Uma imagem com Tipo de letra, Gráficos, texto, captura de ecrã

Descrição gerada automaticamente

**ALGAV REPORT**

**Sprint 2**

**Turma :** 3DE

**Grupo :** 027

André Filipe Magalhães da Silva – 1220633

Alexandre Ribeiro – 1220647

André Pereira – 1221275

Miguel Araújo - 1201291

**Índice**

**Conteúdo**

[**1 - Breve explicação do código básico para agendamento de cirurgias considerando apenas os médicos** 3](#_Toc183371115)

[**2 - Estudo de Complexidade (apenas com médicos)** 4](#_Toc183371116)

[**3 – Duas Heuristics (incluindo a comparação com a melhor solução)** 8](#_Toc183371117)

[**3.1 - Heuristic 1** 8](#_Toc183371118)

[**3.2 - Heuristic 2** 13](#_Toc183371120)

[**4 – Adaptação para todo o staff e fases da operação** 16](#_Toc183371121)

[**5. Conclusões** 19](#_Toc183371122)

# **1 - Breve explicação do código básico para agendamento de cirurgias considerando apenas os médicos**

O predicado principal do código base fornecido para a marcação de cirurgias é o predicado **schedule\_all\_surgeries/2**, em que os dois parâmetros são a sala de operações e o dia específico. Em primeiro lugar vai remover tudo o que se encontra nos predicados dinâmicos utilizando o predicado **retractall/1** e vaicopiar as informações de **agenda\_staff/3 para agenda\_staff1/3** e de **agenda\_operation\_room/3** para **agenda\_operation\_room1/3** utilizando o predicado **findall/3**, de forma a poderem ser manipuladas no futuro sem modificar a base de dados original.Em segundo lugar**,** determina a disponibilidade de cada membro do staff para um dia específico, através do predicado **availability/3** e vai ajustar as agendas dos médicos utilizando o predicado **adapt\_timetable/4** de acordo com os respetivos horários de trabalho. Através de outro predicado **findall/3**, vai ser obtida uma lista com todos os id’s de cirurgias a serem marcadas e em último lugar chama o predicado **availability\_all\_surgeries/3**.

O predicado **availability\_all\_surgeries/3** é que vai ser responsável por efetuar a marcação das cirurgias de forma recursiva para uma determinada sala de operações num determinado dia. Em primeiro lugar, este algoritmo, através de uma chamada do predicado **availability\_operation/5**, vai obter todos os horários disponíveis para a marcação da cirurgia numa determinada sala de operações num determinado dia. Em segundo lugar, vai encontrar o primeiro intervalo de tempo disponível através da utilização do predicado **schedule\_first\_interval/3**. De seguida, vai ser atualizada a agenda com a nova cirurgia utilizando o predicado **insert\_agenda/3** e vai ser atualizada a agenda dos médicos com o predicado **insert\_agenda\_doctors/3**. Este processo será então repetido de forma recursiva até que todas as cirurgias da lista tenham sido agendadas.

# **2 - Estudo de Complexidade (apenas com médicos)**

Este estudo tem como objetivo avaliar a capacidade do sistema de agendamento cirúrgico em lidar com cenários de complexidade crescente, analisando a sua escalabilidade e desempenho à medida que novas cirurgias são introduzidas. O processo de análise começará com três cirurgias simples, e novas operações serão adicionadas gradualmente para determinar o ponto em que o sistema continua a oferecer soluções ótimas ou aceitáveis para o agendamento.

Daremos atenção especial a cirurgias que requerem a colaboração de dois médicos, como s100004, s100005, s100008 e s100009, pois essas operações representam um desafio adicional ao sistema.

Por meio dessa abordagem incremental, busca-se identificar até que ponto o sistema consegue gerenciar a complexidade crescente, destacando limitações potenciais e oportunidades de melhoria no planejamento e na alocação de recursos médicos.

O algoritmo utilizado possui complexidade O(n!) devido ao uso de *permutation/2*, que gera todas as combinações possíveis das n cirurgias, resultando em n! ordens diferentes. Cada permutação envolve verificações e atualizações, mas o crescimento fatorial domina o tempo de execução, tornando o algoritmo inviável para um grande número de elementos. Por essa razão, o estudo inicia-se com um número reduzido de cirurgias.

