplinas a Professores em instituições de ensino. O método adotado para resolução do problema foi a geração de uma solução inicial com um modelo de emparelhamento em grafos e, em seguida, utilizar algoritmos heurísticos de busca local para melhorar a solução obtida. Além disso, foi desenvolvida uma interface gráfica para interação com o usuário, tornando o processo de entrada e visualização de resultados mais agradável e intuitiva. Por fim, utilizamos dados reais do Departamento de Computação e Sistemas (DECSI) da Universidade Federal de Ouro Preto (UFOP) para validar e avaliar o desempenho do modelo de otimização desenvolvido.

**Palavras-chave:** Otimização, Timetabling, Programação Linear, Programação Linear Inteira, Assignment Problem.

# Abstract

This project can be classified as a Task Assignment problem, such that a set of tasks is assigned to a set of resources where the compatibility between them is maximum. The Problem of Assignment of Tasks, also known in the literature as Assignment Problem, is much studied in the area of Combinatorial Optimization. This project seeks to solve, in specific, the Problem of Allocation of Disciplines to Teachers in educational institutions. The method used to solve the problem was to generate an initial solution with a pairing model in graphs and then to use local search heuristic algorithms to improve the solution obtained. In addition, a graphical interface was developed for user interaction, making the process of entry and visualization of results more pleasant and intuitive. Finally, we used real data from the Department of Computing and Systems (DECSI) of the Federal University of Ouro Preto (UFOP) to validate and evaluate the performance of the optimization model developed.

**Keywords:** Optimization, Timetabling, Linear Programming, Integer Linear Programming, Assignment Problem.

# 1 Introdução

O problema da Atribuição de Tarefas é um problema vastamente estudado, sendo tema de diversos estudos e trabalhos. Nas instituições de ensino este problema aparece na formulação da grade horária, sendo relatado na literatura também como timetabling, onde cada professor é alocado a uma ou mais disciplinas.

Neste trabalho foi utilizado um modelo matemático de programação inteira, que recebe como dados de entrada um conjunto de arquivos no formato CSV (Comma Separated Values). Por se tratarem de arquivos puramente textuais, a manipulação desses arquivos pode se tornar inviável, principalmente por pessoas que desconhecem o modelo utilizado e suas características. Diante deste cenário, o trabalho aqui desenvolvido se propõe ao desenvolvimento de um software para tornar essa tarefa mais natural e intuitiva, possibilitando a implementação do modelo em cenários reais.

## 2 Objetivos

O objetivo do presente trabalho é apresentar uma ferramenta que permita a manipulação dos dados de entrada de um modelo matemático de agendamento de horários através de uma interface gráfica de fácil utilização.

## 3 Revisão Bibliográfica

Problemas de Atribuição de Tarefas tendem a ser intratáveis sem o auxílio de modelos computacionais, especialmente quando o número de restrições e recursos a serem alocados é grande. O agendamento de horário deve considerar a disponibilidade dos professores, disciplinas que compartilham horário não podem ser atribuídas a um mesmo professor, a carga horária de cada professor deve ser respeitada, etc.

Como se trata de um problema de otimização combinatória, quanto maior o número de professores e disciplinas maior é o número de soluções a serem avaliadas, tornando a elaboração de horários uma tarefa árdua mesmo em instituições de pequeno porte (PREIS, 2007). Se faz necessário portanto, a utilização de modelos computacionais para auxiliar nesse processo, alcançando soluções melhores em menor tempo.

Como todo problema de alocação de tarefas a recursos, problemas de timetabling possuem restrições invioláveis, ou seja, caso essas restrições sejam violadas a solução encontrada não é uma solução para o problema. Há também as chamadas restrições fracas ou violáveis, que são restrições que se violadas não invalidam a solução. Em geral, para cada restrição fraca presente na formulação é atribuído um valor peso, que em caso de violação é somado à solução final. O objetivo principal é gerar uma tabela de horários que minimize ou elimine conflitos, ou seja, possua o menor peso associado (ARROYO et al., 2002)

O número e o tipo das restrições impostas estão intimamente ligados ao modelo educacional adotado, à instituição em que o modelo será aplicado, leis trabalhistas em vigor, etc. Assim, não é possível a construção de um modelo genérico o bastante para elaborar tabelas de horário para qualquer instituição de ensino (NETO; VIANNA, 2013)

Diversos modelos e algoritmos para solução dos problemas de timetabling foram propostos, como a resolução através de Algoritmos Genéticos, Scatter Search, Tabu Search, algoritmos gulosos, Coloração de Grafos, etc, e até mesmo combinações destes, cada qual com suas próprias particularidades, não sendo possível dizer se um é melhor ou pior que os demais. (PREIS, 2007).

Neste trabalho optou-se por se utilizar um modelo matemático em Programação Inteira Binária, em que todas as variáveis envolvidas na formulação do problema assumem apenas os valores 0(zero) ou 1(um). Esse tipo particular de formulação da Programação Inteira é empregado principalmente quando se tem cenários em que as restrições e soluções podem ser classificadas como válidas ou inválidas, ou seja, assumem valores verdadeiro ou falso(ALVES; DELGADO, 1997). A aplicação desse modelo é perfeitamente aplicável a situação de agendamento de horários, uma vez que podemos classificar as soluções como válidas caso não haja violação de restrições fortes, e inválidas caso contrário.

