



Lisbon School
of Economics
& Management
Universidade de Lisboa

MESTRADO EM

MÉTODOS QUANTITATIVOS PARA A DECISÃO

ECONÓMICA E EMPRESARIAL

TRABALHO FINAL DE MESTRADO

RELATÓRIO DE PROJETO

PROBLEMA DE ESCALONAMENTO DE DOCENTES A

VIGILÂNCIAS: UMA APLICAÇÃO NO ISEG

ACÁCIO DOS SANTOS CARRIÇO REBOCHO

ORIENTAÇÃO:

PROFESSORA DOUTORA LEONOR ALMEIDA LEITE SANTIAGO PINTO

PROFESSOR DOUTOR JOÃO PAULO VICENTE JANELA

OUTUBRO-2023

AGRADECIMENTOS

À Minha Família:

Aos meus pais, não há palavras suficientes para expressar a minha gratidão pelo apoio inabalável que me deram ao longo de todo este caminho. No período universitário, quando os desafios pareciam intransponíveis, sempre estiveram presentes para me encorajar, sustentar e acreditar em mim. O vosso incentivo foi fundamental para chegar até aqui.

A minha irmã, foi também a minha âncora durante este processo. Agradeço a disponibilidade para esclarecer dúvidas, oferecer uma perspetiva externa valiosa e, acima de tudo, por ser a minha maior apoiante.

Aos Meus Professores e Orientadores:

À Professora Leonor Pinto e ao Professor João Janela, a minha gratidão é imensurável. Sem a vossa orientação, *insights* e conhecimentos nas áreas abordadas neste projeto, esta teria sido uma tarefa insuperável. A sua paixão pelo ensino e dedicação aos alunos tornaram esta jornada incrivelmente gratificante.

Já a Professora Cândida Mourão desempenhou um papel crucial na fase inicial deste projeto, apontando-me na direção certa e recomendando os orientadores ideais. O seu discernimento e orientação foram, mais uma vez, essenciais tanto para este projeto como para o meu desenvolvimento pessoal e académico.

Estou profundamente grato a todos os que contribuíram de uma maneira ou de outra para este projeto. As vossas influências permitiram-me alcançar este marco significativo na minha vida. Obrigado por acreditarem em mim e por me inspirarem a fazer o meu melhor.

RESUMO

Este projeto tem como objetivo construir um sistema de apoio à decisão (SAD), o ESCPRO, para o escalonamento de docentes a vigilâncias de exames do departamento de matemática do ISEG - Universidade de Lisboa. Os docentes devem ser escalonados com base na sua categoria, carga horária percentual contratada, saldo prévio de vigilâncias, unidades curriculares lecionadas, possíveis incompatibilidades e indisponibilidades. O ESCPRO inclui a resolução de um modelo de programação linear inteira mista (PLIM).

A implementação do SAD é feita em *Python* recorrendo ao *Gurobi* e ao *OR-Tools* para resolver o modelo. Inicialmente, pensou-se usar V.B.A. com *OpenSolver* mas esta opção foi abandonada por se revelar menos adequada, contudo serviu para comparação de resultados.

O *interface* gráfico (G.U.I. - *Graphical User Interface*), desenvolvido usando a biblioteca *Tkinter* do *Python*, tem o objetivo de facilitar a leitura e resolução do problema pelo utilizador, pois o script com a programação do modelo e formação do G.U.I. é posteriormente convertido num ficheiro executável isolado, de modo que o utilizador não precise de ter conhecimentos de programação para resolver e/ou criar o modelo.

O sistema foi testado com dados fornecidos pelo ISEG, num ficheiro de Excel, tendo produzido um escalonamento em poucos segundos. Deste modo, pode considerar-se que é cumprindo o propósito que presidiu à realização deste projeto: economizar tempo no desempenho desta tarefa que se realiza pelo menos duas vezes por ano.

Palavras-chave: Problema de Escalonamento, PLIM, Python, G.U.I., V.B.A.

ABSTRACT

This project aims at building a decision support system (DSS), the ESCPRO, applied to the professor scheduling problem for exam surveillances of the mathematic department at ISEG – University of Lisbon. The professors should be scheduled to surveillances based on the professors, category, percentage workload, prior surveillance balance, taught course units, individual unavailability's, or incompatibilities. The ESCPRO includes the solving of the mixed integer linear programming (MILP).

The implementation of the DSS it's done throw Python programming language and using either Gurobi or OR-Tools solvers to solve the model. Initially, was stipulated to use V.B.A. programming language and solving the problem throw OpenSolver but this option was abandoned, since it was revealed to be miss aligned, but throw it was possible to create a baseline to compare the results.

The G.U.I. (Graphical User Interface), developed using Python's Tkinter library, intends to facilitate the reading and solving of the problem by the user, since the script with the programming and forming of the G.U.I. is later converted into an isolated executable file, so that the user doesn't need any programming skills to upload and solve the model.

To test the model, we use data provided by ISEG, in an Excel file, producing a scheduling in a few seconds. This way, it can be considered that the main purpose behind this project is successfully fulfilled: saving time in carrying out this task, which is carried out at least twice a year.

Keywords: Scheduling Problem, MLIP, Python, G.U.I., V.B.A.

ÍNDICE

AGRADECIMENTOS.....	I
RESUMO	II
ABSTRACT	III
GLOSSÁRIO	V
ÍNDICE DE TABELAS.....	VI
ÍNDICE DE FIGURAS.....	VI
CAPÍTULO 1: INTRODUÇÃO	1
1.1. Contextualização.....	1
1.2. Descrição do ISEG.....	1
1.3. Estrutura do Relatório.....	4
CAPÍTULO 2: ENQUADRAMENTO DO PROBLEMA	5
2.1. Descrição do Problema	5
CAPÍTULO 3: REVISÃO DA LITERATURA	7
CAPÍTULO 4: METODOLOGIA	9
4.1 Modelo de Programação Linear Inteira Mista	9
4.2 V.B.A.....	12
4.3 Python.....	13
CAPÍTULO 5: INTERFACE GRÁFICA	15
5.1. Caracterização das várias janelas do G.U.I.....	15
5.2. Elaboração do ficheiro Setup do sistema.....	21
CAPÍTULO 6: ANÁLISE COMPARATIVA DE RESULTADOS.....	23
6.1. Análise Qualitativa das soluções ótimas: V.B.A. vs <i>Gurobi</i> vs <i>OR-Tools</i>	23
6.2. Comparação da facilidade de implementação do código: V.B.A. vs <i>Python</i> ...	25
6.3. Análise comparativa de tempo de execução: V.B.A. vs <i>Python</i>	26
CAPÍTULO 7: CONCLUSÃO	28
REFERÊNCIAS BIBLIOGRÁFICAS.....	30
ANEXOS	31

GLOSSÁRIO

ESCPRO – SAD para o escalonamento de docentes a vigilâncias de exames

G.U.I. – *Graphical User Interface*

IICL – Instituto Industrial e Comercial de Lisboa

ISEG – Instituto Superior de Economia e Gestão

MBA – *Master of Business Administration*

PLIM – Programação Linear Inteira

SAD – Sistema de Apoio à Decisão

UC – Unidade(s) Curricular(es)

V.B.A. – *Visual Basic for Applications*

DSS - *Decision Support System*

ÍNDICE DE TABELAS

Tabela 1 - Resumo das várias soluções ótimas encontradas.....	24
Tabela 3 - Número de ocorrências que o docente tem 0,1,2 ou 3 Vigilâncias/Dia	86

ÍNDICE DE FIGURAS

Figura 1- Ilustração do Campus do ISEG (ISEG, 2023).	3
Figura 2 - Executável responsável pela instalação do Sistema	15
Figura 3 - Executável responsável por iniciar o Software.....	16
Figura 4 - Primeira janela do Software.....	16
Figura 5 - Segunda janela do Software.....	17
Figura 6 - Aviso do sistema, o utilizador não introduziu uma localização válida do ficheiro de dados.....	18
Figura 7 - Aviso do sistema, o ficheiro selecionado não é do tipo .xlsx do Excel	18
Figura 8 - Aviso do sistema, o ficheiro selecionado não contém alguma das folhas necessárias com os dados	18
Figura 9 - Aviso do sistema, existe, no mínimo, uma incompatibilidade nos dados do problema.....	19
Figura 10 - Terceira janela do Software	19
Figura 11 - Janela de espera enquanto se determina uma solução ótima para o problema	20
Figura 12 - Quarta, e última, janela do Software.....	21
Figura 13 - Executável do Inno Setup Compiler	22
Figura 14 - Gráfico ilustrativo do aumento percentual do número de ocorrências de docentes com mais de uma vigilância diária, face à nova solução ótima.....	25
Figura 15 - Matriz de Escalonamento – Gurobi	77
Figura 16 - Matriz de Escalonamento – OR-Tools.....	78

Figura 17 - Matriz de Escalonamento – V.B.A.	79
Figura 18 - Matriz de Escalonamento Ótimo – Modelo Alterado.....	80
Figura 19 - Dados Agrupados por Dia – Gurobi	81
Figura 20 - Dados Agrupados por Dia – OR-Tools.....	82
Figura 21 - Dados Agrupados por Dia – V.B.A.	83
Figura 22 - Dados Agrupados por Dia – Modelo Alterado.	84

CAPÍTULO 1: INTRODUÇÃO

1.1. Contextualização

O presente relatório descreve o ESCPRO, um sistema de apoio à decisão (SAD) para afetar dos docentes a vigilâncias nas provas escritas dos períodos de avaliação do ISEG, de modo a equilibrar o número de horas de vigilância atribuídas aos docentes. Com o sistema pretende-se automatizar todo o processo desde a recolha de dados até à produção de um mapa de vigilâncias.

Neste sentido, foi desenvolvido um *interface* gráfico (G.U.I.- *Graphical User Interface*) através do *Python*, mais especificamente da biblioteca *Tkinter*, que permite aos utilizadores introduzir os dados e obter uma solução adequada determinada através de um modelo de programação linear inteira mista (PLIM). Os dados referem-se à caracterização do período de avaliação, contendo assim informação sobre os docentes, as vigilâncias e as unidades curriculares (UC). O G.U.I. foi desenhado para ser intuitivo, fácil de usar e completamente independente do *Python*, o que permite a sua utilização por um público muito mais amplo, mesmo utilizadores sem qualquer tipo de experiência em programação ou otimização.

O relatório, para além do detalhar o processo de desenvolvimento do G.U.I., apresenta a formulação do problema de PLIM que é resolvido para encontrar uma solução.

O projeto tem como objetivo último construir uma ferramenta, o ESCPRO, que permita aumentar a rapidez e eficácia no escalonamento de docentes a vigilâncias nas provas escritas do ISEG.

1.2. Descrição do ISEG

O Instituto Superior de Economia e Gestão (ISEG) pertence à Universidade de Lisboa. O ISEG, fundado em 1911, contando com mais de 110 anos de existência, é a mais antiga faculdade de Economia e Gestão em Portugal. A sua qualidade está patente não apenas na sua longevidade, mas mais significativamente, na reconhecida qualidade

dos seus graduados e na investigação realizada nas áreas das ciências económicas, financeiras e empresariais.

A história da origem do ISEG remonta, no entanto a 1759, ano em que foi criada a Aula do Comércio, um dos primeiros estabelecimentos de ensino oficial no mundo, com o objetivo de formar profissionais no setor do comércio. Cerca de 100 anos depois, a Aula do Comércio é anexada ao Liceu de Lisboa, surgindo a Escola de Comércio, que por sua vez foi posteriormente integrada no Instituto Industrial de Lisboa, e designando-se, a partir daí, por Instituto Industrial e Comercial de Lisboa (IICL). Já em 1911, o IICL dá origem a dois novos institutos, o Instituto Superior Técnico e o Instituto Superior de Comércio tendo este último, mais tarde, dado origem ao ISEG, nome adotado a partir de 1989 (ISEG, 2023).

Atualmente, o ISEG funciona em diversos edifícios, cada um com um propósito específico. Os edifícios Francesinhas 1 (nº 5 na Figura 1) e Francesinhas 2 (nº 6 na Figura 1) destinam-se às aulas dos alunos de licenciatura e mestrado, contendo por isso salas de aula, anfiteatros, salas de estudo, secretarias, bares e uma cantina. Já os edifícios Quelhas (nº 1 e nº 2 na Figura 1) e Bento Jesus Caraça (nº 3 na Figura 1) são sobretudo dedicados aos alunos de nível de formação académica superior, como pós-graduação, mestrado, MBA, doutoramento ou formação executiva. Nos edifícios que acabam de se mencionar situam-se os gabinetes dos docentes e funcionam tanto os centros de investigação como serviços diversos. Por fim, a biblioteca Pereira de Moura (nº 4 na Figura 1), é uma grande infraestrutura com um vasto leque de obras na área das ciências económicas e empresariais, bases bibliográficas, informação estatística e espaços de estudo e leitura.

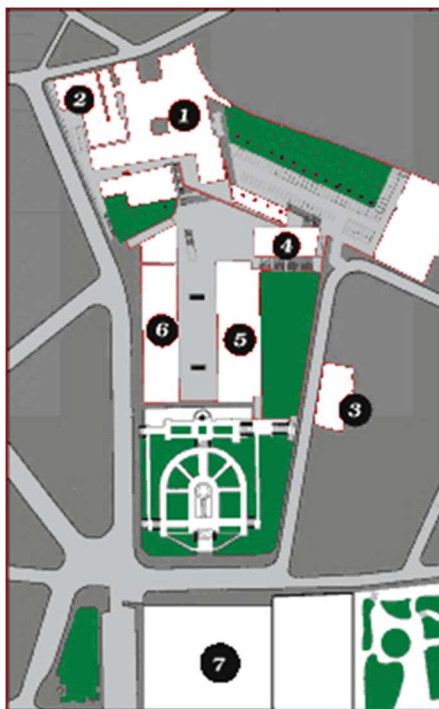


Figura 1- Ilustração do Campus do ISEG (ISEG, 2023).

Em termos de oferta educativa, o ISEG, quer em português quer em inglês, disponibiliza licenciaturas, mestrados, doutoramentos, cursos de formação executiva e um MBA, sobretudo na área das ciências económicas e empresariais.

Apesar da licenciatura em Economia ser um dos seus cursos mais antigos, a sua oferta vai muito para além disso tendo licenciaturas em Gestão, Economia, *Economics*, *Finance*, *Management*, Matemática Aplicada à Economia e à Gestão, Estudos Gerais e Gestão do Desporto. As duas últimas opções são lecionadas em parceria, no primeiro caso com a Faculdade de Letras e, no segundo, com a Faculdade de Motricidade Humana.

A oferta formativa do ISEG é complementada por um MBA, diversos mestrados e doutoramentos, nas áreas da economia, finanças, gestão, marketing, análise de dados entre outras, que procuram dar resposta às necessidades do mercado de trabalho e proporcionar uma formação diferenciada.

De notar que, para corresponder a toda a oferta formativa, o número quer de docentes quer de unidades curriculares é necessariamente ser elevado. Este aspeto, induz complexidade na tarefa de atribuição de vigilâncias aos docentes, algo a reter durante o presente projeto.

1.3. Estrutura do Relatório

A estrutura deste relatório segue uma ordem lógica e sequencial que visa apresentar de forma clara e concisa o ESCPRO, o sistema de apoio à decisão para efetuar o Escalonamento de Docentes a Vigilâncias. Este relatório está organizado em sete Capítulos.

O Capítulo 2 faz o enquadramento do problema, sendo este descrito detalhadamente. No Capítulo 3, Revisão da Literatura, referem-se trabalhos para problemas de escalonamento, em particular situações semelhantes à estudada neste relatório. O Capítulo 4, contém a metodologia onde se detalha o *software* usado no desenvolvimento do sistema e se apresenta a formulação do problema de PLIM resolvido. No Capítulo 5, para além da exposição do processo de criação próprio sistema e suas etapas, apresenta-se um manual do utilizador.

No Capítulo 6, Análise Comparativa de Resultados, são comparados os resultados obtidos da otimização do modelo de programação linear inteira mista obtidos com a implementação em V.B.A. com o *OpenSolver* e em *Python*, tanto recorrendo ao solver *Gurobi* como ao *OR-Tools*. As alternativas de implementação são também avaliadas relativamente ao tempo de execução e à simplicidade do processo de implementação

Por fim, no Capítulo 7, são resumidas as principais conclusões do trabalho e apresentadas sugestões para possíveis trabalhos futuros.

CAPÍTULO 2: ENQUADRAMENTO DO PROBLEMA

2.1. Descrição do Problema

O departamento de Matemática do ISEG é responsável pelo escalonamento de docentes a vigilâncias de todas as UC da sua área lecionadas durante cada época de avaliação. Como se mencionou, o principal objetivo deste projeto criar um sistema para efetuar esta tarefa de forma automática, que se consubstanciou no ESCPRO.

A informação relevante para a resolução do problema tem a ver, por um lado, com a caracterização dos docentes a escalonar e, por outro, com a lista exames ou provas, de unidades curriculares, na época de avaliação em causa. Em relação aos docentes, deve ser considerado qual a sua categoria (assistente, auxiliar, associado ou catedrático), a sua percentagem de contrato, o saldo de vigilâncias na época de avaliação anterior, as vigilâncias próprias e as suas indisponibilidades. Em relação aos exames, é necessário ter em conta a data e horário da prova de avaliação, a eventual incompatibilidade entre vigilâncias e o número de vigilantes necessários em cada prova de avaliação.

Designamos por vigilâncias próprias as vigilâncias a que um docente está obrigado, isto é, as vigilâncias de exames de UC que um docente leciona no semestre correspondente à época de avaliação. As indisponibilidades dos docentes são conhecidas, podendo ocorrer por diversos motivos. A incompatibilidade entre vigilâncias decorre de sobreposições horárias de provas.

Um escalonamento consiste numa lista de vigilâncias atribuída a cada um dos docentes, respeitando as condições que decorrerem dos fatores acima elencados. Mais precisamente, a cada prova são afetos o número de vigilantes estabelecido, e a lista vigilâncias atribuídas a cada docente é compatível e está de acordo com a sua categoria, o seu contrato, e o seu saldo de vigilâncias das épocas anteriores, contém as suas vigilâncias próprias e respeita as suas indisponibilidades. O objetivo é determinar um escalonamento o mais equilibrado possível em termos da distribuição do número de provas atribuídas aos docentes.

Com o aumento do número de UC com provas em cada época, a automatização do processo de escalonamento de vigilâncias a docentes permite poupar bastantes recursos. Essa automatização requer a resolução de um problema de programação linear inteira mista.

Antes de apresentar a formulação do problema, faz-se uma resenha bibliográfica no capítulo que se segue.

CAPÍTULO 3: REVISÃO DA LITERATURA

Podemos definir escalonamento como a ação de organizar e atribuir os recursos disponíveis a tarefas que devem de ser concluídas num período temporal tendo como objetivo, por exemplo, a minimização de custos.