Uma imagem com texto, Tipo de letra, captura de ecrã

Descrição gerada automaticamente

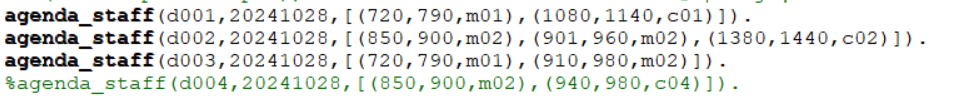
Uma imagem com texto, captura de ecrã, Tipo de letra

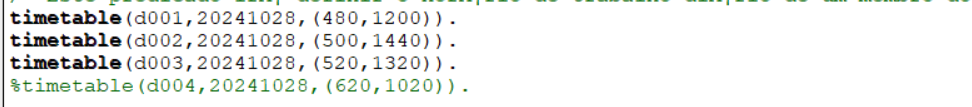
Descrição gerada automaticamente

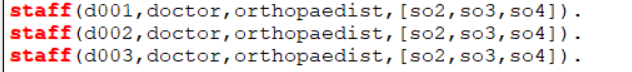
Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

**Dados utilizados no estudo da complexidade:**

****

****

****

****

**Uma imagem com texto, captura de ecrã, Tipo de letra, design

Descrição gerada automaticamente**

****

**Resultados da análise de complexidade:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nº of Surgeries** | **Nº of Solutions** | **Best Schedule of activities (including**  **surgeries) of the operation room** | **Final Time**  **for the last**  **Surgery**  **(minutes)** | **Time to**  **generate the**  **solution (s)** |
| 3 | 6 | [(480, 539, so100001), (540, 629, so100002), (630, 704, so100003)] | 704 | 0.07 |
| 4 | 24 | [(480,539,so100001),(540,614,so100003),  (615,674,so100004),(675,764,so100002)] | 764 | 0.17 |
| 5 | 120 | [(500, 559, so100004), (560, 634, so100003), (635, 709, so100005), (710, 799, so100002), (800, 859, so100001)] | 859 | 0.93 |
| 6 | 720 | [(500,559,so100004),(560,634,so100003),(635,709,  so100005),(710,799,so100002),(800,859,so100001),  (860,919,so100006)] | 919 | 5.45 |
| 7 | 5040 | [(500,559,so100004),(560,634,so100003),(635,709,  so100005),(710,799,so100002),(800,889,so100007),  (890,949,so100001),(950,1009,so100006)] | 1009 | 34.87 |
| 8 | 40320 | [(500,559,so100004),(560,649,so100007),(650,709,  so100008),(710,799,so100002),(800,874,so100003),  (875,934,so100001),(935,994,so100006),  (995,1069,so100005)] | 1069 | 76.94 |
| 9 | 362880 | [(480,539,so100001),(540,614,so100005),(620,679,  so100008),(680,769,so100002),(770,829,so100009),  (830,904,so100003),(905,964,so100006),(965,1024,  so100004),(1025,1114,so100007)] | 1114 | 381.64 |
| 10 | 3628800 | [(480,539,so100001),(540,629,so100007),(630,689,  so100008),(690,779,so100002),(780,839,so100009),  (840,899,so100010),(900,959,so100006),(961,1020,  so100004),(1021,1095,so100005),(1096,1170,so100003)] | 1170 | 1254.77 |
| 11 | 39916800 | [(480,539,so100001),(540,629,so100007),(630,689,  so100008),(690,779,so100002),780,839,so100009),  (840,899,so100006),(900,974  ,so100011),(975,1034,so100004),  (1035,1109,so100003),(1110,1184,so100005),  (1185,1244,so100010)] | 1244 | 8009.08 |

Como é possível observar, os resultados obtidos coincidem com o que era espectável, sendo que para ser gerado um resultado com apenas três cirurgias o resultado consegue ser apresentado em apenas 0.07 segundos em contraste ao resultado com 11 cirurgias que demora um tempo de 8009.08 segundos.

# **3 – Duas Heuristics (incluindo a comparação com a melhor solução)**

## **3.1 - Heuristic 1**

Para a primeira heurística é pedido que as cirurgias sejam priorizadas para o médico que esteja disponível o mais cedo possível, ou seja, o médico que tenha um intervalo de tempo disponível mais cedo que seja superior ao tempo da cirurgia. Por exemplo, se um médico começar a trabalhar às 8 horas e tiver uma reunião às 8h30, então não estará disponível para realizar uma cirurgia às 8h do tipo so2 que demora 60 minutos. Sendo assim, foram criados vários predicado para este efeito:

O predicado **obtain\_better\_sol\_heuristica1/5** terá a mesma função que o predicado **obtain\_better\_sol/5** do código original de funcionar como principal função e é responsável pela contabilização do tempo total do algoritmo e pelos outputs.