Uma característica comum às soluções propostas na literatura é a utilização de heurísticas e meta-heurísticas para limitação do espaço de busca. Isso se faz necessário devido ao grande número de soluções possíveis. Caso não houvesse um limite, o tempo de processamento seria extremamente alto, inviabilizando a utilização dessas ferramentas. Deve-se salientar que, ao se impor um limite de tempo e/ou um número máximo de soluções avaliadas, não se pode garantir a obtenção de resultados finais iguais, tampouco a solução ótima para o dado problema (CAMPOS, 2002)

## 4 Materiais e Métodos

Nessa seção são apresentados as entidades envolvidas no agendamento de horários e as restrições a elas impostas, o modelo matemático utilizado para resolução e o formato dos dados de entrada deste modelo. Além disso, são apresentadas as ferramentas de desenvolvimento utilizadas no projeto.

### 4.1 Entidades

Para resolver o problema de alocação de professores em disciplinas formulou-se quatro entidades:

- *Professor* (Professor)
- *Class* (Turma)
- *Event* (Disciplina)
- *Time* (Tempo)

#### 4.1.1 Professor

A entidade Professor presente no software é a representação dos professores da instituição. Cada professor possui os seguintes atributos:

- **Name.** Nome do professor. Ex.: Neil deGrasse Tyson
- **Tag.** É o que possibilita a atribuição de uma tag sobre o professor. Ex: *DEPCOMP*.
- **Priority.** É um multiplicador para as penalidades de um determinado professor. Quanto maior o Priority, mais crítico e o não atendimento de suas restrições.
- **Preferred number of working days.** É o número mínimo preferível de dias trabalhados semanalmente.
- **Preferred number of working days.** É o número máximo preferível de dias trabalhados semanalmente.
- **Unavailabilities.** São os horários que são indesejados ou indisponíveis para a alocação desse professor. Ex.: *Horário indesejável - Quarta-feira, 15:30 - 17:10.*
- **Undesired Patterns.** São os padrões de folga/dias trabalhados indesejados para este professor. Ex. *Dias trabalhados: Segunda, Quarta, Sexta - Folgas: Terça, Quinta.*



### 4.1.2 Class

A entidade Class (*Classe*) presente no software é a representação das classes da instituição. Cada classe possui os seguintes atributos:

- **Name.** Nome da classe. *Ex.: Engenharia de Minas - 1º Período.*
- **Unavailabilities.** São os horários que são indisponíveis para a alocação dessa classe. *Ex.: Horário indisponível - Sexta-feira, 18:50 - 20:30.*

### 4.1.3 Event

A entidade Event (*Evento*) é a representação de um encontro (*aula*) de uma determinada disciplina. Cada evento possui como atributos os seguintes campos:

- **Subject.** Representa o assunto/disciplina do evento. *Ex.: Cálculo II.*
- **Class.** A classe a que este evento é associado. *Ex.: Eng. Computação - 2º Período.*
- **Compatible Professors.** É o conjunto de professores que estão aptos a lecionar a determinada disciplina. A cada professor é associado um valor peso, de modo que quanto maior o peso pior é alocar esse professor à essa disciplina.
- **Tag.** É o que possibilita a atribuição de uma tag sobre o evento. *Ex: DECSI.*
- **Duration.** Representa o número de encontros semanais que devem ser alocados para esse evento.
- **Split.** É o modo como serão distribuídas o evento durante a semana. *Ex.: O evento que possui 8 horas semanais será dividido em 2 encontros de 4 horas cada. Então Split = 4 + 4.*
- **Minimum gap between meetings.** Indica o número mínimo de dias entre uma reunião e outra.
- **Maximum gap between meetings.** Indica o número máximo de dias entre uma reunião e outra.
- **Avoid Clashes.** Deve-se alocar os recursos sem que haja conflitos, ou seja, sem alocar o mesmo recurso a mais de um evento ao mesmo tempo;
- **Limit Workload.** A cada evento está associada uma carga de trabalho. Essa carga de trabalho é somada a todos os recursos que forem alocados a esse evento. Isso evita que os professores tenham suas cargas de trabalho ultrapassadas.

## 4.2 Modelo Matemático

O modelo matemático utilizado para resolução do problema é uma formulação em programação inteira binária, onde as variáveis assumem os valores 0(zero) ou 1(um). Para encontrar uma solução válida o modelo atua sobre uma série variáveis, variáveis essas que serão avaliadas através de uma função objetivo.

### 4.2.1 Conjuntos e Parâmetros

O modelo tem como conjuntos de entrada:

- $E$  - Eventos
- $P$  - Professores
- $S$  - Disciplinas
- $C$  - Classes
- $D$  - Dias
- $T$  - Tempos/Horários
- $CPE_e$  - Professores compatíveis para o evento  $e$
- $UC_c$  - Horários indisponíveis para a classe  $c$
- $UP_p$  - Horários indisponíveis para o professor  $p$
- $NP_p$  - Horários indesejados para o professor  $p$
- $UP$  - Padrões indesejado de dias folga/dias trabalhados
- $UP_p$  - Padrões indesejado de dias folga/dias trabalhados para o professor  $p$
- $UPB_{up}$  - Dias trabalhados em padrão indesejado
- $UPF_{up}$  - Dias de folga em padrão indesejado

O modelo tem como parâmetros de entrada:

- $d_e$  - Duração do evento  $e$
- $mingap_e$  -  $N^\circ$  mínimo de dias entre os encontros do evento  $e$
- $maxgap_e$  -  $N^\circ$  máximo de dias entre os encontros do evento  $e$
- $dur_p$  - Duração máxima dos eventos que podem ser atribuídos ao professor  $p$

- $mindays_p$  - Número mínimo desejado de dias trabalhados pelo professor  $p$
- $maxdays_p$  - Número máximo desejado de dias trabalhados pelo professor  $p$
- $priority_p$  - Multiplicador de função objetivo para o professor  $p$
- $costEP_{p,e}$  - Custo de atribuição do professor  $p$  ao evento  $e$  e pertencente ao  $CPE$
- $w^D$  - Peso de cada violação do número desejado de dias trabalhados pelo professor
- $w^{UT}$  - Peso de cada atribuição de aula em horário indesejado pelo professor
- $w^{UP}$  - Peso por atribuição que segue um padrão indesejado de dias de folga/trabalho.
- $w^{EP}$  - Peso do multiplicador de violações ocorridas pela atribuição do professor ao evento.

