

AGENDADOR DE HORÁRIOS DE PROVA

# RELATÓRIO

RESULTADOS DA ALOCAÇÃO DAS TURMAS

**DISTRIBUIÇÃO DAS  
TURMAS**

Horário em que cada turma  
deverá realizar a prova

**LISTA DOS ALUNOS  
PREJUDICADOS**

Quais alunos deverão fazer  
prova de segunda chamada  
por conflitos nos horários

**ENCONTRANDO O  
MELHOR RESULTADO**

Explicação de como o robô  
encontra a solução com o  
menor número de conflitos

# Resultados obtidos para realização das provas

Para aplicar as provas com sua solução, serão necessárias 3 provas de Segunda Chamada.

30	Provas no Horário 1
JAVA	10
PYTHON	10
R	10

10	Provas no Horário 2
PHP	10

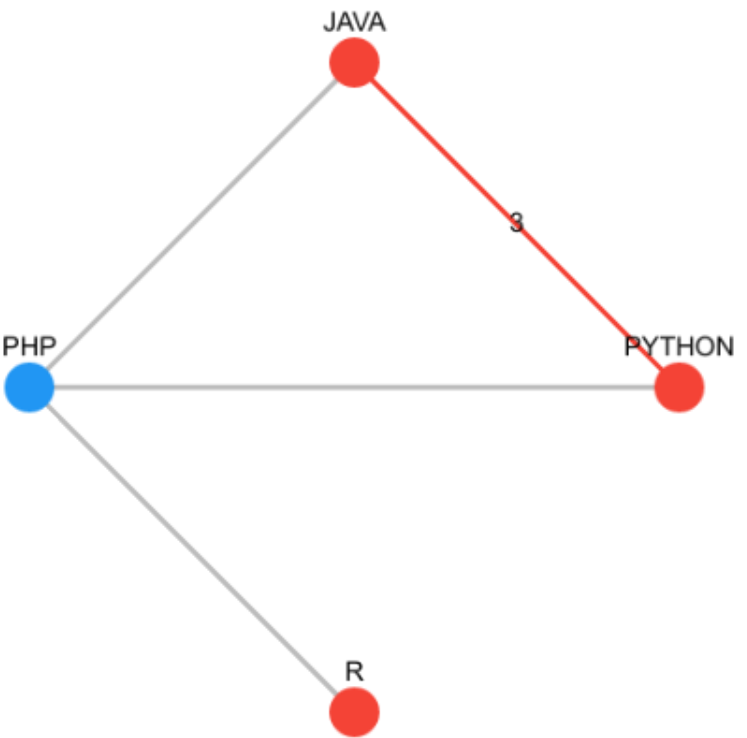
## **Horários para realização dos Exames**

Cada horário, representado por um retângulo de uma cor específica, possui as turmas que deverão realizar os exames em um mesmo horário.

Sobreposto à informação da turma, está a quantidade de alunos matriculados, dessa forma podemos saber o exato número de provas que cada disciplina precisará. O número total de provas necessárias em cada horário é apresentado no topo do retângulo, realizando o somatório das provas por turma, sem o decréscimo das avaliações para os alunos prejudicados, caso existam.

Resultado em forma de Grafo.

Com a visualização do resultado na forma de um grafo pode ser observado graficamente quais as turmas possuem alunos em comum e quais dessas turmas acabaram sendo alocadas em um mesmo horário.



Provas de segunda chamada

A lista a seguir apresenta as turmas que possuem conflitos entre alunos, e os alunos que necessitarão de provas de segunda chamada.

Alunos matriculados em JAVA e PYTHON		
#	Nome	Matrícula
1	Ana	6
2	Betina	3
3	Ulisses	9



## Método de Solução Utilizado

Visando permitir o uso do poder computacional na busca de uma solução para o problema, foi necessário modelar o problema de forma matemática.

Dessa forma foi optado o uso da teoria dos grafos como estrutura matemática visando modelar o problema.