O predicado **obtain\_better\_sol\_heuristica1/2** terá a mesma função que o predicado **obtain\_better\_sol1/2** do código original, sendo responsável por obter e verificar cada sequência de cirurgias, utilizando desta vez a heurística em vez das permutações.

O predicado **update\_better\_sol\_heuristica1/4** exatamente igual ao predicado **original update\_better\_sol/4**.

E predicado **prioritize\_surgeries/3**, que será responsável por ordenar as cirurgias com base no tempo inicial de disponibilidade do médico, também na duração da própria cirurgia, em que as cirurgias com menor duração terão prioridade, pois terminarão mais cedo, e todo o tempo disponível do médico, médicos com menor tempo disponível terão prioridade, utilizando o predicado **findall/3** para obter todas as cirurgias para cada médico , os tempos iniciais de cada intervalo de tempo de disponibilidade, a duração de cada cirurgia e o total de tempo disponível de cada médico, utilizando ainda outros predicados como **compare\_priority/3**, que será utilizado para fazer a ordenação da lista das cirurgias, o predicado **surgery\_duration/2**, que vai obter o tempo de duração da cirurgia (em minutos) a partir do seu id e o predicado **extract\_opcodes/2**, que irá extrair os id’s das cirurgias do “tuple” criado no predicado **findall/3**.

Uma imagem com texto, captura de ecrã

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **N. of surgeries** | **Optimal Solution** | **Final Time for the last Surgery In generate all select better (minutes)** | **Final Time for the last Surgery Using Heuristic (minutes)** | **Time to generate the best solution (s)** | **Time to generate the heuristic solution (s)** | **Solution with the Heuristic** |
| 3 | [(480, 539, so100001), (540, 629, so100002), (630, 704, so100003)] | 704 | 704 | 0.07 | 0.060 | [(480,539,so100001),(540,614,so100003),(615,704,so100002)] |
| 4 | [(480,539,so100001),(540,614,so100003),  (615,674,so100004),(675,764,so100002)] | 764 | 764 | 0.17 | 0.037 | [(480,539,so100001),(540,599,so100004),(600,674,so100003),(675,764,so100002)] |
| 5 | [(500, 559, so100004), (560, 634, so100003), (635, 709, so100005), (710, 799, so100002), (800, 859, so100001)] | 859 | 1055 | 0.93 | 0.039 | [(480,539,so100001),(540,599,so100004),(600,674,so100003),(675,764,so100002),(981,1055,so100005)] |
| 6 | [(500,559,so100004),(560,634,so100003),(635,709,  so100005),(710,799,so100002),(800,859,so100001),  (860,919,so100006)] | 919 | 1055 | 5.45 | 0.042 | [(480,539,so100001),(540,599,so100004),(600,659,so100006),(660,749,so100002),(791,865,so100003),(981,1055,so100005)] |
| 7 | [(500,559,so100004),(560,634,so100003),(635,709,  so100005),(710,799,so100002),(800,889,so100007),  (890,949,so100001),(950,1009,so100006)] | 1009 | 1145 | 34.87 | 0.045 | [(480,539,so100001),(540,599,so100004),(600,659,so100006),(660,749,so100002),(791,865,so100003),(981,1055,so100005),(1056,1145,so100007)] |
| 8 | [(480,539,so100001),(540,614,so100005),(620,679,  so100008),(680,769,so100002),(770,829,so100009),  (830,904,so100003),(905,964,so100006),(965,1024,  so100004),(1025,1114,so100007)] | 1069 | 1220 | 76.94 | 0.072 | [(480,539,so100001),(540,599,so100004),(620,679,so100008),(680,769,so100002),(791,850,so100006),(981,1055,so100003),(1056,1130,so100005),(1131,1220,so100007)] |
| 9 | [(480,539,so100001),(540,614,so100005),(620,679,  so100008),(680,769,so100002),(770,829,so100009),  (830,904,so100003),(905,964,so100006),(965,1024,  so100004),(1025,1114,so100007)] | 1114 | 1310 | 381.64 | 0.039 | [(480,539,so100001),(540,599,so100004),(620,679,so100008),(680,739,so100009),(791,850,so100006),(981,1055,so100003),(1056,1130,so100005),(1131,1220,so100007),(1221,1310,so100002)] |
| 10 | [(480,539,so100001),(540,629,so100007),(630,689,  so100008),(690,779,so100002),(780,839,so100009),  (840,899,so100010),(900,959,so100006),(961,1020,  so100004),(1021,1095,so100005),(1096,1170,so100003)] | 1170 | 1350 | 1254.77 | 0.045 | [(520,579,so100010),(620,679,so100008),(680,739,so100009),(791,850,so100001),(851,910,so100006),(961,1020,so100004),(1021,1095,so100003),(1096,1170,so100005),(1171,1260,so100007),(1261,1350,so100002)] |