### 4.2.2 Variáveis

A principal variável do modelo é a  $x_{s,c,p,d,t}$ . Essa variável é definida como:

$$x_{s,c,p,d,t} = \begin{cases} 1 & \text{se disciplina } s \text{ para a classe } c \text{ é lecionada pelo professor } p \text{ no dia } d, \text{ horário } t \\ 0 & \text{caso contrário} \end{cases}$$

Ademais, são utilizadas as seguintes variáveis auxiliares:

$$y_{p,d} = \begin{cases} 1 & \text{caso o professor } p \text{ esteja alocado a pelo menos um horário no dia } d \\ 0 & \text{caso contrário} \end{cases}$$

$$w_{s,c,d,t} = \begin{cases} 1 & \text{caso o evento } s \text{ esteja alocado à classe } c \text{ no dia } d \text{ no horário } t \\ 0 & \text{caso contrário} \end{cases}$$

$$z_{s,c,p} = \begin{cases} 1 & \text{caso o professor } p \text{ esteja alocado para o evento } s \text{ da classe } c \\ 0 & \text{caso contrário} \end{cases}$$

### 4.2.3 Função Objetivo

A função objetivo do modelo usado pela ferramenta desenvolvida neste trabalho tem quatro (4) objetivos:

1. Minimizar o número de violações em relação à preferência do número de aulas diárias dos docentes.

2. Minimizar o número de alocações em horários indesejados pelo docente.
3. Minimizar o número de alocações nas quais os professores são alocados em padrões indesejados de folga/dias trabalhados.
4. Minimizar o número de violações nas alocações de professores às disciplinas.

### 4.3 Arquivos de Entrada

Como arquivos de entrada para o [modelo matemático](#) optou-se por utilizar o formato CSV (Comma Separated Values). Os arquivos CSV são arquivos de texto formatados de acordo com a regulamentação RFC 4180, que faz uma ordenação de bytes ou um formato de terminador de linha, separando valores com vírgulas. Um exemplo de arquivo CSV é apresentado na Figura 1.

```
Professor;Carga Horaria Maxima;Minimo Dias;Maximo Dias;Horarios Indiponiveis;Horarios Indesejados;Padroes
Indesejados;Prioridade;Tag
Adam Sargeant;6;2;4;;1,3+1,4+2,3,+2,4+3,3+3,4+4,3+4,4+5,3+5,4;10001;1;DECEA
Alana Cavalcante;6;2;3;;5,1+5,2+5,3+5,4;10101+10001;1;DECEA
Alana Pereira;6;2;2;;;1;DEENP
Alexandre Martins;6;2;2;;;1;DEENP
Alexandre Souza;6;2;3;;3,1+3,2+3,3+3,4;10101;1;DECSI
Anliy Sargeant;2;2;2;;1,4+2,4+3,4+4,4+5,4;10001;1;DECEA
Bruno Rabelo;8;2;4;;1,4+2,4+3,4+4,4+5,4;;1;DECSI
Carlos Barbosa;4;2;2;;;10101+10011+11001+10001;1;DEELT
Carlos Barbosa_E1;;;;;;DEELT
Carlos Pontes;6;2;3;;1,1+1,2+1,3+1,4;10101+10001;1;DECEA
Carlos Silva;6;2;3;;;10101+10011+11001;1;DEELT
Carlos Silva_E1;;;;;;DEELT
Concurso Redes;8;2;4;;;0.9;DECSI
Concurso Teoria I;8;2;4;;;0.9;DECSI
Concurso Teoria II;8;2;4;;;0.9;DECSI
Darlan Brito;4;2;3;;5,3+5,4;1;DECSI
Diego Barros;6;2;2;;;10101+10001;1;DECEA
Diego Zuquim;6;2;2;;;10101;1;DECSI
```

Figura 1 – Exemplo de arquivo CSV de entrada.

Para cada conjunto de entidade se tem um arquivo correspondente, sendo cada linha do arquivo uma instância de entidade.

#### 4.3.1 Arquivo de Professores

O arquivo de professores é organizado da seguinte forma: Nome, Carga Horária,  $N^{\circ}$  mínimo de dias trabalhados,  $N^{\circ}$  máximo de dias trabalhados, Horários indisponíveis, Horários indesejados, Padrões indesejados, Prioridade, Tag.

*Exemplo: João Camargos, 6, 2, 3, "2,1+2,2", "5,1+5,2", 10101, 0.7, DECEA*

Neste exemplo temos que o professor João Camargos possui como carga horária máxima 6 horas semanais, deve trabalhar no mínimo 2 dias e no máximo 3, está indisponível nos horários 1 e 2 do segundo dia da semana, tem preferência por não trabalhar nos horários 1 e 2 do quinto dia da semana, tem como padrão indesejado trabalhar nos dias 1,3 e 5 e folgar nos dias 2 e 4, possui prioridade 0.7 e sua tag é DECEA.