Os problemas de escalonamento de recursos têm sido bastante estudados pela comunidade científica. O interesse no seu estudo é motivado pela variedade de aplicações como se pode comprovar pela lista de empresas dos mais diversos setores que necessitam de resolver problemas deste tipo (Brucker *et al.*, 2005) e (Burke *et al.*, 2004). De facto, a qualidade de um escalonamento pode ter um impacto significativo na produtividade e no lucro das organizações. Especialmente quando se pretende minimizar o número de trabalhadores capazes de assegurar uma qualidade de serviço preestabelecida, (Hast, 2017) e (Van Den Bergh *et al.*, 2013).

Em (Bouajaja *et al.*, 2017) podem ser encontradas várias metodologias para chegar a uma solução adequada para problemas de escalonamento, utilizando quer métodos exatos, para encontrar uma solução ótima, quer métodos não exatos, que na maioria das vezes não permitem garantir a obtenção de uma solução ótima, mas são uma boa alternativa em presença de instâncias de grande dimensão.

Segundo (Baker, 1976), em 1954 surgiu umas das primeiras classificações deste tipo de problemas:

- *Shift Scheduling* – onde se planeia o tempo dedicado, num dia de atividade, para a pausa e para o trabalho propriamente dito;
- *Day-off Scheduling* – onde se faz a escala das várias folgas num espaço de tempo, normalmente uma semana de atividade;
- *Tour Scheduling* – onde se planeia o escalonamento dos turnos de trabalho, sendo na prática uma junção das duas classificações anteriores.

Como é natural, este tipo de classificações tem evoluído ao longo do tempo. Mais recentemente, (Ernst *et al.*, 2004) lista um conjunto de módulos com identificação de características que podem eventualmente ser combinadas. Dessa lista constam seis módulos:

- *Demand Modelling* – determinar o número mínimo de funcionários necessário para um certo nível de serviço;

- *Day-off Scheduling* – localizar as folgas dos colaboradores;
- *Shift Scheduling* – selecionar membros dentro dos colaboradores disponíveis para cumprir certas escalas;
- *Line of Work Construction* – determinar sequências de turnos, ao longo do tempo, para cada membro de uma equipa, podem atendidas relações de precedência entre trabalhos a executar;
- *Task Assignment* - afetar tarefas aos vários colaboradores escalonados, possivelmente mais de uma tarefa para cada trabalhador associado a um mesmo turno;
- *Staff Assignment* – distribuir os colaboradores por sequências de turnos.

De acordo com esta tipologia o problema de escalonamento do presente projeto, assemelha-se de certo modo a um problema de *Task Assignment*, onde basicamente cada tarefa, (vigilância) é escalonada a um colaborador (docente), mas podendo existir vários colaboradores a cumprir a mesma tarefa se necessário.

Em (Breslaw, 1976) podemos encontrar uma aplicação de afetação no âmbito de uma universidade. Trata-se da alocação de cursos a faculdades de uma universidade conhecendo os valores que traduzem a preferência de cada curso ser atribuído a cada faculdade. O problema formula-se como um problema de afetação com função objetivo a maximizar a preferência total na universidade. Neste caso, a programação linear resolve o problema. No caso da situação deste projeto é precisamos da programação linear inteira.

CAPÍTULO 4: METODOLOGIA

A primeira seção deste capítulo contém a formulação do problema de PLIM usado para resolver o problema de escalonamento de docentes a vigilâncias no ISEG no ESCPRO. As seções seguintes detalham as implementações usadas: em VBA e em Python.

4.1 Modelo de Programação Linear Inteira Mista

A presente seção destina-se a apresentar o modelo de programação linear inteira mista que é resolvido no sistema desenvolvido.

Considere-se a seguinte notação:

- Doc o conjunto de docentes;
- n o número de docentes;
- $Doc^{cat} \subset Doc$ o conjunto de docentes catedráticos;
- Pro o conjunto de provas;
- m o número de provas;
- M_i o número máximo de vigilâncias a atribuir ao docente i , $i \in Doc$;
- S_i o saldo de vigilâncias anteriores do docente i , $i \in Doc$;
- β_i a percentagem a que é contratado o docente i , $i \in Doc$;
- V_j o número de docentes necessários para a prova j , $j \in Pro$;
- $C_{jk} = \begin{cases} 1 & \text{se as provas } j \text{ e } k \text{ são incompatíveis} \\ 0 & \text{caso contrário} \end{cases}$, $j, k \in Pro$;
- $D_{ij} = \begin{cases} 1 & \text{o docente } i \text{ não pode vigiar a prova } j \\ 0 & \text{caso contrário} \end{cases}$ $i \in Doc, j \in Pro$;
- $P_{ij} = \begin{cases} 1 & \text{o docente } i \text{ leciona a UC da prova } j \\ 0 & \text{caso contrário} \end{cases}$, $i \in Doc, j \in Pro$;
- $S^+ = \{i \in Doc \setminus Doc^{cat} : S_i \geq 0\}$ conjunto de docentes com saldo credor ou nulo
- $S^- = \{i \in Doc \setminus Doc^{cat} : S_i < 0\}$ conjunto de docentes sem saldo credor

O número máximo de vigilâncias a atribuir ao docente i , M_i , depende de vários fatores.

Os docentes catedráticos fazem apenas as vigilâncias próprias, isto é, deve aplicar-se (1).

$$M_i = \sum_{j=1}^m P_{ij}, \quad \text{para } i \in Doc^{cat} \quad (1)$$

As restantes vigilâncias devem ser divididas de forma equilibrada entre os docentes não catedráticos sem esquecer que qualquer docente faz as vigilâncias próprias. Assim, o número máximo de vigilâncias a ser atribuído a um docente não catedrático também deve ter em consideração o seu saldo, S_i , e a percentagem a que é contratado, β_i . Mais concretamente, deve usar-se (2).

$$M_i = \max \left\{ [\bar{M} * \beta_i - S_i], \sum_{j=1}^m P_{ij} \right\}, \quad \text{para } i \in Doc \setminus Doc^{cat} \quad (2)$$

onde a quantidade \bar{M} é uma estimativa para média de vigilâncias que deve ser atribuída a um docente não catedrático contratado a 100%, determinada pela fórmula (3).

$$\bar{M} = \left\lfloor \frac{\sum_{j=1}^m V_j - \sum_{i \in Doc^{cat}} \sum_{j=1}^m P_{ij}}{\sum_{i \in Doc \setminus Doc^{cat}} \beta_i - |Doc^{cat}|} \right\rfloor \quad (3)$$

As variáveis do modelo são:

- $x_{i,j} = \begin{cases} 1 & \text{se o docente } i \text{ é afeto à da prova } j \\ 0 & \text{caso contrário} \end{cases}, i \in Doc, j \in Pro$
- $L =$ minorante para o número de vigilâncias a atribuir a um docente com horário completo ($\beta_i = 100\%$), não catedrático com saldo de vigilâncias anteriores não negativo ($S_i \geq 0$).

$$\text{Max } z = L \quad (4)$$

s.a

$$\sum_{j=1}^m x_{ij} = M_i, \quad i \in S^- \quad (5)$$

$$\sum_{j=1}^m x_{ij} \leq M_i, \quad i \in S^+ \quad (6)$$

$$\sum_{j=1}^m x_{ij} + S_i \geq L, \quad i \in S^+ \text{ e } \beta_i = 1 \quad (7)$$

$$\sum_{i=1}^n x_{ij} = V_j, \quad j \in Pro \quad (8)$$

$$x_{ij} + x_{ik} \leq 2 - C_{jk}, \quad j, k \in Pro \quad (9)$$

$$x_{ij} \leq 1 - D_{ij}, \quad i \in Doc, j \in Pro \quad (10)$$

$$x_{ij} \geq P_{ij}, \quad i \in Doc, j \in Pro \quad (11)$$

$$L \geq 0 \text{ e inteiro} \quad (12)$$

$$x_{ij} \in \{0,1\}, \quad i \in Doc, j \in Pro \quad (13)$$

Com a função objetivo (4) pretende-se maximizar L , o número de vigilâncias atribuídas a um docente com horário completo, de modo a promover o equilíbrio na distribuição.

As restrições (5) obrigam a que o número vigilâncias atribuídas a docentes com saldo negativo sejam o máximo determinado, M_i , enquanto as restrições (6) permitem uma atribuição em número inferior a esse máximo. O conjunto de restrições (7) define a variável L . A garantia que vigilâncias atribuídas a um docente são compatíveis decorre do grupo (9). As indisponibilidades são traduzidas por (10) e a obrigatoriedade das vigilâncias próprias por (11). O âmbito das variáveis do modelo é estabelecido por (12) e (13).

O número de variáveis do modelo é igual a $nm + 1$ e o número de restrições está entre $3nm$ e $3nm + m + 3n$. Nos dados fornecidos pelo ISEG para testar o modelo, referentes às épocas normal e de recurso 1º semestre do ano letivo de 2022/23, o número

de docentes foi $n = 52$ e o número de provas $m = 120$, tratando-se assim de uma instância de grandes dimensões.

4.2 V.B.A.

O primeiro passo nesta implementação foi a contagem do número de docentes que devem vigiar provas no período de avaliações selecionado que é guardado numa variável inteira global. Uma outra variável global regista o número de unidades curriculares a considerar neste período de avaliações. De notar que quando estamos a tratar de duas épocas de avaliação em simultâneo como é o caso dos dados de teste fornecidos (Época Normal e Época de Recurso), uma unidade curricular deve ser duplicada, quando existem duas datas a considerar.

O passo seguinte é a verificação da existência, da folha “MAPA FINAL” que irá servir para definir restrições, variáveis de decisão e guardar a matriz com a solução ótima encontrada. No caso existir atualizam-se os dados, caso contrário é criada a folha e emulada a formatação da matriz em que os dados são normalmente disponibilizados.

Passando agora para a fase onde se definem as variáveis de decisão do problema, função objetivo e tipo de otimização, neste caso será de maximização, foi necessário calcular e guardar o valor de cada M_i num vetor de *variants* global, uma vez que o valor é arredondado por defeito.

Estando feitos praticamente todos os cálculos necessários para a implementação das restrições, faltando apenas fazer o upload de alguns inputs, decidiu-se começar pelas restrições (5) e (6) que estabelecem o número de vigilâncias a atribuir máximo ou exato conforme o saldo do docente. Seguem-se as restrições (8) que garantem o número de vigilantes pretendidos em cada prova.

De seguida são então adicionadas ao modelo as restrições referentes às indisponibilidades e vigilâncias próprias de cada docente, as restrições, ou restrição, de integridade que garantem que todas as variáveis de decisão, à exceção do L , são binárias.

Depois as restrições (9) que impedem que o mesmo docente receba de provas de avaliação incompatíveis. Para tal, foi criada uma matriz binária quadrada, com dimensão do número de provas, que no cruzamento da linha de uma prova com a coluna de outra prova tem um caso estas sejam incompatíveis. Na realidade, como a matriz é simétrica, considerou-se apenas uma matriz triangular superior com diagonal zero. Foram ainda adicionadas restrições na forma de somas para as vigilâncias incompatíveis, ou seja, se

por exemplo ocorrerem três vigilâncias incompatíveis seguidas, em vez de adicionar três restrições ao modelo, a 1ª ser incompatível com a 2ª, a 1ª ser incompatível com a 3ª e a 2ª ser incompatível com a 3ª, apenas se adiciona uma restrição, de que a soma das variáveis de decisão $x_{i,j}$ correspondentes tem de ter no máximo valor igual a um.

Terminamos com as restrições (10) que impõem as vigilâncias próprias e (11) têm em conta as indisponibilidades manifestadas.

4.3 Python

Por sua vez, na implementação via *Python* a criação e resolução do modelo em termos de lógica de código acaba por ser semelhante, uma vez que o *script* foi baseado no do V.B.A. É importante notar, antes de prosseguir, que apesar de existirem dois tipos diferentes de criação e resolução do modelo neste *script*, a sua lógica e passos são semelhantes na prática, sendo apenas diferentes devido aos dois tipos de opções de *solvers*, pelos quais o utilizador do G.U.I. poderá optar, utilizarem métodos de criação de modelos diferentes, o *Gurobi* lê ficheiros “Lp” enquanto que o *OR-Tools* não, mas tal irá ser explicado no próximo capítulo.

Primeiramente é criado, ou a já existir é reiniciado, o próprio ficheiro “Lp”, na localização onde for escolhido o ficheiro Excel com os dados, de nome “Modelo-TFM.lp”, já com a formatação necessária para o caracterizar como um modelo de maximização da função objetivo “L”. De seguida é lida, do ficheiro Excel com os dados toda a informação referente aos docentes e provas de avaliação e guardada em duas *DataFrames*, “docentes” e “exames”. O número de docentes e de provas de avaliação em é também guardado duas variáveis, “n_docentes” e “n_UC”.

Posto isto, é altura de recolher os restantes *inputs* do modelo, assim, é criada uma *DataFrame* referente às vigilâncias próprias, “Pij”, para de seguida se calcular a soma referente a cada docente. Essa mesma é depois adicionada a uma nova coluna na *DataFrame* “docentes” chamada “Soma VP”.

O cálculo do número total de vigilâncias é guardado numa variável, “n_Total_Vigilancias”, para mais tarde ser utilizado no cálculo do \bar{M} , que por sua vez será utilizado no cálculo do M_i de cada docente e adicionado mais uma vez à *DataFrame* “docentes” numa nova coluna designada “Mi”. Para tal, é ainda previamente necessário calcular o número de docentes catedráticos e a soma de todas as percentagens horarias dos docentes.

São ainda lidos os dados referentes às indisponibilidades dos docentes e guardados numa *DataFrame* “Dij”. Além disso, é também adicionada à *DataFrame* “docentes” uma nova coluna designada “Li” onde se marca com “Sim” os docentes não catedráticos com horário completo.

Faltando apenas definir o número de vigilâncias por prova de avaliação, tal informação é recolhida e adicionada a uma nova coluna designada “Vj” na *DataFrame* “exames”.

Estando todos os *inputs* e características do modelo apurados e guardados na memória do computador, basta então apenas adicionar as várias restrições e variáveis de decisão binárias, $x_{i,j}$, ao ficheiro “Modelo-TFM.lp”, sendo as restrições numeradas entre si na forma, “c01:” para a primeira restrição, “c02:” para a segunda restrição e assim sucessivamente.

De forma a não adicionar informação duplicada neste relatório, como já se referiram os tipos e subtipos de restrições associadas a este modelo basta-nos apenas afirmar que tal foi executado de acordo com o exposto anteriormente e através da leitura e comparação dos valores das variáveis e *DataFrames* expostos anteriormente.

De forma a concluir então o ficheiro “Lp” do modelo, são adicionados os nomes das variáveis de decisão binárias e o sinal de marcação do final do modelo, “End”.

Estando concluído o ficheiro “Lp” com o modelo, basta-nos então proceder ao seu upload e otimização. Feita essa otimização, a solução ótima irá ser armazenada numa *DataFrame* “Mapa_Otimo” que irá guardar a solução utilizando uma matriz de morfologia semelhante à utilizada na exposição dos dados iniciais do problema. Sendo posteriormente guardados num ficheiro Excel numa localização escolhida pelo utilizador ou num ficheiro de texto que pretende expor de forma resumida a soma do número de vigilâncias por docente, intitulado de “Resultado-Modelo-TFM.txt”.

CAPÍTULO 5: *INTERFACE GRÁFICA*

O presente capítulo descreve as funcionalidades do sistema e as verificações iniciais efetuadas que dão origem a notificações caso se detete algum erro ou incompatibilidade nos dados.

5.1. Caracterização das várias janelas do G.U.I.

O sistema de Escalonamento de Docentes a Vigilâncias requer a sua instalação. Na subseção seguinte explica-se detalhadamente como obter o ficheiro executável responsável pela instalação, e o executável para iniciar o sistema, estando os ícones dos executáveis na Figura 2 e na Figura 3, respetivamente.



Software-ISEG_S
etup.exe

Figura 2 - Executável responsável pela instalação do Sistema

Uma vez instalado o sistema ESCRO, para iniciar, basta, como habitualmente, um duplo clique sobre o ícone representado na Figura 3.



Software
Escalonamento
Docente-Vigilâncias

Figura 3 - Executável responsável por iniciar o Software

Ao entrar, será exibida a janela de boas-vindas, Figura 4. O utilizador poderá continuar com o processo clicando no botão “Seguinte” ou fechar o sistema clicando no botão “Sair”.

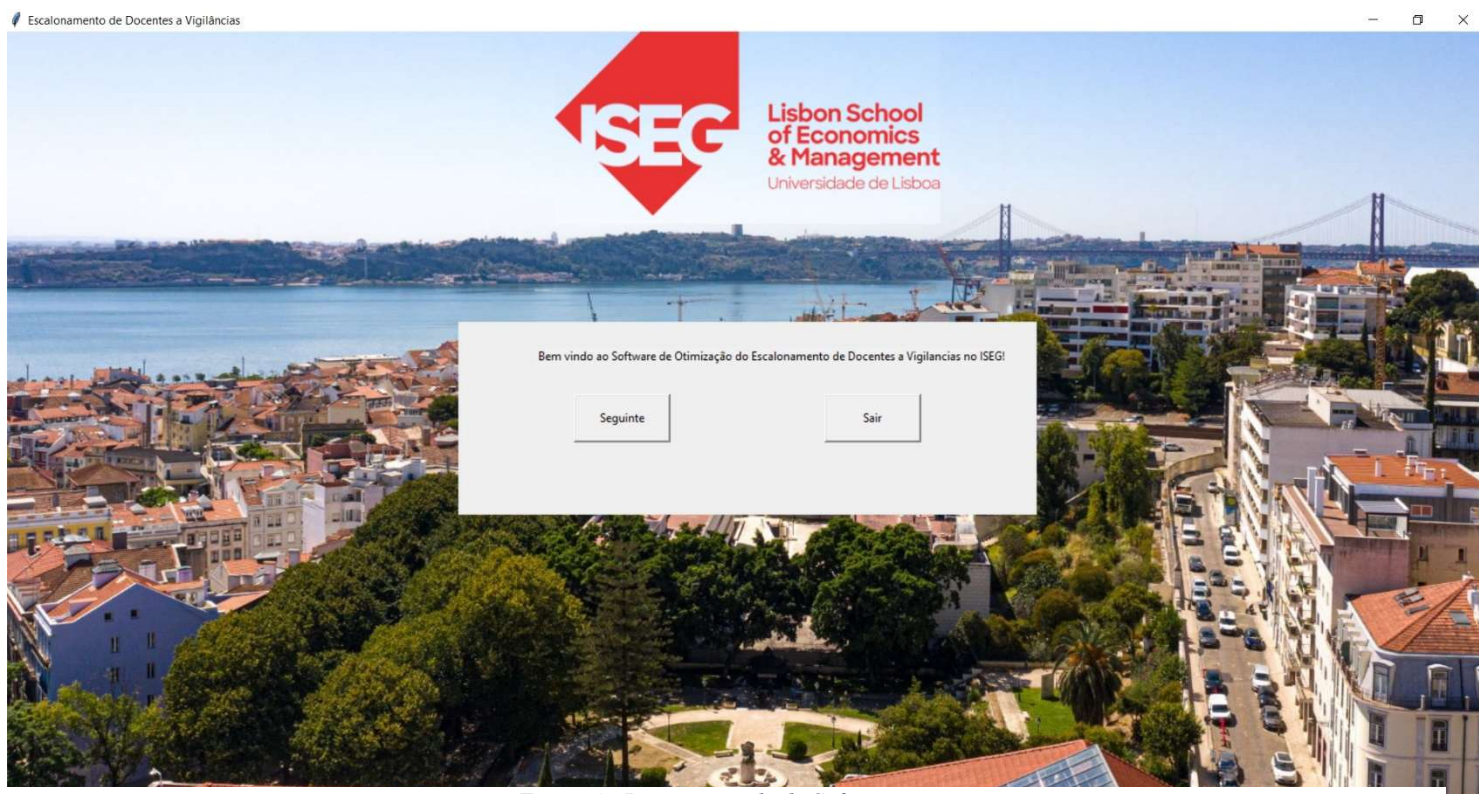


Figura 4 - Primeira janela do Software

Ao escolher continuar, é exibida a segunda janela, ilustrada na Figura 5. Nesta, janela solicita-se a localização do ficheiro Excel, de formato “xlsx”, com os dados do período de avaliação, uma vez introduzido pressiona-se o botão “Alterar”. Para prosseguir clica-se no botão “Seguinte”. Esta ação desencadeia um conjunto de verificações do ficheiro e dos próprios dados sendo o utilizador notificado se forem encontrados erros. Só é possível continuar após a introdução de um ficheiro válido. O utilizador pode também optar por parar o processo, clicando no botão “Sair do interface”.