## O que é Teoria dos Grafos?

Um **grafo** é definido por Szwarcifter (1986) como sendo uma estrutura matemática composta por um conjunto finito não vazio de vértices  $V$  e um conjunto  $E$  de pares, não ordenados, de diferentes vértices denominados arestas.

Um grafo é composto por uma série de elementos entre os quais se destacam:

- **Vértices:** um vértice também é chamado de  $V$  e representado graficamente através de um ponto, a quantidade dos vértices de um grafo é expressa por  $|V|$ , onde um grafo com  $|V| = 1$  é classificado como um grafo trivial;
- **Arestas:** uma aresta  $e$  é formada por um par de vértices, com isso  $e = (v, w)$ . Dessa forma os vértices  $v$  e  $w$  estão conectados as extremidades da aresta e dessa forma são denominados adjacentes. Já uma aresta é denominada incidente aos vértices em seus extremos  $v$  e  $w$ ;

## Como transformamos as turmas em Grafos?

Para utilizar a teoria dos grafos para modelar esse problema foi necessário a utilização dos grafos de interseção.

Pinto (2018) define um Grafo de Interseção, quando seja possível esse grafo  $G(V, E)$  representar as relações entre elementos de um conjunto através das relações entre seus vértices  $v \in V$ .

Dessa forma iremos relacionar os alunos das turmas aos vértices do Grafo, e criaremos arestas que indique a presença de alunos que componham ambas as turmas. Com isso teremos a turma como um grafo, que inclusive pode ser visualizada em sua forma geométrica.

## Definindo os horários de forma automática.

Após realizar a modelagem na forma de um problema de grafo, iremos utilizar essa estrutura para criar uma grade de horário de provas.

Para definir quais provas poderão ser aplicadas em um mesmo horário, iremos considerar os alunos pertencentes á ambas turmas. Com isso, caso duas turmas não possuam alunos em comum, elas poderão realizar seus exames em um mesmo horário.

Agora, para realizarmos a atribuição das turmas em horários aproveitaremos a forma geométrica do grafo, atribuindo uma cor a cada horário e pintando o vértice da cor correspondente ao horário.

A partir desse momento iremos atribuir á primeira turma o primeiro horário, dessa forma

podemos pintar o primeiro vértice da primeira cor (vermelho, por exemplo).

Ao pintar o próximo vértice, poderemos ter um caso onde ele possua alunos em comum com a turma anteriormente pintada, nesse caso iremos pintá-lo com nossa segunda cor (azul, por exemplo). Mas se as turmas não possuírem alunos em comum (ou seja, não sejam ligadas por uma aresta) poderemos usar a mesma cor para pintá-la.

Seguiremos repetindo esse procedimento de olhar em que horário estão as turmas que possuem alunos em comum para não colocarmos ambas as turmas no mesmo horário, o que causaria a necessidade de se aplicar uma prova de segunda chamada para os alunos prejudicados.

Esse processo de pintar os vértices do grafo, de forma que vértices adjacentes não recebam mesma cor, consiste em uma área muito estudada dentro da Teoria de Grafos, que é a Coloração de Grafos.

### O que é Coloração de Grafos?

Colorir um grafo  $G$  propriamente é definido por Goldbarg e Goldbarg (2012) como, atribuir cores aos seus vértices de forma que vértices adjacentes recebam cores distintas.

O principal problema em se colorir um grafo propriamente consiste em determinar qual o menor número possível para se colorir o grafo.

Esse número é chamado de número cromático do grafo ou  $\chi(G)$ . É um problema classificado como NP-Completo, sendo um dos problemas cuja solução não pode ser obtida em tempo polinomial de execução.

Contudo, a solução do problema de exames universitários por meio da coloração própria de vértices, não resolverá todos os casos possíveis, já que é possível existir um caso onde a quantidade de horários disponíveis para realização da prova é menor que o número cromático do grafo.

Nesse caso, como uma coloração própria não será suficiente, adotaremos uma nova forma de solução.