### 4.3.2 Arquivo de Classes

O arquivo de classes é organizado da seguinte forma: Nome, Dias Indisponíveis.

*Exemplo: Engenharia de Minas - 4º Período., "1,1+1,2+2,1+2,2+3,1+3,2+4,1+4,2+5,1+5,2"*

Neste exemplo temos que a classe Engenharia de Minas - 4º Período não pode ser alocada nos horários 1 e 2 dos dias 1,2,3,4 e 5.

### 4.3.3 Arquivo de Eventos

O arquivo de eventos é organizado da seguinte forma: Classe, Disciplina, Tag, Duração, Professores Compatíveis,  $N^{\circ}$  mínimo de dias entre encontros,  $N^{\circ}$  máximo de dias entre encontros.

*Exemplo: Eng. de Computação - 7º Período, Sistemas Operacionais, DECSI, 4, Roberto Oliveira[4] + Maria Costa[0], 1, 2.*

Neste exemplo temos que a disciplina Sistema Operacionais da turma do 7º Período de Engenharia de Computação, de tag DECSI e 4 (quatro) horas semanais, tem como professores compatíveis Roberto Oliveira e Maria Costa. Caso o professor Oliveira seja alocado nessa disciplina, o custo 4 (quatro) será adicionado à função objetivo final. Se a professora Costa for escolhida para lecionar essa disciplina, o custo 0 (zero) será adicionado.

## 4.4 Ferramentas de Desenvolvimento

A linguagem de programação utilizada para o desenvolvimento do software descrito nesse trabalho é o Java versão 11.0.2, produzido e mantido sobre a licença da Oracle Corporation. Para o desenvolvimento da interface gráfica foi utilizada a plataforma de desenvolvimento OpenJFX, também chamada de JavaFX, em sua versão 12.

Como o modelo empregado na solução é um modelo de programação linear, optou-se por se utilizar o resolvidor Gurobi Optimizer em sua versão 8.1, desenvolvido pela Gurobi Optimization.

Como ambiente de desenvolvimento foram utilizadas as seguintes ferramentas:

- IntelliJIDEA versão 2018.2
- SceneBuilder versão 11.0.1
- H2Database versão 1.4.199
- Hibernate versão 5.4.1

## 4.5 Software Desenvolvido

O software desenvolvido apresenta uma interface amigável para que o usuário forneça os dados referentes aos professores, classes, eventos e restrições acerca da instituição de ensino, e obtenha como resposta uma solução válida, ou seja, uma alocação que satisfaça as restrições impostas. A Figura 2 apresenta a tela principal do software. É através dessa tela que o usuário tem acesso ao gerenciamento de todos os recursos da aplicação.

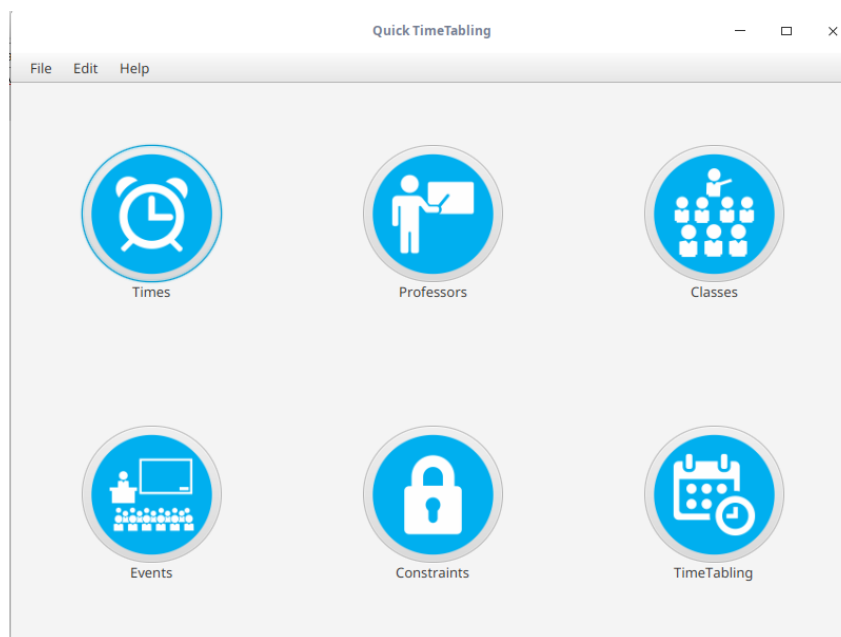


Figura 2 – Tela Principal

A Figura 3 apresenta a tela de gerenciamento de horários. Nessa tela são cadastrados os dias da semana e os horários disponíveis para alocação dos eventos da instituição de ensino. É também nesta tela que os rótulos de cada dia/horário são alterados.

A Figura 4 apresenta ao leitor a tela de gerenciamento das classes/turmas da instituição. Por meio desta tela, o usuário tem acesso à informações e operações como:

- A lista de todas as classes cadastradas no sistema.
- Eventos, i.e, disciplinas associadas a cada classe.
- As restrições impostas sobre cada classe.
- Deleção de uma determinada classe.
- Importação de classes armazenadas em arquivos .csv.

É por esta tela que se tem acesso às telas de cadastro e edição de classes.