Figura 5 - Segunda janela do Software

Quanto às notificações que decorrem da escolha de um ficheiro inválido podem ser de quatro tipos.

O primeiro tipo, ilustrado na Figura 6, resulta de se ter selecionado uma localização inválida, ou seja, a própria localização não existe ou simplesmente se introduziu qualquer localização.



Figura 6 - Aviso do sistema, o utilizador não introduziu uma localização válida do ficheiro de dados

O segundo tipo, ilustrado na Figura 7, ocorre quando o ficheiro escolhido não é do tipo “xlsx”. Apenas é possível importar dados com este formato que é o dos dados provenientes do ISEG.

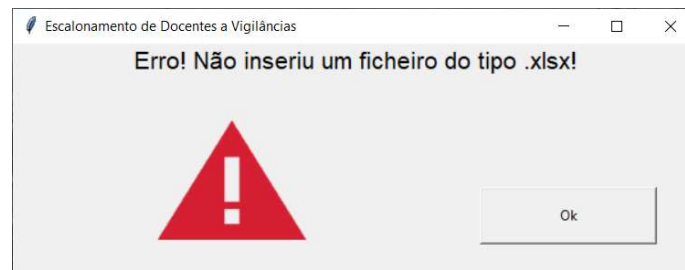


Figura 7 - Aviso do sistema, o ficheiro selecionado não é do tipo .xlsx do Excel

Já o terceiro tipo está relacionado com os próprios dados, ou formatação dos mesmos dentro do caderno do Excel, Figura 8. Neste caso, o tipo de ficheiro está correto, mas os dados não estão estruturados de acordo com o padrão disponibilizado pelo ISEG. Este formato pressupõe a existência de cinco folhas com o nome “Docentes”, “VProprias”, “Exame”, “VigilanciasPedidas” e “Disponibilidades”.

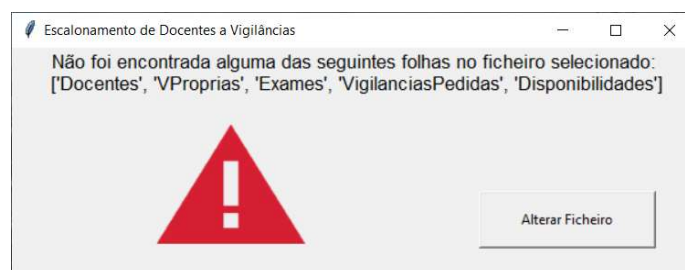


Figura 8 - Aviso do sistema, o ficheiro selecionado não contém alguma das folhas necessárias com os dados

O quarto e último tipo, ilustrado na figura 9, verifica se com os dados introduzidos num ficheiro válido definem um problema com soluções admissíveis. Um exemplo de

situação que impede prosseguir está na Figura 9. Neste caso, o “docente 1” tem uma vigilância própria que o obriga a estar presente e uma indisponibilidade que o impossibilita de estar presente.

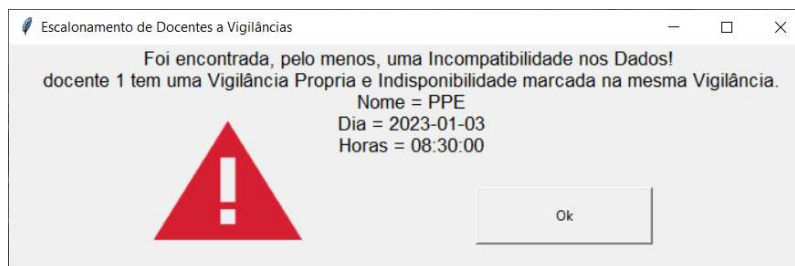


Figura 9 - Aviso do sistema, existe, no mínimo, uma incompatibilidade nos dados do problema

Depois de cumpridas todas as verificações, é exibida a janela ilustrada na Figura 10, que servirá como uma última confirmação dos dados do problema pelo utilizador, tendo por base o número de docentes, número de unidades curriculares do período de avaliação e localização do próprio ficheiro. Feita essa pequena verificação, o utilizador terá então três opções: (1) retroceder à janela anterior, representada pela Figura 5; (2) otimizar o modelo usando o *solver Gurobi*, que requer uma licença de utilização paga ou grátis para docentes e alunos de cursos superiores; ou (3) otimizar o modelo usando o *solver OR-Tools*, um *Freeware* desenvolvido pela *Google*.



Figura 10 - Terceira janela do Software

Ao escolher qualquer um dos dois tipos de otimização o resultado gráfico será igual. Até que o sistema encontre uma solução ótima, o utilizador irá deparar-se com um chamado *waiting screen*, ilustrado na Figura 11, que será basicamente uma janela com um *GIF* de um gráfico circular que irá rodando com o decorrer do tempo e uma mensagem que indica que o processo está a decorrer, ou seja, que o sistema não bloqueou. De notar que não é possível prever qual será a duração do processo uma vez que tal irá depender da própria complexidade do modelo e especificações do próprio computador.



Figura 11 - Janela de espera enquanto se determina uma solução ótima para o problema

Quando o processo termina o utilizador é exibida a quarta e última janela do sistema, ilustrada na Figura 12, com o valor da função objetivo da solução encontrada. Nesta janela são apresentadas quatro opções: (1) fechar simplesmente o programa, perdendo a solução encontrada, (2) retroceder à janela anterior, representada pela Figura 10, (3) ter acesso a um pequeno ficheiro de texto, de formato “txt”, que contém apenas o número de vigilâncias totais atribuídas a cada docente, ou ainda, (4) exportar um ficheiro Excel, de formato “xlsx”, com a matriz de escalonamento encontrada para uma

localização a definir pelo utilizador, com formato igual ao das matrizes com os dados fornecidos pelo ISEG.



Figura 12 - Quarta, e última, janela do Software

5.2. Elaboração do ficheiro Setup do sistema

Estando o sistema a funcionar de forma intuitiva decidiu-se trabalhar no sentido de melhorar a sua acessibilidade. Inicialmente, era obrigatório ter numa mesma localização as ilustrações, *script* do *Python* e o ficheiro com os dados. Esta condicionante para ser alterada exigiria do utilizador conhecimentos de programação, nomeadamente em *Python*.

Para contornar essas dificuldades foi necessário recorrer a duas ferramentas externas para criar um ficheiro executável que instala o sistema com todos os seus componentes, e permite utilizá-lo sem recorrer a um editor de código.

Em primeiro lugar, o conjunto de *scripts Python* foi convertido num ficheiro executável. Para tal utilizou-se o módulo “*auto-py-to-exe*”, existindo várias opções a definir, dependendo do intuito do utilizador, como por exemplo quais as componentes internas que o *script* utiliza, se é pretendido que a linha de comandos seja visível ou não durante a execução do mesmo ou se pretende definir um ícone personalizado para o

executável. A título de curiosidade, no presente caso foi necessário importar uma biblioteca invisível no código, a “openpyxl.cell._writer”.

Estando criado o executável que inicia o *software*, falta apenas automatizar a criação do executável (instalador) que permite a partilha e instalação do mesmo. Para tal, recorremos a uma aplicação gratuita chamada “*Inno Setup Compiler*”, ícone representado pela Figura 13, que nos permite chegar ao executável pretendido. Durante esse processo definimos vários requisitos tais como que linguagem, ou linguagens, pode o instalador utilizar, quem é o autor do *software*, qual a sua versão ou qual deve ser a localização da instalação do *software*, sendo a Figura 2 uma representação do executável de instalação obtido.



Figura 13 - Executável do Inno Setup Compiler

CAPÍTULO 6: ANÁLISE COMPARATIVA DE RESULTADOS

6.1. Análise Qualitativa das soluções ótimas: V.B.A. vs *Gurobi* vs *OR-Tools*

Numa tentativa de comparar as três soluções ótimas alternativas, via *Gurobi*, *OR-Tools* e V.B.A., optou-se por recorrer a características da solução que não são tidas em conta na formulação do modelo.

No presente caso, avaliaram-se as soluções quanto à distribuição do número de vigilâncias diárias por docente. Apesar de discutível, considera-se ideal uma solução em que todos os docentes têm no máximo apenas uma vigilância por dia. Neste sentido, de modo a promover a obtenção de soluções ideais, introduziram-se alterações no modelo inicial ((4) a (13)) que se resolveu registando-se assim uma quarta solução. A solução que provém deste modelo alterado designar-se-á por M. Alterado.

Antes de comparar as quatro soluções, via *Gurobi*, *OR-Tools*, V.B.A. e com o M. Alterado, apresentamos as modificações operadas no modelo. Em primeiro lugar, e para garantir a optimalidade do ponto de vista do modelo ((4) a (13)), fixou-se o valor de L , isto é, escreveram-se as restrições (7) com $L = 6$, valor ótimo do modelo. Em segundo lugar, foi necessário definir um conjunto de variáveis u_d inteiras que significam o número máximo de vigilâncias que é atribuído a um docente no dia d . Note-se que, como é conhecido o dia em que se realiza cada prova, pode ser definida uma partição no conjunto Pro , o conjunto de provas, por dias. Para esse efeito, designe-se por:

$$Pro_d = \{j \in Pro: j \text{ se realiza no dia } d\} \quad d \in D,$$

onde D é o conjunto de dias da época de provas. Podem então escrever-se as restrições

$$\sum_{j \in Pro_d}^m x_{ij} \leq u_d \quad i \in Doc, d \in D \quad (14)$$

Por fim, a função objetivo do modelo alterado é:

$$\min z_A = \sum_{d \in D} u_d$$

O detalhe das soluções que agora se comparam encontra-se no Anexo III. A Figura 15 tem a solução do *Gurobi*, a Figura 16 a solução do *OR-Tools*, a Figura 17 diz respeito à solução do *OpenSolver*, e, por fim, na Figura 18 pode observar-se a solução obtida com o modelo alterado.

As vigilâncias de cada docente agrupadas por dia para as quatro soluções, constam no Anexo III. Mais concretamente, a Figura 19 corresponde à solução do *Gurobi*, a Figura 20 à do *OR-Tools*, a Figura 21 à do V.B.A e a Figura 22 à do modelo alterado. A Tabela 2 do Anexo III agrega a informação das quatro soluções.

Para comparar os resultados resumiu-se toda esta informação na Tabela 1.

	<i>Gurobi</i>	<i>OR-Tools</i>	<i>VBA</i>	<i>M.Alterado.</i>
<i>N. de ocorrências com:</i>				
<i>0 vigilâncias diárias</i>	868	863	863	856
<i>1 vigilância diária</i>	198	208	209	222
<i>2 vigilâncias diárias</i>	26	21	19	14
<i>3 vigilâncias diárias</i>	0	0	1	0
<i>N. de docentes com 2 ou mais vigilâncias diárias mais de 1 vez</i>	5	2	2	2

Tabela 1 - Resumo das várias soluções ótimas encontradas

Nas quatro primeiras linhas da Tabela 1, os valores dizem respeito ao número de ocorrências registadas tendo em atenção o número de vigilâncias dos docentes por dia. Assim, o valor na primeira linha e na coluna *Gurobi* significa que nesta solução somando para cada dia da época o número de docentes sem vigilâncias atribuídas se obtém 868. Na segunda linha, por exemplo na coluna *OR-Tools* temos 208 que é a soma do número de docentes que tem uma vigilância para o conjunto de dias da época se usada a solução do *OR-Tools*. A terceira linha refere-se soma do número de repetições de docentes ao longo dos dias da época, enquanto a quarta linha tem o mesmo registo, mas considerando 3 vigilâncias no mesmo dia. A última linha conta o número de docentes que foi escalonado para duas vigilâncias ou mais, mais do que um dia da época.

A Figura 14 representa o acréscimo percentual do número de ocorrências em que há repetição de docentes por dia, de cada uma das soluções face à solução do modelo alterado. Na observação da figura é facilmente perceptível que existe uma diferença percentual significativa das três soluções iniciais: o V.B.A. regista um aumento percentual de aproximadamente 42,88%, o *OR-Tools* um aumento de 50% e o *Gurobi* um aumento aproximado de 85,71%. Sendo, portanto, o V.B.A. que apresenta a solução inicial com menor aumento seguida de perto pela solução do *OR-Tools*.

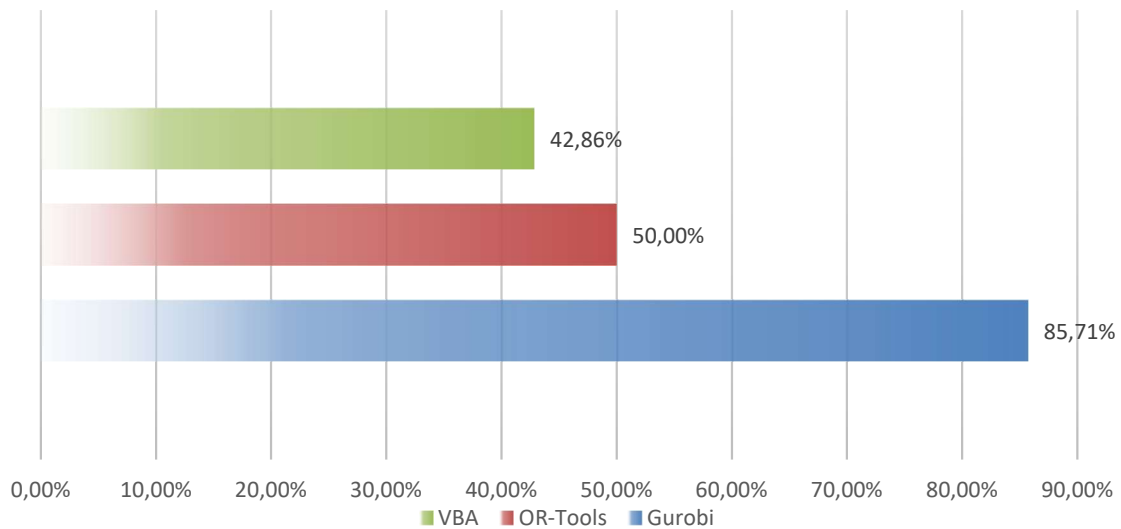


Figura 14 - Gráfico ilustrativo do aumento percentual do número de ocorrências de docentes com mais de uma vigilância diária, face à nova solução ótima

Observando agora os valores da última linha da Tabela 1, verifica-se que apenas a solução do *Gurobi* difere da solução do modelo alterado, apresentado cinco docentes, em vez de dois com vigilâncias repetidas em mais de um dia.

Note-se também que apenas uma das soluções, a solução do V.B.A., regista soluções em que há dias com docentes com três vigilâncias, neste caso apenas um.

6.2. Comparação da facilidade de implementação do código: V.B.A. vs *Python*

Iremos agora proceder à análise de resultados com ênfase na facilidade de implementação do próprio código que deu origem ao modelo e que permite a sua posterior otimização. Como referido anteriormente, foram utilizadas as linguagens de programação

V.B.A. e *Python*, e enumeraremos agora algumas das dificuldades sentidas durante a elaboração deste projeto.

Tendo em conta que os dados representativos do problema de escalonamento estão guardados num ficheiro Excel, seria natural esperar que o V.B.A. tivesse alguma vantagem inicial no processo, devido à compatibilidade com os dados armazenados no ficheiro. No entanto, a forma de escrita, ou lógica, da própria implementação do código é bastante mais intuitiva no *Python*. Por exemplo, no caso do cálculo das somas das variáveis de decisão, $x_{i,j}$, utilizadas na implementação das restrições relativas às incompatibilidades, o *Python* possui uma sintaxe mais concisa e direta. Em contraste, no V.B.A., é necessário armazenar o valor dessas mesmas somas em células individuais da folha onde será guardado o escalonamento ótimo, o que pode complicar a implementação e tornar o resultado visualmente menos atraente e mais complexo, para utilizadores não familiarizados com a programação.

Além disso, o *Python* é uma linguagem de programação amplamente utilizada, o que resulta numa vasta gama de bibliotecas e ferramentas disponíveis, oferecendo várias alternativas para atender às necessidades de implementação. Isso facilita a criação e resolução do modelo de otimização em *Python*, como evidenciado pela diferença significativa no número de linhas de código necessárias para ler e criar as restrições das incompatibilidades, no caso do V.B.A., linhas 314 a 427 do Anexo I, e no caso do *Python*, linhas 230 a 271 do Anexo II, uma diferença significativa.

Por outro lado, uma das vantagens do V.B.A. em relação ao *Python* é a capacidade de compactar o número de arquivos necessários para obter uma solução ótima. Ao armazenar o *script* e os dados num único ficheiro Excel, o utilizador só precisaria de um ficheiro, enquanto na implementação em *Python* seriam necessários, pelo menos, dois ficheiros: o ficheiro de dados e o ficheiro com o *script* ou *software*.