Considerando que não será possível separar todas as turmas que possuem alunos em comum em horários de forma que não gere conflitos, iremos buscar uma solução que minimize a quantidade de provas de segunda chamada, ou seja, a número de alunos prejudicados.

Para sabermos quantos alunos estarão sendo prejudicados precisaremos incluir mais uma etapa em nossa modelagem.

Iremos atribuir às arestas do grafo um peso, correspondente a quantidade de alunos que pertencem às duas turmas. Dessa forma teremos um **Grafo Ponderado em Arestas**.

Colorir o grafo agora consistirá em encontrar uma coloração não própria cuja soma dos pesos das arestas incidentes em vértices coloridos com mesma cor seja o menor possível.

Esse processo é chamado de Generalized Graph Coloring, onde efetuamos um tipo especial de coloração de modo a atingir algum objetivo pré-estabelecido.

### Generalized Graph Coloring Problem?

Definido por Vredeveld (2002), o k-GGCP é um problema que consiste em um grafo  $G = (V, E)$ , uma função de peso  $z : E \rightarrow Z$  nas arestas, e um número inteiro  $k \geq 2$ .

Em que deseja-se encontrar uma atribuição de cores  $c : V \rightarrow \{1, \dots, k\}$  dos vértices que minimize o peso total das arestas monocromáticas do grafo, isto é, arestas que incidem em vértices com mesma cor.

### **Enfim o Resultado!**

Ao buscar a distribuição que cause o menor número de conflitos chegaremos à coloração que será o resultado desejado para o problema. Assim como o obtido pelo robô no sistema.

## **O Software de Agendamento de Horários de Prova**

Este sistema foi desenvolvido por **Filipe Rodrigues Cardoso da Silva**. A partir do ano de 2019, com objetivo de ser um protótipo funcional para o Trabalho de Conclusão de Curso de Análise de Sistemas Informatizados, intitulado de **"Utilizando Coloração de Grafos de Interseção para Resolver o Problema de Programação de Horários de Exames Universitários"** vinculado a Faculdade de Educação Tecnológica do Estado do Rio de Janeiro - FAETERJ-Rio.

O Agendador tem como finalidade criar horários de provas universitárias de forma que o menor número possível de alunos tenham que realizar provas de segunda chamada, em decorrência de ter duas ou mais provas de diferentes turmas num mesmo horário.

O sistema foi desenvolvido usando MariaDB para o banco de dados, PHP no backend e HTML, CSS e JavaScript no frontend. Utilizando o framework Bootstrap para criar uma interface mais coesa e simples, e a biblioteca Cytoscape.js para possibilitar gerar grafos interativos.

Todo o código do sistema, a íntegra do trabalho escrito e os slides da apresentação podem ser obtidos através do repositório oficial do trabalho no Github: <https://github.com/FilipeRodrigues3003/TCC>.

## REFERÊNCIAS

GOLDBARG, Marco Cesar; GOLDBARG, Elizabeth Ferreira Gouvêa. **Grafos: Conceitos, algoritmos e aplicações**. [S.l.]: Elsevier Editora Ltda, 2012. v. 1.

PINTO, José Wilson Coura. **Grafos ORTH[h, s, t]**. 2018. Tese (Tese (Doutorado)) — UFRJ/COPPE, 2018.

SZWARCIFTER, Jayme Luiz. **Grafos e Algoritmos Computacionais**. 2. ed. [S.l.]: Campus, 1986. 35-42, 169-195 p.

VREDEVELD, Tjark. **Combinatorial Approximation Algorithms: Guaranteed versus experimental performance**. 2002. Tese (PhD) — Technische Universiteit Eindhoven, 2002.

SILVA, Filipe Rodrigues Cardoso da. **Utilizando Coloração de Grafos de Interseção para Resolver o Problema de Programação de Horários de Exames Universitários**. 2020. (Trabalho de Conclusão de Curso - TCC) - Faculdade de Educação Tecnológica do Estado do Rio de Janeiro - FAETERJ-Rio, 2020.