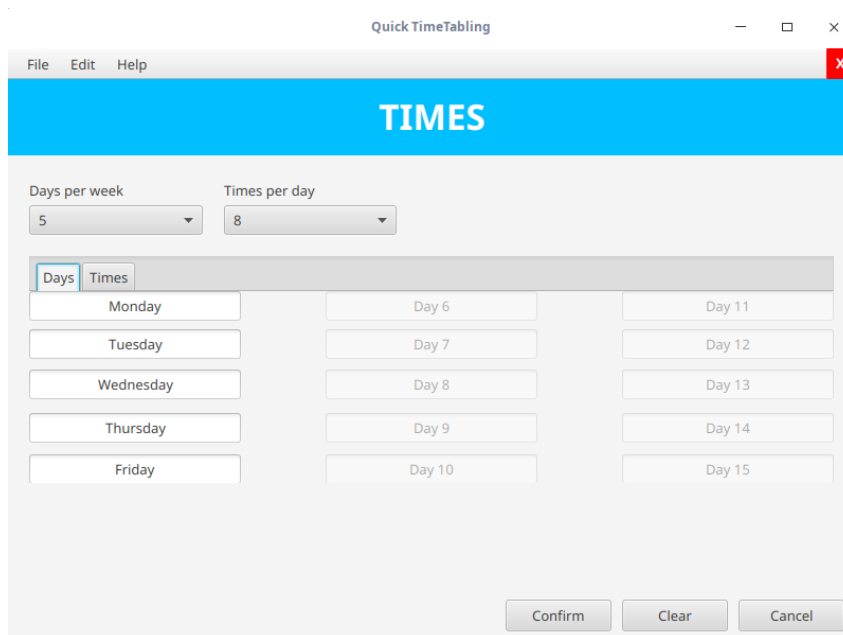


Figura 3 – Gerenciamento de Horários

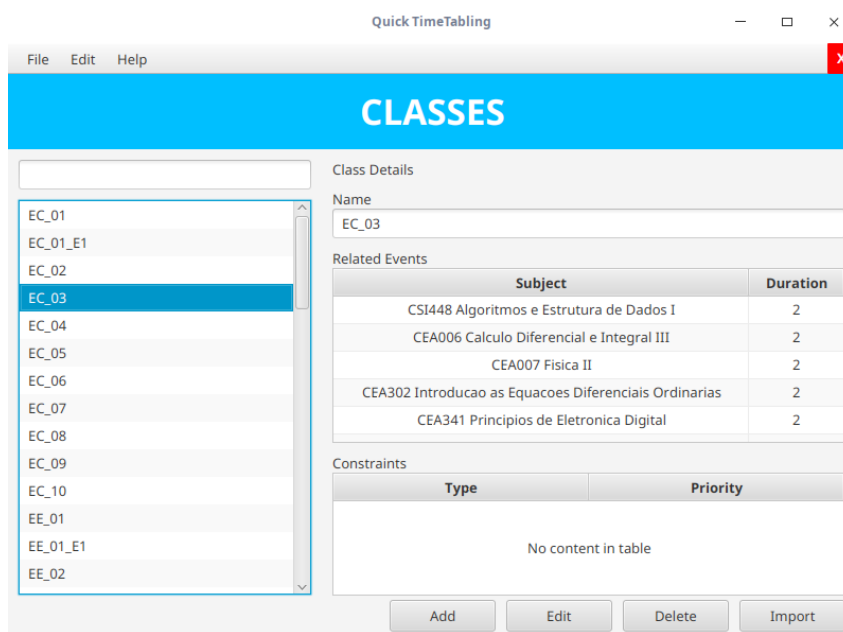


Figura 4 – Gerenciamento de Classes/Turmas

A Figura 5 mostra a tela de cadastro/edição de uma classe. É a apresentado ao usuário o agendamento semanal para tal classe, de modo que as células demarcadas de vermelho representam os horários indisponíveis e as verdes os horários disponíveis. A alteração do *status* para os horários é feita clicando sobre as células.

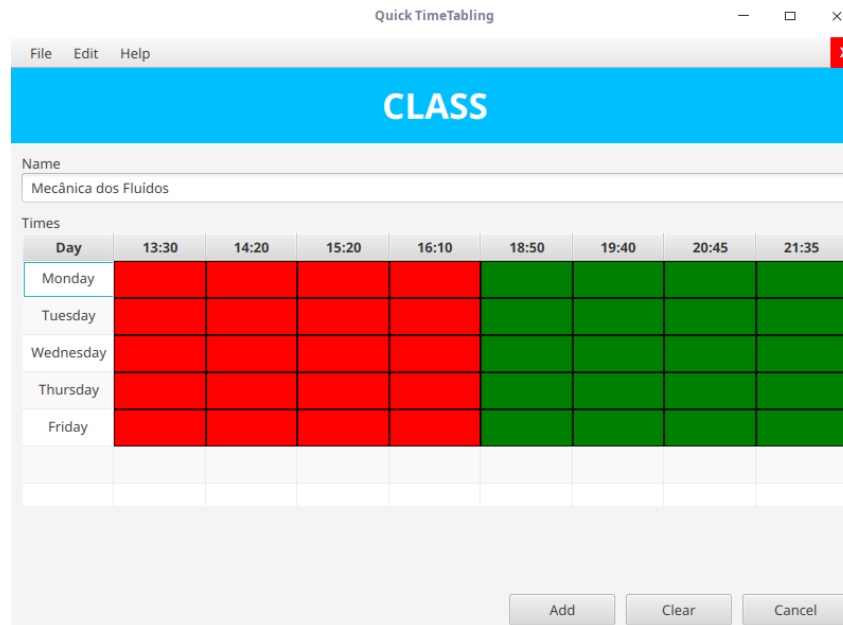


Figura 5 – Cadastro/Edição de Classes

A Figura 6 apresenta a tela de gerenciamento de professores. Esta tela exibe todos os professores registrados no sistema, além de exibir informações como restrições e eventos a que estes estão relacionados. É possível importar dados de professores armazenados no formato csv. O acesso às telas de cadastramento e edição também é realizado por meio desta tela.