6.3. Análise comparativa de tempo de execução: V.B.A. vs *Python*

O tempo total de execução entre V.B.A. e *Python* é um aspeto importante para avaliar e comparar os diferentes métodos utilizados neste projeto. Contudo, nesta avaliação há que ter em consideração que a duração do processo de otimização é influenciada principalmente pelos componentes do computador onde a execução ocorre.

Para realizar a comparação, foram registados os tempos de execução médios obtidos a partir de cinco rondas de otimização para cada método. Os testes foram realizados num computador equipado com um processador Intel i7-101750H, um SSD NVMe de 1TB com velocidades de leitura e escrita de 3500 e 2100 MB/s, respetivamente, e 16GB de RAM DDR4.

No caso do V.B.A., o tempo médio de execução foi de aproximadamente 2 minutos e 10 segundos. No que toca ao *Python*, uma vez que podem ser utilizados dois tipos de *solvers* diferentes, optou-se por medir os seus tempos médios de execução em separado. Utilizando o *Gurobi*, os tempos médios de execução foram consideravelmente reduzidos em comparação com o V.B.A., com uma média de 1 minuto e 4 segundos. Por sua vez, a utilização do *OR-Tools* proporcionou um desempenho ainda melhor, com um tempo médio de execução de cerca de 35 segundos. Essa redução significativa no tempo de execução em relação ao V.B.A. de ambos os *solvers* utilizados na implementação em *Python* destaca a eficiência para resolver o problema de escalonamento desta implementação.

É importante ressaltar que esses resultados são específicos para o *hardware* e as configurações do computador utilizado nos testes. A duração do processo de otimização pode variar em diferentes sistemas, dependendo dos recursos disponíveis.

CAPÍTULO 7: CONCLUSÃO

Neste projeto, foi construído o sistema ESCPRO, para resolver o problema de escalonamento de docentes a vigilâncias no ISEG. Formulou-se um modelo de programação linear inteira e desenvolveu-se um G.U.I. para automatizar todo o processo de distribuição de vigilâncias aos docentes. tendo sido explorados três *softwares* para a resolução do problema de PLIM: *Gurobi*, a biblioteca *OR-Tools* e o *OpenSolver* do Excel.

De forma a comparar as soluções obtidas foi assumido que idealmente o mapa de vigilâncias não devia atribuir mais do que uma vigilância a cada docente por dia. Para promover soluções com estas características foram introduzidas alterações no modelo inicial, obtendo-se com a resolução do modelo alterado uma quarta solução.

Após a comparação foi possível concluir que a solução *OR-Tools* seria a melhor, mas há que ter em consideração que esta conclusão depende não só dos dados de teste como também da aleatoriedade da solução obtida entre os ótimos alternativos.

É de salientar que o modelo alterado, pode proporcionar uma melhoria se implementado ESCPRO, caso se entenda ideal minimizar o número de vigilâncias atribuídas a um docente no mesmo dia.

Em relação à facilidade de implementação do código, a linguagem *Python* apresentou claramente vantagens sobre o V.B.A., pois sendo a sua sintaxe mais concisa e direta tornou mais simples tanto a implementação de restrições como os restantes cálculos.

Para além disso, o elevado e diversificado número de bibliotecas e ferramentas disponíveis ao programador de *Python*, acabara por culminar numa implementação mais eficiente e flexível, sendo claramente a opção mais adequada.

Quanto ao tempo de execução, o V.B.A. apresentou um tempo médio de execução de aproximadamente 2 minutos e 10 segundos, tendo tanto o *Gurobi* como o *OR-Tools* reduzido significativamente o tempo de execução, cerca de 1 minuto e 4 segundos e 35 segundos, respetivamente. Estes resultados evidenciam mais uma vez a vantagem, agora em tempo de execução, que o *Python* proporciona, sobretudo com o *OR-Tools*.

Em conclusão, este projeto desenvolveu-se um SAD, ESCPRO, em *Python*, que automatizou o processo de otimização do problema de escalonamento

Recomenda-se que futuros estudos explorem outras características de avaliação e acrescentem ao modelo inicial restrições adicionais, a traduzir as preferências, grau proximidade dos docentes com unidades curriculares das provas, por exemplo.

REFERÊNCIAS BIBLIOGRÁFICAS

- Baker, K. R., 1976. *Workforce allocation in cyclical scheduling problems: A survey. Operational Research Quarterly* 27, pp. 155-167.
- Bouajaja, Sana & Dridi, Najoua. 2017. *A survey on human resource allocation problem and its applications. Operational Research* 17, pp. 339-369.
- Breslaw, J. A., 1976. *A linear programming solution to the faculty assignment problem. Socio-Economic Planning Science* 10, pp. 227-230.
- Brucker, P., Qu, R., Burke, E., & Post, G., 2005. *A Decomposition, Construction and Post-Processing Approach for Nurse Rostering. Proceeding of the 2nd Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA'05)*, pp. 397-406.
- Burke, E., De Causmaecker, P., Berghe, G. V., & Van Landeghem, H., 2004. *The state of the art of nurse rostering. Journal of Scheduling*.
- Dantzig, G., 1954. *A Comment on Edie's "Traffic Delays at Toll Booths". Journal of the Operations Research Society of America* 2, pp. 339-341.
- Ernst, A., Jiang, H., Krishnamoorthy, M. & Sier, D., 2004. *Staff scheduling and rostering: A review of applications, methods and models. European Journal of Operational Research* 153, pp. 3-27.
- Hast, J., 2017. *Optimal work shift scheduling: a heuristic approach. Aalto University, School of Science*.
- ISEG, 2023. Página online do Instituto Superior de Economia e Gestão. Acedido em 26 de abril de 2023. <https://www.iseg.ulisboa.pt/sobre/historia/>
- ISEG, 2023. Imagem ilustrativa da Planta do ISEG. ISEG Handbook, Versão 1, pp. 10.
- Van Den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E., & De Boeck, L., 2013. *Personnel Scheduling: A literature review. European Journal of Operational Research*, 226(3), pp. 367-385.

ANEXOS

ANEXO I (V.B.A)

```
Option Explicit
```

```
Dim n_docentes As Integer, n_UC As Integer, contador As Integer,
```

```
Mi() As Variant
```

```
Dim intervalo As Range
```

```
Dim docentes_Li() As String
```

```
Dim celula As Range
```

```
Dim Saldo_Vigilancias() As Integer
```

```
Sub main()
```

```
    Call contar_docentes
```

```
    Call contar_UC
```

```
    Call Sheet_Final
```

```
    Call Definir_L
```

```
    Call Calculo_M
```

```
    Call Restricoes_Mi
```

```
    Call Restricoes_Vj
```

```
    Call Restricoes_0
```

```
    Call Restricoes_1
```

```
    Call Restricoes_Binarias
```

```
    Call Restricoes_Cjk
```

```
    Call Restricoes_L
```

```
    OpenSolver.RunOpenSolver
```

```
End Sub
```

```
Public Sub Sheet_Final()
```

```
    Dim ws As Worksheet
```

```
    contador = 0
```

```
    For Each ws In ThisWorkbook.Worksheets
```

```
        If ws.Name = "MAPA FINAL" Then
```

```
        contador = contador + 1
    End If
Next

    If contador = 1 Then
        Worksheets("MAPA FINAL").Activate
        SolverReset
    Else
        Worksheets.Add After:=Worksheets(Worksheets.Count)
        ActiveSheet.Name = "MAPA FINAL"

        'Copiar Vigilancias

ThisWorkbook.Sheets("VProprias").Range(ThisWorkbook.Sheets("VPropri
as").Cells(1, 3), ThisWorkbook.Sheets("VProprias").Cells(5, 3 +
n_UC - 1)).Copy ThisWorkbook.Sheets("MAPA
FINAL").Range(ThisWorkbook.Sheets("MAPA FINAL").Cells(1, 3),
ThisWorkbook.Sheets("MAPA FINAL").Cells(5, 3 + n_UC - 1))
        'Copiar Docentes

ThisWorkbook.Sheets("VProprias").Range(ThisWorkbook.Sheets("VPropri
as").Cells(6, 1), ThisWorkbook.Sheets("VProprias").Cells(6 +
n_docentes - 1, 2)).Copy ThisWorkbook.Sheets("MAPA
FINAL").Range(ThisWorkbook.Sheets("MAPA FINAL").Cells(6, 1),
ThisWorkbook.Sheets("MAPA FINAL").Cells(6 + n_docentes - 1, 2))

    End If

End Sub

Public Sub contar_docentes()

Dim Ultima_Linha As Integer
Dim i As Integer

Worksheets("VProprias").Activate

'Defenir o nº de docentes no problema
```



```
Ultima_Linha = Cells(Rows.Count, "A").End(xlUp).Row 'Encontra a  
ultima linha com informação na coluna A
```

```
contador = 0 ' "Reiniciar" valor do contador
```

```
For i = 1 To Ultima_Linha ' Percorre as linhas da coluna A e diz  
quantas tem informação
```

```
    If Not IsEmpty(Cells(i + 5, "A")) Then
```

```
        contador = contador + 1
```

```
    End If
```

```
Next i
```

```
n_docentes = contador 'Guardar o n° de docentes em determinado  
problema
```

```
End Sub
```

```
Public Sub contar_UC()
```

```
Dim Ultima_Coluna As Integer
```

```
Dim i As Integer
```

```
Worksheets("VProprias").Activate
```

```
'Defenir o n° de UC no problema
```

```
contador = 0 '"Reiniciar" valor do contador
```

```
Ultima_Coluna = Cells(4, Columns.Count).End(xlUp).Column 'Encontra  
a ultima coluna com informação na linha 4
```

```
For i = 1 To Ultima_Coluna ' Percorre as linhas da linha 4 e diz  
quantas tem informação
```

```
    If Not IsEmpty(Cells(4, i)) Then
```

```
        contador = contador + 1
```

```
    End If
```

```
Next i
```

```
n_UC = contador 'Guardar o n° UC em determinado problema
```

```
End Sub
```

```
Public Sub Calculo_M()  
Dim Total_Vigilancias As Integer, Total_Vigilancias_Pedidas As  
Integer, Total_Vigilancias_Proprias As Integer, soma As Integer  
Dim n_Docentes_Cat As Integer, Vigilancias_Proprias_Cat() As  
String, Soma_VP_Cat As Integer, j As Integer,  
Soma_Docente_Percentagem As Double  
Dim M As Double  
Dim i As Integer
```

```
Worksheets("VigilanciasPedidas").Activate
```

```
For i = 1 To n_UC  
Total_Vigilancias_Pedidas = Total_Vigilancias_Pedidas +  
Application.Cells(6, i + 1) 'Soma das Vigilancias Pedidas  
Total_Vigilancias_Proprias = Total_Vigilancias_Proprias +  
Application.Cells(7, i + 1) 'Soma das Vigilancias Proprias  
Next i
```

```
Total_Vigilancias = Total_Vigilancias_Pedidas +  
Total_Vigilancias_Proprias
```

```
Worksheets("Docentes").Activate
```

```
For i = 1 To n_UC  
If Application.Cells(i, 3) = "Cat" Then  
n_Docentes_Cat = n_Docentes_Cat + 1  
End If  
Next i
```

```
ReDim Vigilancias_Proprias_Cat(1 To n_Docentes_Cat)
```

```
For i = 1 To n_UC  
If Application.Cells(i, 3) = "Cat" Then  
soma = soma + 1  
Vigilancias_Proprias_Cat(soma) = Application.Cells(i, 1)  
End If
```

```
Next i

For i = 1 To n_docentes
    Soma_Docente_Percentagem = Soma_Docente_Percentagem +
Application.Cells(i + 1, 4)
Next i

    Worksheets("VProprias").Activate

For i = 1 To n_Docentes_Cat
    For j = 1 To n_docentes
        If Application.Cells(j + 5, 1) =
Vigilancias_Proprias_Cat(i) Then
            Soma_VP_Cat = Soma_VP_Cat + Application.Cells(j + 5, 2)
        End If
    Next j
Next i

M = ((Total_Vigilancias - Soma_VP_Cat) / (Soma_Docente_Percentagem
- n_Docentes_Cat))

Worksheets("Docentes").Activate

ReDim Mi(1 To n_docentes)
ReDim Saldo_Vigilancias(1 To n_docentes)

For i = 1 To n_docentes
    Saldo_Vigilancias(i) = Application.Cells(i + 1, 5)
    Mi(i) = WorksheetFunction.RoundDown(M * Application.Cells(i +
1, 4) - Saldo_Vigilancias(i), 0)
Next i

End Sub

Public Sub Restricoes_Mi()

Dim j As Integer
Dim i As Integer
Dim Valor_VP() As Variant
```

```
ReDim Valor_VP(1 To n_docentes)

Worksheets("VProprias").Activate

For i = 1 To n_docentes
    Valor_VP(i) = Application.Cells(i + 5, 2)
Next i

Worksheets("MAPA FINAL").Activate

For i = 1 To n_docentes
    If Valor_VP(i) <= Mi(i) And Saldo_Vigilancias(i) < 0 Then
        SolverAdd CellRef:=Cells(i + 5, 2), Relation:=2,
FormulaText:="=" & Mi(i) 'G4 = I4"
    ElseIf Valor_VP(i) <= Mi(i) And Saldo_Vigilancias(i) >= 0 Then
        SolverAdd CellRef:=Cells(i + 5, 2), Relation:=1,
FormulaText:="=" & Mi(i) 'G4 <= I4"
    ElseIf Valor_VP(i) > Mi(i) And Saldo_Vigilancias(i) < 0 Then
        SolverAdd CellRef:=Cells(i + 5, 2), Relation:=2,
FormulaText:="=" & Valor_VP(i) 'G4 = I4"
    ElseIf Valor_VP(i) > Mi(i) And Saldo_Vigilancias(i) >= 0 Then
        SolverAdd CellRef:=Cells(i + 5, 2), Relation:=1,
FormulaText:="=" & Valor_VP(i) 'G4 <= I4"
    End If
Next i

'O tipo de entradas do CellRef e FormulaText tem que ser celulas ou
range de celulas ou tipo Variant

End Sub

Public Sub Restricoes_Vj()

Dim Vj() As Variant
Dim i As Integer

ReDim Vj(1 To n_UC)
```

```
Worksheets("VigilanciasPedidas").Activate

For i = 1 To n_UC
    Vj(i) = Application.Cells(8, i + 1)
Next i

Worksheets("MAPA FINAL").Activate

For i = 1 To n_UC
    SolverAdd CellRef:=Cells(5, i + 2), Relation:=2,
FormulaText:="=" & Vj(i) 'G6 = I6"
Next i

End Sub

Public Sub Restricoes_0()

Dim Dij() As Integer, j As Integer
Dim i As Integer

ReDim Dij(1 To n_docentes, 1 To n_UC)

    Worksheets("Disponibilidades").Activate

For i = 1 To n_docentes
    For j = 1 To n_UC
        Dij(i, j) = Application.Cells(i + 5, j + 1)
    Next j
Next i

    Worksheets("MAPA FINAL").Activate

For i = 1 To n_docentes
    For j = 1 To n_UC
        If Dij(i, j) = 1 Then
            SolverAdd CellRef:=Cells(i + 5, j + 2), Relation:=2,
FormulaText:="0" 'G6 = I6"
        End If
    Next j
```

```
Next i
'Quando  $D_{ij} = 1$ ,  $X_{ij} = 0$ 

End Sub

Public Sub Restricoes_1()

Dim Pij() As Integer, j As Integer
Dim i As Integer

ReDim Pij(1 To n_docentes, 1 To n_UC)

    Worksheets("VProprias").Activate

For i = 1 To n_docentes
    For j = 1 To n_UC
        Pij(i, j) = Application.Cells(i + 5, j + 2)
    Next j
Next i

    Worksheets("MAPA FINAL").Activate

For i = 1 To n_docentes
    For j = 1 To n_UC
        If Pij(i, j) = 1 Then
            SolverAdd CellRef:=Cells(i + 5, j + 2), Relation:=2,
FormulaText:="1" 'G6 = I6"
        End If
    Next j
Next i

'Quando  $P_{ij} = 1$ ,  $X_{ij} = 1$ 

End Sub

Public Sub Restricoes_Binarias()

    Worksheets("MAPA FINAL").Activate
```

```
SolverAdd CellRef:=Cells(6, 3).Resize(n_docentes, n_UC),
Relation:=5, FormulaText:="binário"

End Sub

Public Sub Restricoes_Cjk()

    Dim Cjk() As Integer
    Dim j As Integer, k As Integer, i As Integer, n As Integer
    Dim leitor() As String
    Dim primeiro As Integer, ultimo As Integer, memoria_ultimo As
Integer
    Dim incompativel As Integer
    Dim vetor() As Integer

    ReDim Segundo_Membro(1 To n_UC, 1 To n_UC)
    ReDim Cjk(1 To n_UC, 1 To n_UC)
    ReDim leitor(1 To n_UC, 3)
    'coluna1=NOME;coluna2=Data;coluna3=Hora

    Worksheets("Exames").Activate

    For i = 1 To n_UC ' linhas
        For k = 1 To 3 ' dimensões do vetor leitor()
            For j = 1 To 5 ' j=5 é a coluna do horário real
                If Not IsEmpty(Application.Cells(i, 5)) And k = 3
Then 'Da hora real, se existir, ao vetor
                    leitor(i, k) = Application.Cells(i, 5)
                ElseIf k = 3 Then ' Da hora ao vetor
                    leitor(i, k) = Application.Cells(i, 2)
                ElseIf k = 2 And j = 1 Then ' Da Data ao vetor
                    leitor(i, k) = Application.Cells(i, j)
                ElseIf k = 1 And j = 4 Then ' Da Nome da UC ao
vetor
                    leitor(i, k) = Application.Cells(i, j)
                End If
            Next j
        Next k
    Next i
```

```
Worksheets("MAPA FINAL").Activate

For j = 1 To n_UC
    For k = 1 To n_UC
        If leitor(j, 2) = leitor(k, 2) And leitor(j, 3) =
leitor(k, 3) And j > k Then
            ' se datas = e horas =, com j > k
            Cjk(j, k) = 1
            incompativel = incompativel + 1

            ElseIf leitor(j, 2) = leitor(k, 2) And (leitor(j, 3) +
(2 / 24)) > leitor(k, 3) And leitor(j, 3) < leitor(k, 3) And j > k
Then
                ' se datas = e horas j + 120 minutos > k, com hora de j
< hora de k, e j > k
                Cjk(j, k) = 1
                incompativel = incompativel + 1

                ElseIf leitor(j, 2) = leitor(k, 2) And (leitor(k, 3) +
(2 / 24)) > leitor(j, 3) And leitor(j, 3) > leitor(k, 3) And j > k
Then
                    ' se datas = e horas k + 120 minutos > j, com hora de j
> hora de k, e j > k
                    Cjk(j, k) = 1
                    incompativel = incompativel + 1

            End If
        Next k
    Next j

ReDim vetor(1 To 2, 1 To incompativel)

For i = 1 To incompativel
    For j = 1 To n_UC
        For k = 1 To n_UC
            If Cjk(j, k) = 1 And j = i Then
                If ultimo < j Then
                    ultimo = j
                End If
            End If
        Next k
    Next j
Next i
```



```

        If primeiro < k And primeiro < ultimo Then
            primeiro = k
        End If
    End If
End If
Next k

If primeiro <> 0 And ultimo <> 0 Then
    vetor(1, i) = primeiro
    vetor(2, i) = ultimo
End If
Next j
Next i

For i = 1 To imcompativel
    For k = imcompativel To 1 Step -1
        If vetor(1, i) = vetor(1, k) And vetor(2, k) > vetor(2,
i) And memoria_ultimo < vetor(2, k) Then
            memoria_ultimo = vetor(2, k)
            For n = 1 To n_docentes
                Application.Cells(n + 5, i + 3 + n_UC) =
"=Sum(" & Cells(n + 5, vetor(1, i) + 2).Address & ":" & Cells(n +
5, vetor(2, k) + 2).Address & ")"
                SolverAdd CellRef:=Cells(n + 5, i + 3 + n_UC),
Relation:=1, FormulaText:="1" 'G4 <= I4"
            Next n
        ElseIf vetor(1, i) = vetor(1, k) And i > k And
memoria_ultimo < vetor(2, k) Then
            memoria_ultimo = vetor(2, i)
            For n = 1 To n_docentes
                Application.Cells(n + 5, i + 3 + n_UC) =
"=Sum(" & Cells(n + 5, vetor(1, i) + 2).Address & ":" & Cells(n +
5, vetor(2, i) + 2).Address & ")"
                SolverAdd CellRef:=Cells(n + 5, i + 3 + n_UC),
Relation:=1, FormulaText:="1" 'G4 <= I4"
            Next n
        End If
    Next k
Next i