A Figura 7 apresenta a tela de cadastro e edição de professores. Apresentando campos como carga horária máxima, números mínimo e máximo de trabalho, prioridade, tag, etc, esta tela possui campos para atribuição de todas as informações necessárias para o modelo relacionadas aos professores. Assim como no cadastro de classes, a tela de cadastro de professores apresenta uma tabela com células indicando os horários disponíveis e indisponíveis para o determinado professor. Adicionalmente, temos a cor amarela, que indica os horários indesejados por aquele professor. Ainda nessa tela, é possível cadastrar os padrões indesejados de folga/dias trabalhados, como é mostrado na Figura 8.

A Figura 9 expõe ao leitor a tela de gerenciamento de eventos cadastrados no sistema. Navegando por esta tela, o usuário tem acesso a dados como:

- Todas os eventos cadastrados no sistema.
- A lista de professores e seus respectivos custos de atribuição para cada evento.
- A duração de cada evento.



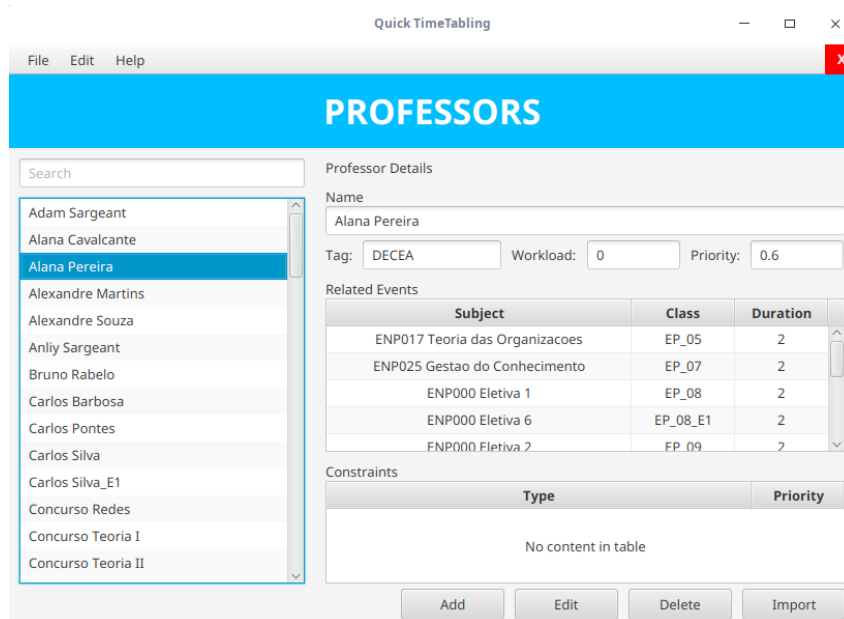


Figura 6 – Tela de Gerenciamento de Professores

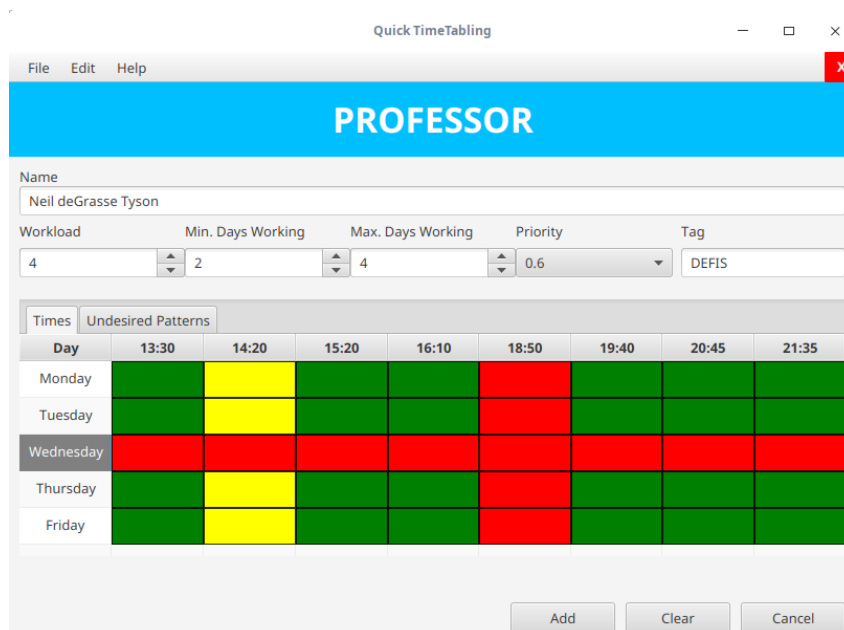