```

```
End Sub

Public Sub Restricoes_L()

Dim soma As Integer, j As Integer, k As Integer
Dim i As Integer

'Soma de todos Xij, para todos os docentes que tem %=1 e são não
catedraticos, tem que ser >= Li(minorante para este docente i)

    Worksheets("Docentes").Activate

    For i = 1 To n_docentes
        If Application.Cells(i + 1, 3) <> "Cat" And
Application.Cells(i + 1, 4) = "1" Then
            soma = soma + 1
        End If
    Next i

    ReDim docentes_Li(1 To soma)

    soma = 0

    For i = 1 To n_docentes
        If Application.Cells(i + 1, 3) <> "Cat" And
Application.Cells(i + 1, 4) = "1" Then
            soma = soma + 1
            docentes_Li(soma) = Application.Cells(i + 1, 1)
        End If
    Next i

    Worksheets("MAPA FINAL").Activate

    For i = 1 To n_docentes
        For k = 1 To soma
            If docentes_Li(k) = Application.Cells(i + 5, 1) And
Saldo_Vigilancias(i) = 0 Then
```

```
        SolverAdd CellRef:=Cells(i + 5, 2), Relation:=3,  
FormulaText:=celula.Address 'G8 >= I8  
        Exit For  
        ElseIf docentes_Li(k) = Application.Cells(i + 5, 1) And  
Saldo_Vigilancias(i) > 0 Then  
            Application.Cells(n_docentes + 11 + i, 3) = "=" &  
celula.Address & " - " & Saldo_Vigilancias(i)  
            SolverAdd CellRef:=Cells(i + 5, 2), Relation:=3,  
FormulaText:=Cells(n_docentes + 11 + i, 3).Address 'G8 >= I8  
            Exit For  
        End If  
    Next k  
Next i
```

End Sub

```
Public Sub Definir_L()
```

```
    Worksheets("MAPA FINAL").Activate
```

```
    Set intervalo = Cells(6, 3).Resize(n_docentes, n_UC) 'Define o  
intervalo das variaveis de decisão Xij
```

```
    Set celula = Application.Cells(n_docentes + 7, 3)
```

```
    Application.Cells(n_docentes + 8, 2) = "F.O."
```

```
    Application.Cells(n_docentes + 7, 2) = "L"
```

```
    Application.Cells(n_docentes + 8, 3) = "=" & celula.Address
```

```
    Set intervalo = Union(intervalo, celula)
```

```
    SolverOk SetCell:=Cells(n_docentes + 8, 3), MaxMinVal:=1,  
ValueOf:=0, ByChange:=intervalo.Address, Engine:=2,  
EngineDesc:="Simplex LP"
```

End Sub

ANEXO II (Python)

```
import os
import tkinter as tk
from tkinter import filedialog
import pandas as pd
import math
import datetime
import gurobipy as gp
from ortools.linear_solver import pywraplp
from PIL import Image, ImageTk
import threading
import time
import openpyxl
import winsound
import sys
import numpy as np

#https://stackoverflow.com/questions/31836104/pyinstaller-and-
#onefile-how-to-include-an-image-in-the-exe-file
def resource_path(relative_path):
    """ Get absolute path to resource, works for dev and for
    PyInstaller """
    try:
        # PyInstaller creates a temp folder and stores path in
        _MEIPASS
        base_path = sys._MEIPASS
    except Exception:
        base_path = os.path.abspath(".")

    return os.path.join(base_path, relative_path)

def tocar_som(som_local):
    winsound.PlaySound(som_local, winsound.SND_FILENAME)

def espera():
    global janela_espera
    janela_espera = tk.Toplevel(janela_3) #Cria janela filha da
    janela_3
    janela_espera.title("Escalonamento de Docentes a Vigilâncias")
```

```

janela_espera.geometry("600x300")
janela_espera.geometry("+460+295")
image = Image.open(resource_path("spinner.gif"))
frames = [] #Lista que vai guardar os frames do GIF
duration = image.info['duration'] // 2 #Defenir a velocidade do
GIF

# Tirar cada frame e adicionar à lista
try:
    while True:
        frames.append(ImageTk.PhotoImage(image))
        image.seek(len(frames)) # mudar frame
        image.info['duration'] = duration
except EOFError:
    pass

# Criar label para meter Gif
label = tk.Label(janela_espera)
label.pack()
texto = tk.Label(janela_espera, text="Por favor aguarde a
Otimização do Problema.",font=("", 16))
texto.pack()
label.place(x=175, y=40)
texto.place(x=90, y=10)

# Função que passa os frames do Gif
def passar_frames(frame):
    label.configure(image=frame) #Problema está aqui
    janela_espera.after(duration, passar_frames,
frames[(frames.index(frame) + 1) % len(frames)])

    passar_frames(frames[0])

janela_espera.update()

def otimizacao_problema_gurobi():

    with open(f'{os.path.dirname(nome)}/Modelo-TFM.lp', 'w') as f:
f.write('Maximize\n Obj: L \n\nSubject to \n')

```

```

n_Restricoes = 0

docentes = pd.read_excel(nome,
sheet_name='Docentes',usecols=['Nome','Categoria','percentagem','Sa
ldo'])

lista_docentes = pd.read_excel(nome, sheet_name='Docentes',
usecols=['Nome']).values.flatten().tolist()
n_docentes = len(docentes)

exames = pd.read_excel(nome, sheet_name='Exames', header=None,
usecols=[0, 1, 3, 4])
lista_UC = pd.read_excel(nome, sheet_name='Exames',
header=None, usecols=[3]).T.reset_index(drop=True).iloc[0].tolist()
n_UC = exames.shape[0]

Variaveis_Decisao = pd.DataFrame(index=[None]*n_docentes,
columns=[None]*n_UC)
Variaveis_Decisao.columns = lista_UC[:n_UC]
Variaveis_Decisao.index = lista_docentes

for i in range(0,n_docentes):
    for j in range(0, n_UC):
        Variaveis_Decisao.iloc[i, j] = f'X({i+1},{j+1})'

def get_value(row):
    if not pd.isna(row[4]):
        return row[4]
    else:
        return row[1]

exames['Hora Real'] = exames.apply(get_value, axis=1)

Pij = pd.read_excel(nome, sheet_name='VProprias',skiprows=5,
header=None).iloc[:, 2:]

Soma_VP_Docente = []

for index, row in Pij.iterrows():
    Soma_VP_Docente.append(row.sum())

```

```
docentes['Soma VP'] = Soma_VP_Docente

n_Vigilancias_Proprias = pd.read_excel(nome,
sheet_name='VigilanciasPedidas',header=None).iloc[6, 1:].sum()
n_Vigilancias_Pedidas= pd.read_excel(nome,
sheet_name='VigilanciasPedidas',header=None).iloc[5, 1:].sum()
n_Total_Vigilancias = n_Vigilancias_Proprias +
n_Vigilancias_Pedidas

Dij = pd.read_excel(nome,
sheet_name='Disponibilidades',skiprows=5, header=None).iloc[:, 1:]

docentes['Li']=''
for index, row in docentes.iterrows():
    if row['Categoria'] != 'Cat' and row['percentagem'] == 1:
        docentes.loc[index, 'Li'] = 'Sim'

Soma_VP_Cat = 0
n_docentes_cat = 0

for index, row in docentes.iterrows():
    if row['Categoria'] == 'Cat':
        Soma_VP_Cat += row['Soma VP']
        n_docentes_cat = n_docentes_cat + 1

Soma_Todas_Percentagens = 0

for index, row in docentes.iterrows():
    Soma_Todas_Percentagens += row['percentagem']

M = 0
M= (n_Total_Vigilancias - Soma_VP_Cat)/(Soma_Todas_Percentagens
- n_docentes_cat)

Mi = []
for index, row in docentes.iterrows():
    Mi.append(math.floor( M * row['percentagem'] -
row['Saldo']))
```

```

docentes['Mi'] = Mi

Vj = []
Vj = pd.read_excel(nome,
sheet_name='VigilanciasPedidas',header=None).iloc[7, 1:]
Vj=Vj.reset_index(drop=True)
exames['Vj'] = Vj

#Escrever Varios Tipos de Restrições no Modelo-TFM

#Restrições Mi
for index, row in docentes.iterrows():
    if row['Soma VP'] <= row['Mi'] and row['Saldo'] < 0 :
        n_Restricoes += 1
        primeiro_membro= ''
        primeiro_membro = " +
".join(Variaveis_Decisao.iloc[index])
        with open(f'{os.path.dirname(nome)}/Modelo-TFM.lp',
'a') as f: f.write(f' c{n_Restricoes}: {primeiro_membro} =
{row["Mi"]}\n\n')
        elif row['Soma VP'] <= row['Mi'] and row['Saldo'] >= 0 :
            n_Restricoes += 1
            primeiro_membro= ''
            primeiro_membro = " +
".join(Variaveis_Decisao.iloc[index])
            with open(f'{os.path.dirname(nome)}/Modelo-TFM.lp',
'a') as f: f.write(f' c{n_Restricoes}: {primeiro_membro} <=
{row["Mi"]}\n\n')
            elif row['Soma VP'] > row['Mi'] and row['Saldo'] < 0 :
                n_Restricoes += 1
                primeiro_membro= ''
                primeiro_membro = " +
".join(Variaveis_Decisao.iloc[index])
                with open(f'{os.path.dirname(nome)}/Modelo-TFM.lp',
'a') as f: f.write(f' c{n_Restricoes}: {primeiro_membro} =
{row["Soma VP"]}\n\n')
                elif row['Soma VP'] > row['Mi'] and row['Saldo'] >= 0 :
                    n_Restricoes += 1

```



```

        primeiro_membro= ''
        primeiro_membro = " +
".join(Variaveis_Decisao.iloc[index])
        with open(f'{os.path.dirname(nome)}/Modelo-TFM.lp',
'a') as f: f.write(f' c{n_Restricoes}: {primeiro_membro} <=
{row["Soma VP"]}\n\n')

#Restrições L
for index, row in docentes.iterrows():
    if row['Li'] == 'Sim' and row['Saldo'] == 0:
        n_Restricoes += 1
        primeiro_membro= ''
        primeiro_membro = " +
".join(Variaveis_Decisao.iloc[index])
        with open(f'{os.path.dirname(nome)}/Modelo-TFM.lp',
'a') as f: f.write(f' c{n_Restricoes}: {primeiro_membro} - L >=
0\n\n')
        elif row['Li'] == 'Sim' and row['Saldo'] > 0:
            n_Restricoes += 1
            primeiro_membro= ''
            primeiro_membro = " +
".join(Variaveis_Decisao.iloc[index])
            with open(f'{os.path.dirname(nome)}/Modelo-TFM.lp',
'a') as f: f.write(f' c{n_Restricoes}: {primeiro_membro} - L >= -
{row["Saldo"]}\n\n')

#Restrições Vj
for i in range(0, n_UC):
    n_Restricoes += 1
    primeiro_membro= ''
    primeiro_membro = " + ".join(Variaveis_Decisao.iloc[:, i])
    with open(f'{os.path.dirname(nome)}/Modelo-TFM.lp', 'a') as
f: f.write(f' c{n_Restricoes}: {primeiro_membro} =
{exames.iloc[i]["Vj"]}\n\n')

#Restrições Cjk
Cjk= [[0 for j in range(n_UC)] for i in range(n_UC)]

```

```

    for j, row in exames.iterrows(): #For que começa no inicio e
vai pro fim
        for k, row_k in exames.loc[::-1].iterrows(): #For que
começa no fim e vai pro inicio
            if exames.loc[j, 0] == exames.loc[k, 0] and j > k:
                if exames.loc[j, 'Hora Real'] == exames.loc[k,
'Hora Real'] :
                    Cjk[j][k]=1
                elif
datetime.datetime.combine(datetime.datetime.today(), exames.loc[j,
'Hora Real']) + datetime.timedelta(hours=2) >
datetime.datetime.combine(datetime.datetime.today(), exames.loc[k,
'Hora Real']) and exames.loc[j, 'Hora Real'] < exames.loc[k, 'Hora
Real'] :
                    Cjk[j][k]=1
                elif
datetime.datetime.combine(datetime.datetime.today(), exames.loc[k,
'Hora Real']) + datetime.timedelta(hours=2) >
datetime.datetime.combine(datetime.datetime.today(), exames.loc[j,
'Hora Real']) and exames.loc[j, 'Hora Real'] > exames.loc[k, 'Hora
Real'] :
                    Cjk[j][k]=1

    for i in range(0,n_docentes):
        ultimo=0
        for j, row in exames.iterrows(): #For que começa no inicio
e vai pro fim
            for k, row_k in exames.loc[::-1].iterrows(): #For que
começa no fim e vai pro inicio
                if Cjk[k][j]==1 and j>=ultimo:
                    n_Restricoes += 1
                    primeiro_membro= ''
                    primeiro_membro = " +
".join(Variaveis_Decisao.iloc[i, j:k+1])
                    with open(f'{os.path.dirname(nome)}/Modelo-
TFM.lp', 'a') as f: f.write(f' c{n_Restricoes}: {primeiro_membro}
<= 1\n\n')

                    ultimo = k
                    break

```

```
#Restrições Dij
for i in range(0,n_docentes):
    for j in range(0,n_UC):
        if Dij.iloc[i,j] == 1:
            n_Restricoes += 1
            with open(f'{os.path.dirname(nome)}/Modelo-TFM.lp',
'a') as f: f.write(f' c{n_Restricoes}:
{Variaveis_Decisao.iloc[i,j]} = 0\n\n')

#Restrições Pij
for i in range(0,n_docentes):
    for j in range(0,n_UC):
        if Pij.iloc[i,j] == 1:
            n_Restricoes += 1
            with open(f'{os.path.dirname(nome)}/Modelo-TFM.lp',
'a') as f: f.write(f' c{n_Restricoes}:
{Variaveis_Decisao.iloc[i,j]} = 1\n\n')

#Terminar ficheiro
cumulativo = ''
for i in range(0, n_docentes):
    row_names = map(str, Variaveis_Decisao.iloc[i, :])
    cumulativo += ' '.join(row_names) + ' '

with open(f'{os.path.dirname(nome)}/Modelo-TFM.lp', 'a') as f:
f.write(f'\nBounds\n\nBinary\n {cumulativo} \n\nEnd')

#Resolver Problema
modelo = gp.read(f'{os.path.dirname(nome)}/Modelo-TFM.lp")

modelo.optimize()

#Criar ficheiro texto para guardar resultados
with open(f'{os.path.dirname(nome)}/Resultado-Modelo-TFM.txt',
'w') as f: f.write(f'Valor das Vigilancias Totais por
Docente:\n\n')

#Da Valor total das Vigilancias por Docente
```

```
for i in range(0, n_docentes):
    soma = 0
    for variable in modelo.getVars():
        if variable.varName in
Variaveis_Decisao.iloc[i,:].tolist():
        soma = soma + variable.X
    with open(f'{os.path.dirname(nome)}/Resultado-Modelo-
TFM.txt', 'a') as f: f.write(f'{lista_docentes[i]} = {soma}\n')

global valor_objetivo_gurobi
valor_objetivo_gurobi = modelo.objVal

#Criar DataFrame com valor das Variaveis de Decisão
nivel0=pd.read_excel(nome, sheet_name='Exames', header=None,
usecols=[0]).T.reset_index(drop=True).iloc[0].tolist()
data_conversao = [d.strftime('%Y-%m-%d') for d in nivel0]
nivel1=pd.read_excel(nome, sheet_name='Exames', header=None,
usecols=[2]).T.reset_index(drop=True).iloc[0].tolist()
nivel2= pd.read_excel(nome, sheet_name='Exames', header=None,
usecols=[1]).T.reset_index(drop=True).iloc[0].tolist()
nivel3 = pd.read_excel(nome, sheet_name='Exames', header=None,
usecols=[3]).T.reset_index(drop=True).iloc[0].tolist()

index_multinivel = [data_conversao, nivel1, nivel2, nivel3]

global Mapa_Otimo
Mapa_Otimo = pd.DataFrame(index=[None]*n_docentes,
columns=[None]*n_UC)
Mapa_Otimo.columns = index_multinivel[:n_UC]
Mapa_Otimo.index = lista_docentes

for i in range(0,n_docentes):
    for j in range(0, n_UC):
        Mapa_Otimo.iloc[i, j] = f'X({i+1},{j+1})'

variaveis_valores = {var.VarName: var.X for var in
modelo.getVars()}

Mapa_Otimo = Mapa_Otimo.replace(variaveis_valores)
```

```
#####
```

```
def otimizacao_problema_OR_Tools():

    #Criar Objeto Solver
    modelo = pywraplp.Solver.CreateSolver('CBC')

    docentes = pd.read_excel(nome,
sheet_name='Docentes',usecols=['Nome','Categoria','percentagem','Sa
lido'])
    lista_docentes = pd.read_excel(nome, sheet_name='Docentes',
usecols=['Nome']).values.flatten().tolist()
    n_docentes = len(docentes)

    exames = pd.read_excel(nome, sheet_name='Exames', header=None,
usecols=[0, 1, 3, 4])
    lista_UC = pd.read_excel(nome, sheet_name='Exames',
header=None, usecols=[3]).T.reset_index(drop=True).iloc[0].tolist()
    n_UC = exames.shape[0]

    Variaveis_Decisao = pd.DataFrame(index=[None]*n_docentes,
columns=[None]*n_UC)
    Variaveis_Decisao.columns = lista_UC[:n_UC]
    Variaveis_Decisao.index = lista_docentes

    for i in range(0,n_docentes):
        for j in range(0, n_UC):
            Variaveis_Decisao.iloc[i, j] = f'X({i+1},{j+1})'

    #Criar Variaveis de decisão do modelo
    for i in range(0,n_docentes):
        for j in range(0, n_UC):
            modelo.IntVar(0, 1, f'X({i+1},{j+1})')

    #Criar L
    L = modelo.IntVar(0, modelo.infinity(), 'L')
```

```

def get_value(row):
    if not pd.isna(row[4]):
        return row[4]
    else:
        return row[1]

exames['Hora Real'] = exames.apply(get_value, axis=1)

Pij = pd.read_excel(nome, sheet_name='VProprias', skiprows=5,
header=None).iloc[:, 2:]

Soma_VP_Docente = []

for index, row in Pij.iterrows():
    Soma_VP_Docente.append(row.sum())

docentes['Soma VP'] = Soma_VP_Docente

n_Vigilancias_Proprias = pd.read_excel(nome,
sheet_name='VigilanciasPedidas', header=None).iloc[6, 1:].sum()
n_Vigilancias_Pedidas= pd.read_excel(nome,
sheet_name='VigilanciasPedidas', header=None).iloc[5, 1:].sum()
n_Total_Vigilancias = n_Vigilancias_Proprias +
n_Vigilancias_Pedidas

Dij = pd.read_excel(nome,
sheet_name='Disponibilidades', skiprows=5, header=None).iloc[:, 1:]

docentes['Li']=''
for index, row in docentes.iterrows():
    if row['Categoria'] != 'Cat' and row['percentagem'] == 1:
        docentes.loc[index, 'Li'] = 'Sim'

Soma_VP_Cat = 0
n_docentes_cat = 0

for index, row in docentes.iterrows():
    if row['Categoria'] == 'Cat':
        Soma_VP_Cat += row['Soma VP']

```

```

n_docentes_cat = n_docentes_cat + 1

Soma_Todas_Percentagens = 0

for index, row in docentes.iterrows():
    Soma_Todas_Percentagens += row['percentagem']

M = 0
M= (n_Total_Vigilancias - Soma_VP_Cat)/(Soma_Todas_Percentagens
- n_docentes_cat)

Mi = []
for index, row in docentes.iterrows():
    Mi.append(math.floor( M * row['percentagem'] -
row['Saldo']))

docentes['Mi'] = Mi

Vj = []
Vj = pd.read_excel(nome,
sheet_name='VigilanciasPedidas',header=None).iloc[7, 1:]
Vj=Vj.reset_index(drop=True)
exames['Vj'] = Vj

#Escrever Varios Tipos de Restrições no Modelo-TFM

#Restrições Mi
for index, row in docentes.iterrows():
    primeiro_membro = " + ".join(Variaveis_Decisao.iloc[index])
    soma_variaveis = [modelo.LookupVariable(v) for v in
Variaveis_Decisao.iloc[index]]

    if row['Soma VP'] <= row['Mi'] and row['Saldo'] < 0:
        modelo.Add(modelo.Sum(soma_variaveis) == row["Mi"])
    elif row['Soma VP'] <= row['Mi'] and row['Saldo'] >= 0:
        modelo.Add(modelo.Sum(soma_variaveis) <= row["Mi"])
    elif row['Soma VP'] > row['Mi'] and row['Saldo'] < 0:

```

```

        modelo.Add(modelo.Sum(soma_variaveis) == row["Soma
VP"])
        elif row['Soma VP'] > row['Mi'] and row['Saldo'] >= 0:
            modelo.Add(modelo.Sum(soma_variaveis) <= row["Soma
VP"])

#Restrições L
for index, row in docentes.iterrows():
    primeiro_membro = " + ".join(Variaveis_Decisao.iloc[index])
    variaveis = [modelo.LookupVariable(v) for v in
Variaveis_Decisao.iloc[index]]
    if row['Li'] == 'Sim' and row['Saldo'] == 0:
        modelo.Add(modelo.Sum(variaveis) -
modelo.LookupVariable('L') >= 0)
    elif row['Li'] == 'Sim' and row['Saldo'] > 0:
        modelo.Add(modelo.Sum(variaveis) -
modelo.LookupVariable('L') + row["Saldo"] >= 0)

#Restrições Vj
for i in range(0, n_UC):
    primeiro_membro = " + ".join(Variaveis_Decisao.iloc[:, i])
    variaveis = [modelo.LookupVariable(v) for v in
Variaveis_Decisao.iloc[:, i]]
    modelo.Add(modelo.Sum(variaveis) == exames.iloc[i]["Vj"])

#Restrições Cjk
Cjk= [[0 for j in range(n_UC)] for i in range(n_UC)]

for j, row in exames.iterrows(): #For que começa no inicio e
vai pro fim
    for k, row_k in exames.loc[:, -1].iterrows(): #For que
começa no fim e vai pro inicio
        if exames.loc[j, 0] == exames.loc[k, 0] and j > k:
            if exames.loc[j, 'Hora Real'] == exames.loc[k,
'Hora Real'] :
                Cjk[j][k]=1
            elif
datetime.datetime.combine(datetime.datetime.today(), exames.loc[j,
'Hora Real']) + datetime.timedelta(hours=2) >

```



```

datetime.datetime.combine(datetime.datetime.today(), exames.loc[k,
'Hora Real']) and exames.loc[j, 'Hora Real'] < exames.loc[k, 'Hora
Real'] :

        Cjk[j][k]=1
        elif
datetime.datetime.combine(datetime.datetime.today(), exames.loc[k,
'Hora Real']) + datetime.timedelta(hours=2) >
datetime.datetime.combine(datetime.datetime.today(), exames.loc[j,
'Hora Real']) and exames.loc[j, 'Hora Real'] > exames.loc[k, 'Hora
Real'] :

        Cjk[j][k]=1

for i in range(n_docentes):
    ultimo = 0
    for j, row in exames.iterrows():
        for k, row_k in exames.loc[:, -1].iterrows():
            if Cjk[k][j] == 1 and j >= ultimo:
                primeiro_membro = " +
".join(Variaveis_Decisao.iloc[i, j:k+1])
                variaveis = [modelo.LookupVariable(v) for v in
Variaveis_Decisao.iloc[i, j:k+1]]
                modelo.Add(modelo.Sum(variaveis) <= 1)
                ultimo = k
                break

#Restrições Dij
for i in range(0, n_docentes):
    for j in range(0, n_UC):
        if Dij.iloc[i,j] == 1:

modelo.Add(modelo.LookupVariable(Variaveis_Decisao.iloc[i,j]) == 0)

#Restrições Pij
for i in range(0, n_docentes):
    for j in range(0, n_UC):
        if Pij.iloc[i,j] == 1:

modelo.Add(modelo.LookupVariable(Variaveis_Decisao.iloc[i,j]) == 1)

```

```
# Criar Função Objetivo
modelo.Maximize(L)

#Resolver Problema
modelo.Solve()

# #Criar ficheiro texto para guardar resultados
with open(f'{os.path.dirname(nome)}/Resultado-Modelo-TFM.txt',
'w') as f: f.write(f'Valor das Vigilancias Totais por
Docente:\n\n')

# Da Valor total das Vigilancias por Docente
soma = 0
contador = 0
i=-1
for variable in modelo.variables():
    contador = contador +1
    soma = soma + variable.solution_value()
    if contador == n_UC:
        i = i + 1
        with open(f'{os.path.dirname(nome)}/Resultado-
Modelo-TFM.txt', 'a') as f:
            f.write(f'{lista_docentes[i]} = {soma}\n')
        contador = 0
        soma = 0

global valor_objetivo_OR_Tools
valor_objetivo_OR_Tools = modelo.Objective().Value()

#Criar DataFrame com valor das Variaveis de Decisão
nivel0=pd.read_excel(nome, sheet_name='Exames', header=None,
usecols=[0]).T.reset_index(drop=True).iloc[0].tolist()
data_conversao = [d.strftime('%Y-%m-%d') for d in nivel0]
nivel1=pd.read_excel(nome, sheet_name='Exames', header=None,
usecols=[2]).T.reset_index(drop=True).iloc[0].tolist()
nivel2= pd.read_excel(nome, sheet_name='Exames', header=None,
usecols=[1]).T.reset_index(drop=True).iloc[0].tolist()
nivel3 = pd.read_excel(nome, sheet_name='Exames', header=None,
usecols=[3]).T.reset_index(drop=True).iloc[0].tolist()
```

```

index_multinivel = [data_conversao, nivel1, nivel2, nivel3]

global Mapa_Otimo
Mapa_Otimo = pd.DataFrame(index=[None]*n_docentes,
columns=[None]*n_UC)
Mapa_Otimo.columns = index_multinivel[:n_UC]
Mapa_Otimo.index = lista_docentes

j=0
i=0
for variable in modelo.variables():
    Mapa_Otimo.iloc[i, j] = variable.solution_value()
    j = j + 1
    if j == n_UC:
        i = i + 1
        j = 0
        if i == n_docentes:
            break

#####

def abrir_docentes():
    local_docentes = f'{os.path.dirname(nome)}/Resultado-Modelo-
TFM.txt'
    os.startfile(local_docentes)

def atualizar_canvas():
    painel_janela_2.delete("all")
    if nome != '':
        caixa = painel_janela_2.create_rectangle(10, 10, 490, 40,
fill="white")
        texto = painel_janela_2.create_text(250, 25, text= nome )
    else:
        caixa = painel_janela_2.create_rectangle(10, 10, 490, 40,
fill="white") #Cria caixa branca no painel
        texto = painel_janela_2.create_text(50, 25, text= 'Nada a
inserir' )

```

```
def escolher_localizacao_dados():
    global nome
    nome = filedialog.askopenfilename()
    atualizar_canvas()

def exportar_mapa():
    local_mapa = filedialog.asksaveasfilename(initialfile="Mapa
Ótimo", defaultextension=".xlsx", filetypes=[("Livro do Excel",
"*.*"), ("All Files", "*.*")])
    Mapa_Otimo.to_excel(local_mapa)

# Função para Criar Janela 2, fechando a anterior
def criar_janela_2():
    fechar_janela_1()
    global janela_2
    janela_2 = tk.Tk()
    janela_2.title("Escalonamento de Docentes a Vigilâncias")
    janela_2.state('zoomed')
    image_o = Image.open(resource_path("login.jpg"))
    image = ImageTk.PhotoImage(image_o)
    image_label = tk.Label(janela_2, image=image)
    image_label.image = image
    image_label.place(x=0, y=0, relwidth=1, relheight=1)
    r = 238
    g = 238
    b = 238
    color = f"#{r:02x}{g:02x}{b:02x}"
    frame_janela2 = tk.Frame(janela_2, bg=color)
    frame_janela2.place(relx=0.5, rely=0.5, anchor=tk.CENTER,
width=600, height=200)
    global botao_alterar_diretorio
    botao_alterar_diretorio = tk.Button(frame_janela2,
text="Alterar", command=escolher_localizacao_dados)
    botao_alterar_diretorio.pack()
    global botao_seguinte_janela_2
    botao_seguinte_janela_2 = tk.Button(frame_janela2,
text="Seguinte", command=criar_janela_3)
    botao_seguinte_janela_2.pack()
```

```

        botoao_sair = tk.Button(frame_janela2, text="Sair do
Interface",command=janela_2.destroy)
        botoao_sair.pack()
        label_janela_2 = tk.Label(frame_janela2, text="Indique a
localização do ficheiro com os dados do respetivo problema:")
        label_janela_2.pack()
        global painel_janela_2
        painel_janela_2 = tk.Canvas(frame_janela2, width=490,
height=40) #Cria painel onde é possível desenhar
        painel_janela_2.pack()
        caixa = painel_janela_2.create_rectangle(10, 10, 490, 40,
fill="white") #Cria caixa branca no painel
        texto = painel_janela_2.create_text(50, 25, text= 'Nada a
inserir' )
        label_janela_2.place(x=10, y=5)
        painel_janela_2.place(x=3, y=30)
        botoao_alterar_diretorio.place(x=510, y=35, width=75, height=40)
        botoao_sair.place(x=409, y=125, width=155, height=50)
        botoao_seguinte_janela_2.place(x=33, y=125, width=155,
height=50)
        image_o3 = Image.open(resource_path("ISEG_Logo.png"))
        image3 = ImageTk.PhotoImage(image_o3)
        image_label2 = tk.Label(janela_2, image=image3, borderwidth=0)
        image_label2.image = image3
        image_label2.pack()

# Função para Criar Janela 5, fechando a anterior
def criar_janela_5_gurobi():
    espera()
    thread = threading.Thread(target=otimizacao_problema_gurobi)
    thread.start()
    while thread.is_alive(): # Verificar se a thread ainda está a
correr
        janela_espera.update()
        time.sleep(0.01) # Espera 100 milésimos de segundo para
atualizar frame
    fechar_janela_3()
    janela_5 = tk.Tk()
    janela_5.title("Escalonamento de Docentes a Vigilâncias")

```

```
janela_5.state('zoomed')
image_o = Image.open(resource_path("login.jpg"))
image = ImageTk.PhotoImage(image_o)
image_label = tk.Label(janela_5, image=image)
image_label.image = image
image_label.place(x=0, y=0, relwidth=1, relheight=1)
r = 238
g = 238
b = 238
color = f"#{r:02x}{g:02x}{b:02x}"
frame_janela5 = tk.Frame(janela_5, bg=color)
frame_janela5.place(relx=0.5, rely=0.5, anchor=tk.CENTER,
width=600, height=200)
label_janela_5 = tk.Label(frame_janela5, text="Valor Ótimo de
L:")
label_janela_5.pack()
global botao_ver_vigilancias
botao_ver_vigilancias = tk.Button(frame_janela5, text="Ver o
Número de Vigilâncias por Docente", command=abrir_docentes)
botao_ver_vigilancias.pack()
global botao_exportar
botao_exportar = tk.Button(frame_janela5, text="Exportar Mapa
de Escalonamento Ótimo", command=exportar_mapa)
botao_exportar.pack()
global botao_retroceder_janela_5
botao_retroceder_janela_5 = tk.Button(frame_janela5,
text="Retroceder", command=comando_botao_retroceder_janela_5)
botao_retroceder_janela_5.pack()
botao_fechar = tk.Button(frame_janela5, text="Fechar
Programa", command=janela_5.destroy)
botao_fechar.pack()
painell = tk.Canvas(frame_janela5, width=50, height=40) #Cria
painel onde é possível desenhar
painell.pack()
caixa1 = painell.create_rectangle(10, 10, 30, 30, fill="white")
#Cria caixa branca no painel
numero1 = painell.create_text(20, 20, text=
valor_objetivo_gurobi) #Insere número dentro da caixa
label_janela_5.place(x=10, y=10)
```

```

painell.place(x=105, y=1)
botao_ver_vigilancias.place(x=10, y=50, width=250, height=40)
botao_exportar.place(x=10, y=100, width=250, height=40)
botao_retroceder_janela_5.place(x=350, y=100, width=175,
height=40)
botao_fechar.place(x=350, y=50, width=175, height=40)
image_o3 = Image.open(resource_path("ISEG_Logo.png"))
image3 = ImageTk.PhotoImage(image_o3)
image_label2 = tk.Label(janela_5, image=image3, borderwidth=0)
image_label2.image = image3
image_label2.pack()

def criar_janela_5_OR_Tools():
    espera()
    thread = threading.Thread(target=otimizacao_problema_OR_Tools)
    thread.start()
    while thread.is_alive(): # Verificar se a thread ainda está a
correr
        janela_espera.update()
        time.sleep(0.001) # Espera 10 milésimos de segundo para
atualizar frame
    fechar_janela_3()
    janela_5 = tk.Tk()
    janela_5.title("Escalonamento de Docentes a Vigilâncias")
    janela_5.state('zoomed')
    image_o = Image.open(resource_path("login.jpg"))
    image = ImageTk.PhotoImage(image_o)
    image_label = tk.Label(janela_5, image=image)
    image_label.image = image
    image_label.place(x=0, y=0, relwidth=1, relheight=1)
    r = 238
    g = 238
    b = 238
    color = f"#{r:02x}{g:02x}{b:02x}"
    frame_janela5 = tk.Frame(janela_5, bg=color)
    frame_janela5.place(relx=0.5, rely=0.5, anchor=tk.CENTER,
width=600, height=200)
    label_janela_5 = tk.Label(frame_janela5, text="Valor Ótimo de
L:")