Figura 7 – Tela de Cadastro/Edição de Professores

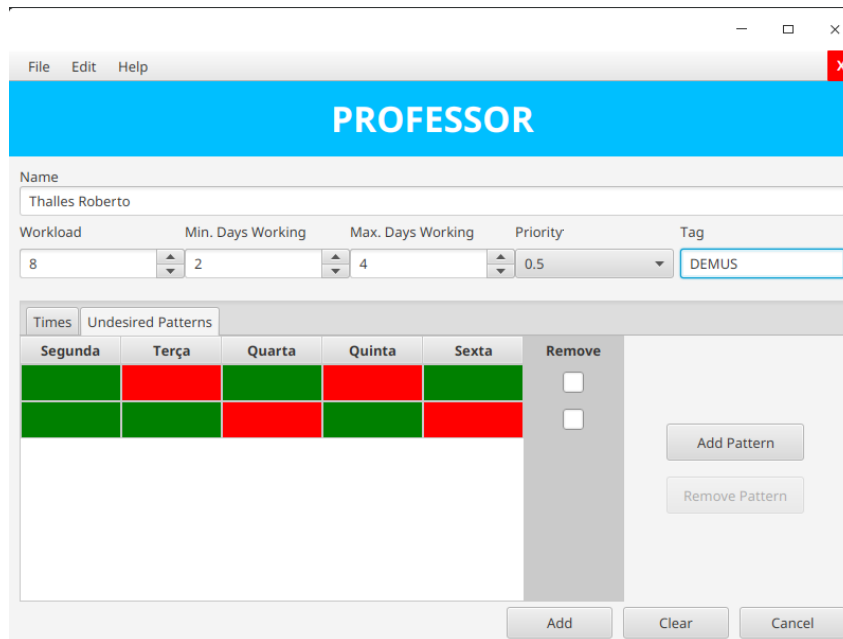


Figura 8 – Tela de Cadastro de Padrões Indesejados

- A classe ao qual este está atribuído.

Para aumentar a produtividade e facilidade no uso do software, foram posicionados nessa interface diversos filtros como, 'Filtrar por Professor', 'Filtrar por assunto', entre outros. A tela de gerenciamento também dá acesso a ações como deleção de eventos, importação de eventos armazenados em arquivos .csv, cadastramento de novos eventos, etc.

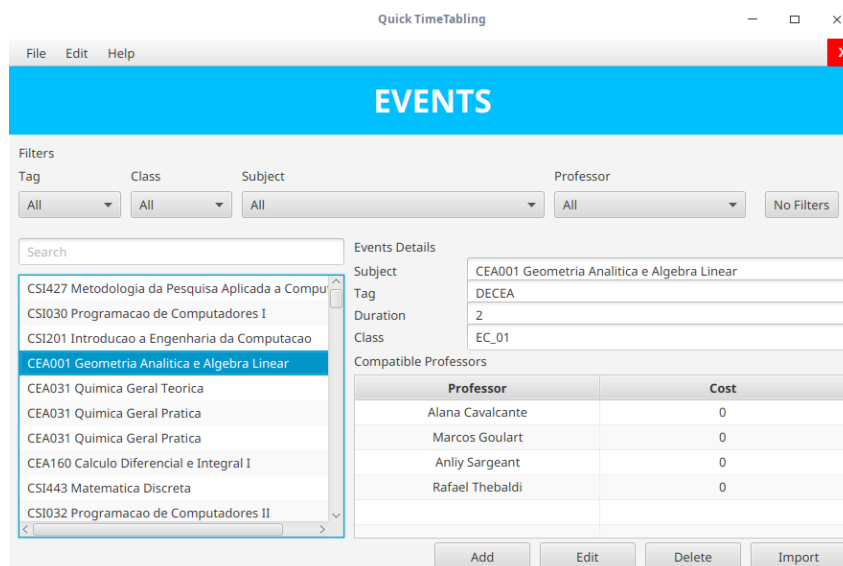
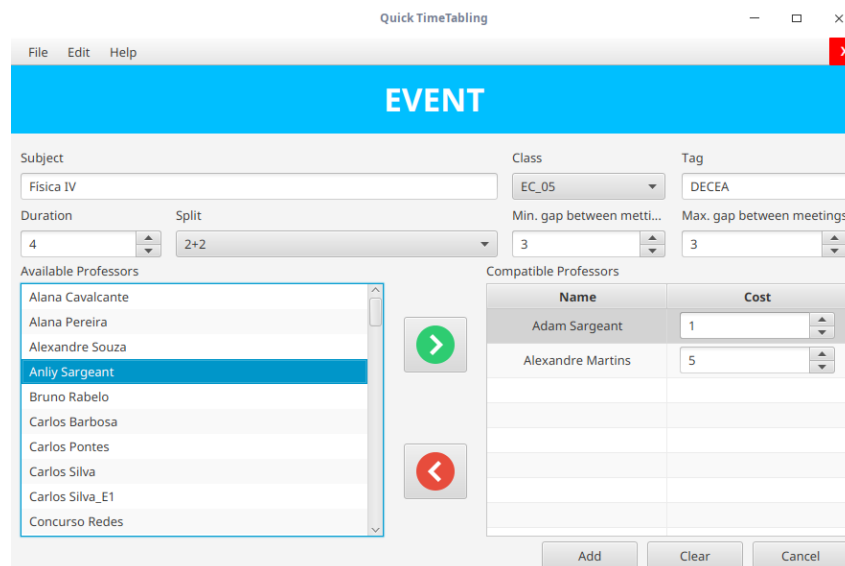


Figura 9 – Tela de Gerenciamento de Eventos

A Figura 10 exibe a tela de cadastro e edição de Eventos. Nesta tela o usuário tem a possibilidade de atribuir aos eventos:

- Um nome/assunto.
- Uma carga horária.
- Uma classe previamente cadastrada.
- Uma *tag*, e.g., Departamento de Biologia - DEB.
- Uma lista de professores elegíveis para lecionar tal disciplina e seus respectivos custos de atribuição.
- Restrições como modo de distribuição da carga horária durante a semana, espaçamento entre reuniões, etc.



The screenshot shows the 'EVENT' form in the 'Quick TimeTabling' application. The form is divided into several sections:

- Subject:** Física IV
- Class:** EC\_05
- Tag:** DECEA
- Duration:** 4
- Split:** 2+2
- Min. gap between meetings:** 3
- Max. gap between meetings:** 3

Below these fields, there are two lists of professors:

- Available Professors:** A list with 'Anily Sargeant' selected.
- Compatible Professors:** A table with columns 'Name' and 'Cost'.

Name	Cost
Adam Sargeant	1
Alexandre Martins	5

At the bottom of the form, there are three buttons: 'Add', 'Clear', and 'Cancel'.

Figura 10 – Tela de Cadastro/Edição de Eventos

Por último, porém não menos importante, temos a tela de apresentação da solução gerada pelo modelo, ou seja, uma alocação válida para os dados de entrada. A alocação é apresentada na forma de uma tabela semanal, onde as células representam os eventos alocados para o professor ou classe selecionado na lista lateral. As violações de restrições para a dada alocação são apresentadas na tabela abaixo do quadro de horários. Para melhorar a experiência do usuário, cada evento tem sua célula pintada com uma cor distinta das demais, facilitando a visualização. A solução gerada pode ser exportada no formato HTML. Esta tela é mostrada na Figura 11.

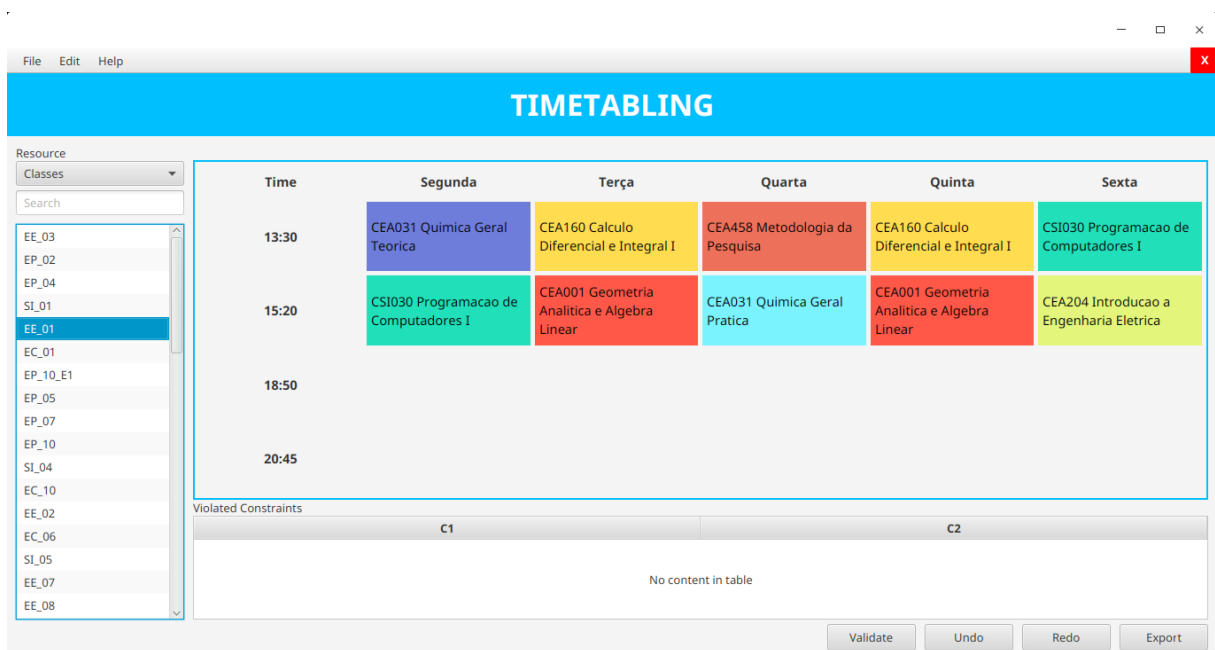


Figura 11 – Tela de Exibição da Solução

## 5 Resultados

Neste capítulo são apresentados os resultados alcançados durante todo o trabalho, bem como uma discussão e comparação com os resultados encontrados na literatura, destacando a importância desta pesquisa no contexto acadêmico.

# 6 Considerações Finais

## 6.1 Conclusão

A ferramenta desenvolvida neste projeto torna os processos de manipulação e geração de quadros de horários muito mais fáceis, uma vez que conta com uma interface de interação com os usuários. É também notável a redução do tempo necessário para fornecimento das entradas necessárias para o modelo matemático, o que traz agilidade e produtividade nesta tarefa. O código produzido pode ser acessado no link <https://github.com/FelipeTomazEC/QuickTimeTabling>.

## 6.2 Trabalhos Futuros

O software produzido, apesar de contar com *features* interessantes, possui oportunidades de melhorias, tais como:

- Possibilidade de alocação de salas.
- Uma versão para acesso via *web*.
- Versões para os sistemas operacionais Linux e Mac OS.

# Referências

ALVES, R.; DELGADO, C. Programação linear inteira. 1997. 3

ARROYO, J. E. C. et al. Heurísticas e metaheurísticas para otimização combinatória multiobjetivo. [sn], 2002. 3

CAMPOS, F. Algoritmo genético na resolução do problema da grade horária. *Trabalho de Conclusão de Curso. Salvador: Universidade Federal da Bahia*, 2002. 4

NETO, E. P.; VIANNA, D. S. Heurísticas eficientes para o problema de geração de grade escolar automatizada. *REVISTA ELETRÔNICA PRODUÇÃO E ENGENHARIA*, v. 4, n. 1, p. 330–337, 2013. 3

PREIS, T. A. Protótipo gerador de grades horárias para instituições de ensino. 2007. 3