```

```
label_janela_5.pack()
global botao_ver_vigilancias
botao_ver_vigilancias = tk.Button(frame_janela5, text="Ver o
Número de Vigilâncias por Docente",command=abrir_docentes)
botao_ver_vigilancias.pack()
global botao_exportar
botao_exportar = tk.Button(frame_janela5, text="Exportar Mapa
de Escalonamento Ótimo",command=exportar_mapa)
botao_exportar.pack()
global botao_retroceder_janela_5
botao_retroceder_janela_5 = tk.Button(frame_janela5,
text="Retroceder",command=comando_botao_retroceder_janela_5)
botao_retroceder_janela_5.pack()
botao_fechar = tk.Button(frame_janela5, text="Fechar
Programa",command=janela_5.destroy)
botao_fechar.pack()
painell = tk.Canvas(frame_janela5, width=50, height=40) #Cria
painel onde é possível desenhar
painell.pack()
caixa1 = painell.create_rectangle(10, 10, 30, 30, fill="white")
#Cria caixa branca no painel
numero1 = painell.create_text(20, 20, text=
valor_objetivo_OR_Tools) #Insere número dentro da caixa
label_janela_5.place(x=10, y=10)
painell.place(x=105, y=1)
botao_ver_vigilancias.place(x=10, y=50, width=250, height=40)
botao_exportar.place(x=10, y=100, width=250, height=40)
botao_retroceder_janela_5.place(x=350, y=100, width=175,
height=40)
botao_fechar.place(x=350, y=50, width=175, height=40)
image_o3 = Image.open(resource_path("ISEG_Logo.png"))
image3 = ImageTk.PhotoImage(image_o3)
image_label2 = tk.Label(janela_5, image=image3, borderwidth=0)
image_label2.image = image3
image_label2.pack()

# Função para Criar Janela 3, fechando a anterior
def criar_janela_3():
```



```
if 'nome' in locals() or 'nome' in globals():
    file_name, file_extension = os.path.splitext(nome)
    if file_extension != ".xlsx":
        janela_avisos3 = tk.Toplevel(janela_2)
        janela_avisos3.title("Escalonamento de Docentes a
Vigilâncias")
        janela_avisos3.geometry("600x200")
        janela_avisos3.geometry("+460+295")
        label_janela_avisos3 = tk.Label(janela_avisos3,
text="Erro! Não inseriu um ficheiro do tipo .xlsx!", font=("", 16))
        label_janela_avisos3.pack()
        botao_ok = tk.Button(janela_avisos3, text="Ok",
command=janela_avisos3.destroy)
        botao_ok.pack()
        botao_ok.place(x=409, y=125, width=155, height=50)
        image_o2 = Image.open(resource_path("error.png"))
        image2 = ImageTk.PhotoImage(image_o2)
        image_label2 = tk.Label(janela_avisos3, image=image2)
        image_label2.image = image2
        image_label2.pack()
        image_label2.place(x=125, y=65)
        som_local = resource_path("Paragem Crítica.wav")
        som = threading.Thread(target=tocar_som,
args=(som_local,))
        som.start()
    else:
        ficheiro_excel = openpyxl.load_workbook(nome)
        sheets_a_encontrar = ['Docentes', 'VProprias', 'Exames',
'VigilanciasPedidas', 'Disponibilidades']
        folha_encontrada = False
        for sheet_name in ficheiro_excel.sheetnames:
            if sheet_name in sheets_a_encontrar:
                folha_encontrada= True
                break
        if not folha_encontrada:
            janela_avisos = tk.Toplevel(janela_2)
            janela_avisos.title("Escalonamento de Docentes a
Vigilâncias")
            janela_avisos.geometry("600x200")
```

```
        janela_avisos.geometry("+460+295")
        label_janela_avisos = tk.Label(janela_avisos,
text="Não foi encontrada alguma das seguintes folhas no ficheiro
selecionado:\n {}".format(sheets_a_encontrar),font=("", 13))
        label_janela_avisos.pack()
        botao_alt_ficheiro = tk.Button(janela_avisos,
text="Alterar Ficheiro", command=janela_avisos.destroy)
        botao_alt_ficheiro.pack()
        botao_alt_ficheiro.place(x=409, y=125,
width=155, height=50)
        image_o2 =
Image.open(resource_path("error.png"))
        image2 = ImageTk.PhotoImage(image_o2)
        image_label2 = tk.Label(janela_avisos,
image=image2)
        image_label2.image = image2
        image_label2.pack()
        image_label2.place(x=125, y=65)
        som_local = resource_path("Paragem
Crítica.wav")
        som = threading.Thread(target=tocar_som,
args=(som_local,))
        som.start()
    else:
        #Definir nº docentes e lista
        docentes = pd.read_excel(nome,
sheet_name='Docentes', usecols=['Nome', 'Categoria', 'percentagem', 'Sa
ldo'])
        lista_docentes = pd.read_excel(nome,
sheet_name='Docentes', usecols=['Nome']).values.flatten().tolist()
        n_docentes = len(docentes)
        #Definir nº UC e lista
        exames = pd.read_excel(nome,
sheet_name='Exames', header=None, usecols=[0, 1, 3, 4])
        lista_UC = pd.read_excel(nome,
sheet_name='Exames', header=None,
usecols=[3]).T.reset_index(drop=True).iloc[0].tolist()
        n_UC = exames.shape[0]
```

```
#Criar Index Multinivel
nivel0=pd.read_excel(nome, sheet_name='Exames',
header=None, usecols=[0]).T.reset_index(drop=True).iloc[0].tolist()
data_conversao = [d.strftime('%Y-%m-%d') for d
in nivel0]

nivel1=pd.read_excel(nome, sheet_name='Exames',
header=None, usecols=[2]).T.reset_index(drop=True).iloc[0].tolist()
nivel2= pd.read_excel(nome,
sheet_name='Exames', header=None,
usecols=[1]).T.reset_index(drop=True).iloc[0].tolist()
nivel3 = pd.read_excel(nome,
sheet_name='Exames', header=None,
usecols=[3]).T.reset_index(drop=True).iloc[0].tolist()
index_multinivel = [data_conversao, nivel1,
nivel2, nivel3]

#Criar DataFrame com Vigilancias Proprias
Matriz_VP = pd.read_excel(nome,
sheet_name='VProprias',skiprows=4).iloc[:, 2:]
Matriz_VP.columns = index_multinivel[:n_UC]
Matriz_VP.index = lista_docentes

#Criar DataFrame com Indisponibilidades
Matriz_Ind = pd.read_excel(nome,
sheet_name='Disponibilidades',skiprows=4).iloc[:, 1:]
Matriz_Ind.columns = index_multinivel[:n_UC]
Matriz_Ind.index = lista_docentes

#Cria Vetor Booleano, True=Existe
Incompatibilidade, False=Não Existe
comparador = (Matriz_VP.values == 1) &
(Matriz_Ind.values == 1)
linha_idx, coluna_idx = np.where(comparador)

if comparador.any(): #Se alguma delas for True,
dá mensagem de erro

janela_aviso = tk.Toplevel(janela_2)
```

```
janela_avisotitle("Escalonamento de
Docentes a Vigilâncias")
janela_avisogeometry("700x200")
janela_avisogeometry("+460+295")
label_janela_avisotext=f"Foi encontrada, pelo menos, uma Incompatibilidade nos
Dados!\n {Matriz_VP.index[linha_idx[0]]} tem uma Vigilância Propria
e Indisponibilidade marcada na mesma Vigilância.\n Nome =
{Matriz_VP.columns.get_level_values(3)[coluna_idx[0]]}\n Dia =
{Matriz_VP.columns.get_level_values(0)[coluna_idx[0]]}\n Horas =
{Matriz_VP.columns.get_level_values(2)[coluna_idx[0]]}",font=("",
13))
label_janela_avisopack()
botao_alt_ficheiro =
tk.Button(janela_avisotext="Ok", command=janela_avisodestroy)
botao_alt_ficheiro.pack()
botao_alt_ficheiro.place(x=409, y=125,
width=155, height=50)
image_o2 =
Image.open(resource_path("error.png"))
image2 = ImageTk.PhotoImage(image_o2)
image_label2 = tk.Label(janela_avisotext="Foi encontrada, pelo menos, uma Incompatibilidade nos
Dados!\n {Matriz_VP.index[linha_idx[0]]} tem uma Vigilância Propria
e Indisponibilidade marcada na mesma Vigilância.\n Nome =
{Matriz_VP.columns.get_level_values(3)[coluna_idx[0]]}\n Dia =
{Matriz_VP.columns.get_level_values(0)[coluna_idx[0]]}\n Horas =
{Matriz_VP.columns.get_level_values(2)[coluna_idx[0]]}",font=("",
13), image=image2)
image_label2.image = image2
image_label2.pack()
image_label2.place(x=125, y=65)
som_local = resource_path("Paragem
Crítica.wav")
som = threading.Thread(target=tocar_som,
args=(som_local,))
som.start()
else:
fechar_janela_2()
global janela_3
janela_3 = tk.Tk()
janela_3.title("Escalonamento de Docentes a
Vigilâncias")
janela_3.state('zoomed')
```

```
        image_o =
Image.open(resource_path("login.jpg"))
        image = ImageTk.PhotoImage(image_o)
        image_label = tk.Label(janela_3,
image=image)
        image_label.image = image
        image_label.place(x=0, y=0, relwidth=1,
relheight=1)
        r = 238
        g = 238
        b = 238
        color = f"#{r:02x}{g:02x}{b:02x}"
        frame_janela3 = tk.Frame(janela_3,
bg=color)
        frame_janela3.place(relx=0.5, rely=0.5,
anchor=tk.CENTER, width=600, height=200)
        label1_janela_3 = tk.Label(frame_janela3,
text="Por favor confira os dados do problema:")
        label1_janela_3.pack()
        label2_janela_3 = tk.Label(frame_janela3,
text="Número de Docentes:")
        label2_janela_3.pack()
        label3_janela_3 = tk.Label(frame_janela3,
text="Número de Exames:")
        label3_janela_3.pack()
        label4_janela_3 = tk.Label(frame_janela3,
text="Localização dos Dados:")
        label4_janela_3.pack()
        painell1 = tk.Canvas(frame_janela3,
width=100, height=50) #Cria painel onde é possível desenhar
        painell1.pack()
        caixa1 = painell1.create_rectangle(10, 10,
90, 40, fill="white") #Cria caixa branca no painel
        numerol1 = painell1.create_text(50, 25, text=
"Sem Dados") #Insere número dentro da caixa
        painel2 = tk.Canvas(frame_janela3,
width=100, height=50) #Cria painel onde é possível desenhar
        painel2.pack()
```

```
        caixa2 = painel2.create_rectangle(10, 10,
90, 40, fill="white") #Cria caixa branca no painel
        numero2 = painel2.create_text(50, 25,
text="Sem Dados") #Insere número dentro da caixa
        global botao_otimizar_gurobi
        botao_otimizar_gurobi =
tk.Button(frame_janela3, text="Otimizar com Gurobi\n(Requer
Licença)", command=criar_janela_5_gurobi)
        botao_otimizar_gurobi.pack()
        global botao_otimizar_OR_Tools
        botao_otimizar_OR_Tools =
tk.Button(frame_janela3, text="Otimizar com OR-Tools\n(Freeware)",
command=criar_janela_5_OR_Tools)
        botao_otimizar_OR_Tools.pack()
        global botao_retroceder_janela_3
        botao_retroceder_janela_3 =
tk.Button(frame_janela3, text="Retroceder",
command=comando_botao_retroceder_janela_3)
        botao_retroceder_janela_3.pack()
        painel_janela_3 = tk.Canvas(frame_janela3,
width=490, height=40) #Cria painel onde é possível desenhar
        painel_janela_3.pack()
        caixa =
painel_janela_3.create_rectangle(10, 10, 490, 40, fill="white")
#Cria caixa branca no painel
        texto = painel_janela_3.create_text(50, 25,
text= 'Nada a inserir' )
        label1_janela_3.place(x=10, y=2)
        label2_janela_3.place(x=10, y=35)
        label3_janela_3.place(x=230, y=35)
        label4_janela_3.place(x=10, y=70)
        painel1.place(x=127, y=20)
        painel2.place(x=354, y=20)
        botao_otimizar_gurobi.place(x=20, y=140,
width=130, height=50)
        botao_otimizar_OR_Tools.place(x=240, y=140,
width=150, height=50)
        botao_retroceder_janela_3.place(x=480,
y=140, width=100, height=50)
```

```

        painel_janela_3.place(x=3, y=85)
        image_o3 =
Image.open(resource_path("ISEG_Logo.png"))
        image3 = ImageTk.PhotoImage(image_o3)
        image_label2 = tk.Label(janela_3,
image=image3, borderwidth=0)
        image_label2.image = image3
        image_label2.pack()
        if nome != '':
            docentes = pd.read_excel(nome,
sheet_name='Docentes', usecols=['Nome', 'Categoria', 'percentagem', 'Sa
lido'])
            n_docentes = len(docentes)
            exames = pd.read_excel(nome,
sheet_name='Exames', header=None, usecols=[0, 1, 3, 4])
            n_UC = exames.shape[0]
            painel1.delete(numero1)
            painel2.delete(numero2)
            caixa =
painel_janela_3.create_rectangle(10, 10, 490, 40, fill="white")
            texto =
painel_janela_3.create_text(250, 25, text= nome )
            numero1 = painel1.create_text(50, 25,
text= n_docentes)
            numero2 = painel2.create_text(50, 25,
text=n_UC)
        else:
            janela_avisos = tk.Toplevel(janela_2)
            janela_avisos.title("Escalonamento de Docentes a
Vigilâncias")
            janela_avisos.geometry("600x200")
            janela_avisos.geometry("+460+295")
            label_janela_avisos = tk.Label(janela_avisos, text="Erro!
Não inseriu uma localização válida do ficheiro!", font=("", 16))
            label_janela_avisos.pack()
            botao_ok = tk.Button(janela_avisos, text="Ok",
command=janela_avisos.destroy)
            botao_ok.pack()
            botao_ok.place(x=409, y=125, width=155, height=50)

```

```
image_o2 = Image.open(resource_path("error.png"))
image2 = ImageTk.PhotoImage(image_o2)
image_label2 = tk.Label(janela_avisos2, image=image2)
image_label2.image = image2
image_label2.pack()
image_label2.place(x=125, y=65)
som_local = resource_path("Paragem Crítica.wav")
som = threading.Thread(target=tocar_som, args=(som_local,))
som.start()

def comando_botao_retroceder_janela_3():
    fechar_janela_3()
    global janela_2
    janela_2 = tk.Tk()
    janela_2.title("Escalonamento de Docentes a Vigilâncias")
    janela_2.state('zoomed')
    image_o = Image.open(resource_path("login.jpg"))
    image = ImageTk.PhotoImage(image_o)
    image_label = tk.Label(janela_2, image=image)
    image_label.image = image
    image_label.place(x=0, y=0, relwidth=1, relheight=1)
    r = 238
    g = 238
    b = 238
    color = f"#{r:02x}{g:02x}{b:02x}"
    frame_janela2 = tk.Frame(janela_2, bg=color)
    frame_janela2.place(relx=0.5, rely=0.5, anchor=tk.CENTER,
width=600, height=200)
    global botao_alterar_diretorio
    botao_alterar_diretorio = tk.Button(frame_janela2,
text="Alterar", command=escolher_localizacao_dados)
    botao_alterar_diretorio.pack()
    global botao_seguinte_janela_2
    botao_seguinte_janela_2 = tk.Button(frame_janela2,
text="Seguinte", command=criar_janela_3)
    botao_seguinte_janela_2.pack()
    botao_sair = tk.Button(frame_janela2, text="Sair do
Interface", command=janela_2.destroy)
    botao_sair.pack()
```



```
label_janela_2 = tk.Label(frame_janela2, text="Indique a
localização do ficheiro com os dados do respetivo problema:")
label_janela_2.pack()
global painel_janela_2
painel_janela_2 = tk.Canvas(frame_janela2, width=490,
height=40) #Cria painel onde é possível desenhar
painel_janela_2.pack()
caixa = painel_janela_2.create_rectangle(10, 10, 490, 40,
fill="white") #Cria caixa branca no painel
texto = painel_janela_2.create_text(50, 25, text= 'Nada a
inserir' )
label_janela_2.place(x=10, y=5)
painel_janela_2.place(x=3, y=30)
botao_alterar_diretorio.place(x=510, y=35, width=75, height=40)
botao_sair.place(x=409, y=125, width=155, height=50)
botao_seguinte_janela_2.place(x=33, y=125, width=155,
height=50)
image_o3 = Image.open(resource_path("ISEG_Logo.png"))
image3 = ImageTk.PhotoImage(image_o3)
image_label2 = tk.Label(janela_2, image=image3, borderwidth=0)
image_label2.image = image3
image_label2.pack()
if nome != '':
    caixa = painel_janela_2.create_rectangle(10, 10, 490, 40,
fill="white")
    texto = painel_janela_2.create_text(250, 25, text= nome )

def comando_botao_retroceder_janela_5():
    fechar_janela_5()
    global janela_2
    janela_2 = tk.Tk()
    janela_2.title("Escalonamento de Docentes a Vigilâncias")
    janela_2.state('zoomed')
    image_o = Image.open(resource_path("login.jpg"))
    image = ImageTk.PhotoImage(image_o)
    image_label = tk.Label(janela_2, image=image)
    image_label.image = image
    image_label.place(x=0, y=0, relwidth=1, relheight=1)
    r = 238
```

```
g = 238
b = 238
color = f"#{r:02x}{g:02x}{b:02x}"
frame_janela2 = tk.Frame(janela_2, bg=color)
frame_janela2.place(relx=0.5, rely=0.5, anchor=tk.CENTER,
width=600, height=200)
global botao_alterar_diretorio
botao_alterar_diretorio = tk.Button(frame_janela2,
text="Alterar", command=escolher_localizacao_dados)
botao_alterar_diretorio.pack()
global botao_seguinte_janela_2
botao_seguinte_janela_2 = tk.Button(frame_janela2,
text="Seguinte", command=criar_janela_3)
botao_seguinte_janela_2.pack()
botao_sair = tk.Button(frame_janela2, text="Sair do
Interface", command=janela_2.destroy)
botao_sair.pack()
label_janela_2 = tk.Label(frame_janela2, text="Indique a
localização do ficheiro com os dados do respetivo problema:")
label_janela_2.pack()
global painel_janela_2
painel_janela_2 = tk.Canvas(frame_janela2, width=490,
height=40) #Cria painel onde é possível desenhar
painel_janela_2.pack()
caixa = painel_janela_2.create_rectangle(10, 10, 490, 40,
fill="white") #Cria caixa branca no painel
texto = painel_janela_2.create_text(50, 25, text= 'Nada a
inserir' )
label_janela_2.place(x=10, y=5)
painel_janela_2.place(x=3, y=30)
botao_alterar_diretorio.place(x=510, y=35, width=75, height=40)
botao_sair.place(x=409, y=125, width=155, height=50)
botao_seguinte_janela_2.place(x=33, y=125, width=155,
height=50)
image_o3 = Image.open(resource_path("ISEG_Logo.png"))
image3 = ImageTk.PhotoImage(image_o3)
image_label2 = tk.Label(janela_2, image=image3, borderwidth=0)
image_label2.image = image3
image_label2.pack()
```

```
    if nome != '':
        caixa = painel_janela_2.create_rectangle(10, 10, 490, 40,
fill="white")
        texto = painel_janela_2.create_text(250, 25, text= nome )

def fechar_janela_3(): # Função que guarda o nome da pagina onde o
botão é clicado
    top_level_widget = botao_retroceder_janela_3.wininfo_toplevel()
    top_level_widget.destroy()

def fechar_janela_5(): # Função que guarda o nome da pagina onde o
botão é clicado
    top_level_widget = botao_retroceder_janela_5.wininfo_toplevel()
    top_level_widget.destroy()

def fechar_janela_1(): # Função que guarda o nome da pagina onde o
botão é clicado
    top_level_widget = entrar_botao.wininfo_toplevel()
    top_level_widget.destroy()

def fechar_janela_2(): # Função que guarda o nome da pagina onde o
botão é clicado
    top_level_widget = botao_seguinte_janela_2.wininfo_toplevel()
    top_level_widget.destroy()

janela_1 = tk.Tk() #Criar Janela Inicial
janela_1.title("Escalonamento de Docentes a Vigilâncias")
janela_1.state('zoomed')
image_o = Image.open(resource_path("login.jpg"))
image = ImageTk.PhotoImage(image_o)
image_label = tk.Label(janela_1, image=image)
image_label.place(x=0, y=0, relwidth=1, relheight=1)
r = 238
g = 238
b = 238
color = f"#{r:02x}{g:02x}{b:02x}"
frame = tk.Frame(janela_1, bg=color)
frame.place(relx=0.5, rely=0.5, anchor=tk.CENTER, width=600,
height=200)
```

```
introducao = tk.Label(frame, text="Bem vindo ao Software de
Otimização do Escalonamento de Docentes a Vigilancias no ISEG!")
#Inserir Label na janela
introducao.pack()
introducao.place(x=80, y=25)

image_o3 = Image.open(resource_path("ISEG_Logo.png"))
image3 = ImageTk.PhotoImage(image_o3)
image_label2 = tk.Label(janela_1, image=image3, borderwidth=0)
image_label2.image = image3
image_label2.pack()

#Criar Botão "Entrar"
entrar_botao = tk.Button(frame, text="Seguinte",
command=criar_janela_2)
entrar_botao.pack()
entrar_botao.place(x=120, y=75, width=100, height=50)

# Criar Botão "Sair"
sair_botao = tk.Button(frame, text="Sair",
command=janela_1.destroy)
sair_botao.pack()
sair_botao.place(x=380, y=75, width=100, height=50)

janela_1.mainloop() #Mostrar janela inicial
```

Anexo III

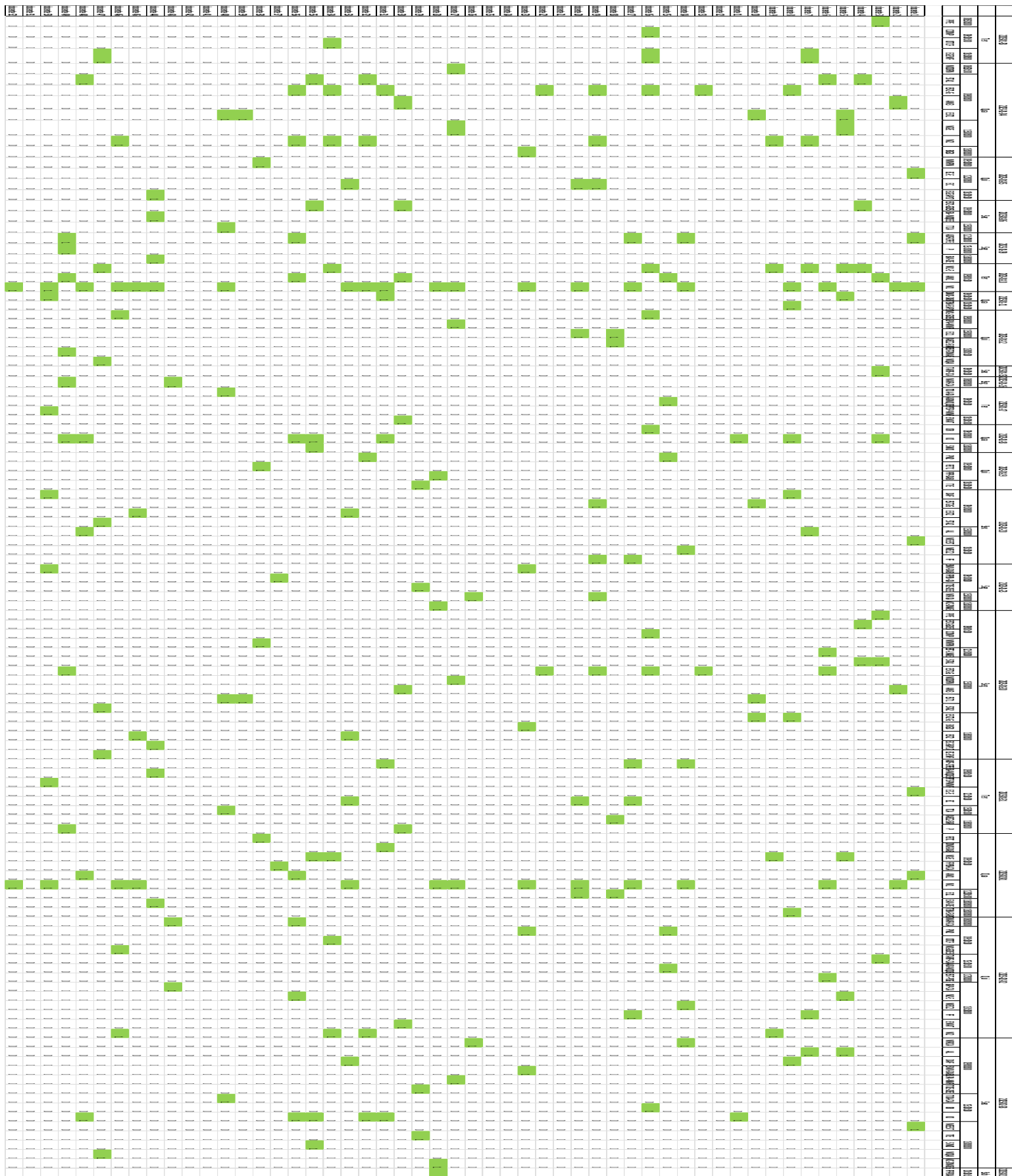


Figura 15 - Matriz de Escalonamento – Gurobi

Figura 16 - Matriz de Escalonamento – OR-Tools

Figura 17 - Matriz de Escalonamento – V.B.A.

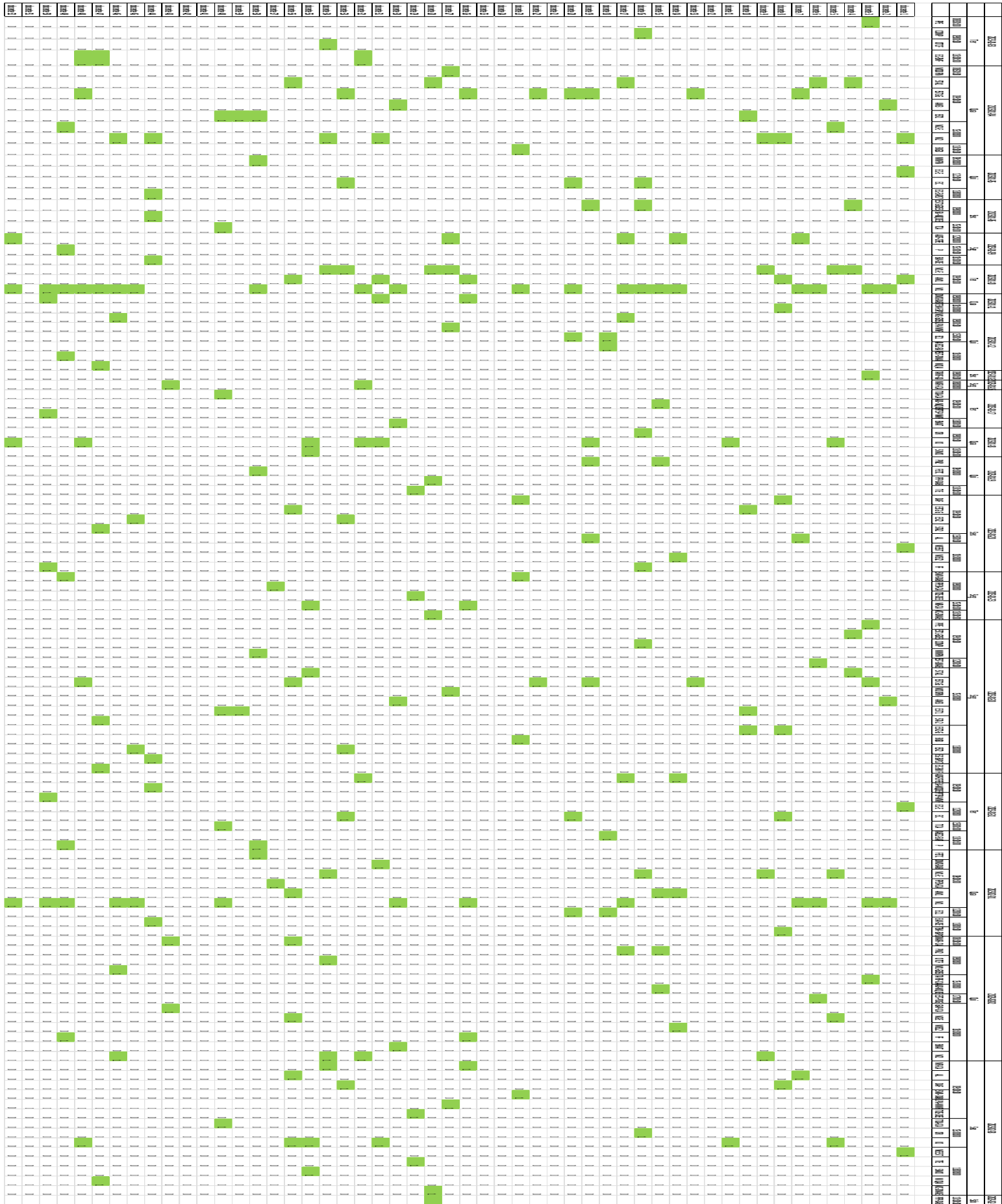


Figura 18 - Matriz de Escalonamento Ótimo – Modelo Alterado.

	2023-01-03	2023-01-04	2023-01-05	2023-01-06	2023-01-09	2023-01-10	2023-01-11	2023-01-12	2023-01-13	2023-01-16	2023-01-17	2023-01-18	2023-01-19	2023-01-20	2023-01-23	2023-01-30	2023-01-31	2023-02-01	2023-02-02	2023-02-03	2023-02-04
docente 1	0	0	1	0	1	1	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0
docente 2	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
docente 3	1	0	0	0	0	1	0	0	1	0	0	1	0	0	0	2	0	0	1	0	0
docente 4	0	1	0	1	0	1	0	0	1	0	0	0	0	0	0	2	0	0	0	0	0
docente 5	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
docente 6	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	2	0	0	1	1	0
docente 7	1	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	1	0
docente 8	0	1	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	1	1	0	0
docente 9	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	1	0	0
docente 10	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	2	0	0	0	0	0
docente 11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
docente 12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 13	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
docente 14	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	1	1	1	0
docente 15	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	0	0
docente 16	2	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	2	0	0	0	0
docente 17	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	1	1	0
docente 18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
docente 19	0	2	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	0
docente 20	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	1	2	0	0	0
docente 21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 22	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
docente 23	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	1	1	0
docente 24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
docente 27	0	2	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	1
docente 28	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	1
docente 29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	2
docente 30	0	1	0	1	0	1	0	0	1	0	1	0	0	0	0	1	1	0	1	1	0
docente 31	0	1	0	0	0	1	0	0	1	0	0	1	0	0	0	0	1	1	0	1	0
docente 32	0	2	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	1	0
docente 33	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	1	1	1	0	1	0
docente 34	1	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	2	0	0
docente 35	0	1	0	1	0	1	0	0	0	0	0	2	0	0	0	0	0	1	0	2	0
docente 36	0	2	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	1	2	1	0
docente 37	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
docente 38	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0
docente 39	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
docente 40	0	1	0	1	0	1	0	0	1	0	0	0	0	0	1	0	1	0	0	1	0
docente 41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 43	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	2	0	0
docente 44	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0
docente 45	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0
docente 46	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	2	0	0
docente 47	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
docente 48	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0
docente 49	0	0	0	0	0	2	0	0	1	0	0	1	0	0	0	1	1	0	0	0	0
docente 50	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	0	0	0
docente 51	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
docente 52	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0

Figura 19 - Dados Agrupados por Dia – Gurobi

	2023-01-03	2023-01-04	2023-01-05	2023-01-06	2023-01-09	2023-01-10	2023-01-11	2023-01-12	2023-01-13	2023-01-16	2023-01-17	2023-01-18	2023-01-19	2023-01-20	2023-01-23	2023-01-30	2023-01-31	2023-02-01	2023-02-02	2023-02-03	2023-02-04
docente 1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
docente 2	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
docente 3	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0
docente 4	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0
docente 5	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
docente 6	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0
docente 7	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
docente 8	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
docente 9	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
docente 10	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
docente 11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 13	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
docente 14	0	2	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
docente 15	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 16	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	2	1	0
docente 17	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 19	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 22	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 23	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 27	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 30	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 32	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 34	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 35	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 36	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 39	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 40	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 46	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 47	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 48	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 49	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 20 - Dados Agrupados por Dia – OR-Tools

	2023-01-03	2023-01-04	2023-01-05	2023-01-06	2023-01-09	2023-01-10	2023-01-11	2023-01-12	2023-01-13	2023-01-16	2023-01-17	2023-01-18	2023-01-19	2023-01-20	2023-01-23	2023-01-30	2023-01-31	2023-02-01	2023-02-02	2023-02-03	2023-02-04
docente 1	0	1	1	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
docente 2	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
docente 3	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0
docente 4	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0
docente 5	1	2	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
docente 6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 7	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0
docente 8	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
docente 9	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
docente 10	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0	0	0	0	0
docente 11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 13	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
docente 14	0	2	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0
docente 15	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0
docente 16	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	1	2	0
docente 17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 19	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 20	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 22	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 23	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 27	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 30	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 31	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 32	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 34	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 35	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0
docente 36	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
docente 37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 38	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 39	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 40	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 44	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 46	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 47	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 48	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 49	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 21 - Dados Agrupados por Dia – V.B.A.

	2023-01-03	2023-01-04	2023-01-05	2023-01-06	2023-01-09	2023-01-10	2023-01-11	2023-01-12	2023-01-13	2023-01-16	2023-01-17	2023-01-18	2023-01-19	2023-01-20	2023-01-23	2023-01-30	2023-01-31	2023-02-01	2023-02-02	2023-02-03	2023-02-04	
docente 1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 4	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 5	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 6	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 7	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 8	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 9	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 10	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 13	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 16	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 17	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 19	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 20	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 22	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 23	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 26	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 27	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 28	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 30	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 31	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 32	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 33	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 34	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 36	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 38	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 39	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 40	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 44	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 46	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 47	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 48	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 49	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
docente 52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 22 – Dados Agrupados por Dia – Modelo Alterado.

	<i>Gurobi</i>			<i>OR-Tools</i>			<i>VBA</i>				<i>M. Alterado</i>		
	0	1	2	0	1	2	0	1	2	3	0	1	2
<i>docente 1</i>	14	7	0	14	7	0	14	7	0	0	15	6	0
<i>docente 2</i>	17	4	0	17	4	0	17	4	0	0	17	4	0
<i>docente 3</i>	15	5	1	15	6	0	14	7	0	0	15	5	1
<i>docente 4</i>	17	3	1	17	3	1	17	3	1	0	17	3	1
<i>docente 5</i>	15	5	1	16	4	1	15	5	1	0	15	6	0
<i>docente 6</i>	16	4	1	15	6	0	16	5	0	0	16	5	0
<i>docente 7</i>	15	6	0	15	5	1	15	5	1	0	15	6	0
<i>docente 8</i>	13	8	0	14	6	1	14	6	1	0	13	8	0
<i>docente 9</i>	17	4	0	17	4	0	17	4	0	0	17	4	0
<i>docente 10</i>	18	2	1	18	2	1	18	2	1	0	18	2	1
<i>docente 11</i>	19	2	0	19	2	0	19	2	0	0	19	2	0
<i>docente 12</i>	21	0	0	21	0	0	21	0	0	0	21	0	0
<i>docente 13</i>	19	2	0	19	2	0	19	2	0	0	19	2	0
<i>docente 14</i>	14	7	0	15	5	1	15	5	1	0	15	6	0
<i>docente 15</i>	17	3	1	16	4	1	16	4	1	0	16	4	1
<i>docente 16</i>	14	5	2	13	7	1	13	7	1	0	12	9	0
<i>docente 17</i>	15	5	1	14	7	0	14	7	0	0	14	7	0
<i>docente 18</i>	18	2	1	18	2	1	18	2	1	0	18	2	1
<i>docente 19</i>	16	3	2	14	7	0	14	7	0	0	15	6	0
<i>docente 20</i>	16	4	1	15	6	0	15	6	0	0	15	6	0
<i>docente 21</i>	21	0	0	21	0	0	21	0	0	0	21	0	0
<i>docente 22</i>	19	2	0	19	2	0	19	2	0	0	19	2	0
<i>docente 23</i>	14	7	0	14	7	0	14	7	0	0	15	6	0
<i>docente 24</i>	21	0	0	21	0	0	21	0	0	0	21	0	0
<i>docente 25</i>	21	0	0	21	0	0	21	0	0	0	21	0	0
<i>docente 26</i>	19	2	0	19	2	0	19	2	0	0	14	7	0
<i>docente 27</i>	15	5	1	15	5	1	14	7	0	0	15	6	0
<i>docente 28</i>	15	6	0	15	6	0	15	6	0	0	15	6	0
<i>docente 29</i>	18	2	1	18	2	1	18	2	1	0	18	2	1
<i>docente 30</i>	14	7	0	14	7	0	15	6	0	0	15	6	0
<i>docente 31</i>	14	7	0	15	6	0	14	7	0	0	15	6	0
<i>docente 32</i>	16	4	1	15	5	1	16	4	0	1	15	6	0
<i>docente 33</i>	14	7	0	15	6	0	14	7	0	0	14	7	0
<i>docente 34</i>	16	3	2	15	5	1	15	5	1	0	15	5	1
<i>docente 35</i>	16	3	2	17	1	3	17	1	3	0	17	2	2
<i>docente 36</i>	14	5	2	14	5	2	14	5	2	0	14	5	2

docente 37	19	2	0	19	2	0	19	2	0	0	19	2	0
docente 38	17	4	0	16	5	0	16	5	0	0	14	7	0
docente 39	19	2	0	19	2	0	19	2	0	0	19	2	0
docente 40	14	7	0	14	7	0	14	7	0	0	14	7	0
docente 41	21	0	0	21	0	0	21	0	0	0	21	0	0
docente 42	21	0	0	21	0	0	21	0	0	0	21	0	0
docente 43	19	1	1	19	1	1	19	1	1	0	19	1	1
docente 44	14	7	0	14	7	0	14	7	0	0	14	7	0
docente 45	17	4	0	17	4	0	17	4	0	0	17	4	0
docente 46	16	4	1	16	4	1	16	4	1	0	16	4	1
docente 47	15	5	1	16	4	1	16	4	1	0	15	5	1
docente 48	15	6	0	14	7	0	15	6	0	0	15	6	0
docente 49	14	6	1	13	8	0	13	8	0	0	13	8	0
docente 50	14	7	0	14	7	0	15	6	0	0	15	6	0
docente 51	21	0	0	21	0	0	21	0	0	0	21	0	0
docente 52	19	2	0	19	2	0	19	2	0	0	17	4	0
Soma Valor	868	198	26	863	208	21	863	209	19	1	856	222	14

Tabela 2 - Número de ocorrências que o docente tem 0,1,2 ou 3 Vigilâncias/Dia