Comparando os valores obtidos com os valores da melhor solução possível, verificamos que apesar de a solução não ser a melhor possível em termos de tempo final da última cirurgia, será bastante mais eficiente em termos de demora do algoritmo demorando apenas algumas frações de segundo, para o mesmo número de cirurgias em que o algoritmo utilizando as permutações demorava horas.

## **3.2 - Heuristic 2**

Nesta seção, será apresentada uma heurística desenvolvida para a otimização do agendamento de cirurgias. A heurística utiliza técnicas baseadas em geração de combinações de operações, verificação de restrições e avaliação iterativa de soluções. Durante a sua execução, são considerados fatores como a disponibilidade de médicos e salas o tempo ocupado por cada médico e tempo final das operações realizadas.

De seguida, vamos explicar os predicados mais revelantes responsáveis por implementar essa heurística. De ressaltar que o código que vai ser explicado a seguir tem apenas em conta os doutores, ainda não tem em conta o resto do staff para uma operação.

O processo é conduzido pelos seguintes predicados principais:

**obtain\_better\_sol\_heuristica2/5**: Coordena a execução geral da heurística. Mede o tempo de execução, chama o predicado principal de geração de soluções (obtain\_better\_sol2/2), recupera a melhor solução armazenada e calcula o tempo final da última cirurgia com **calcular\_tempo\_final\_realizado/3**.

**obtain\_better\_sol2/2**: Gera combinações de cirurgias, começando com uma solução inicial padrão armazenada com asserta(better\_sol/5). Ele coleta os códigos de cirurgia, testa diferentes permutações (permutation/2), e atualiza as agendas de médicos e salas. Após verificar a viabilidade das combinações com **availability\_all\_surgeries/3**, a solução é avaliada e possivelmente melhorada usando **update\_better\_sol2/4**.

**update\_better\_sol2/4**: Compara a solução atual com a armazenada. Utiliza **calcular\_percentagem\_maxima/3** para determinar a eficiência da alocação com base na ocupação máxima dos médicos. Se a solução atual for melhor, ela substitui a armazenada.

**calcular\_percentagem\_maxima/3**: Calcula o maior percentual de ocupação dos médicos considerando suas agendas e horários disponíveis. Isso é usado como métrica para avaliar a qualidade de cada solução.

**calcular\_tempo\_final\_realizado/3**: Identifica o maior tempo final entre as cirurgias realizadas, representando o término do uso da sala naquele dia.

Esses predicados trabalham juntos para testar combinações, respeitar restrições de disponibilidade e atualizar iterativamente a solução armazenada

**Código utilizado:**

**Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente**

**Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente**

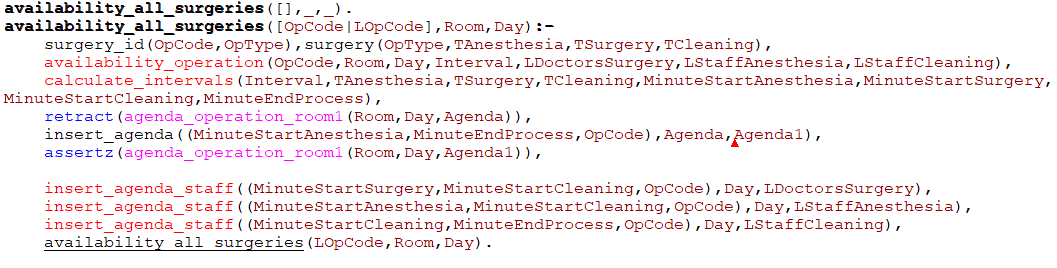
Após esta apresentação do predicado que implementa a heurística, a tabela abaixo contém os resultados obtidos durante os testes. Esta tabela resume as soluções encontradas, comparando os tempos de finalização da última cirurgia e o desempenho da heurística em diferentes cenários.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **N. of**  **surgeries** | **Optimal solution** | **Final Time for the last Surgery In generate all select better (minutes)** | **Final Time for the last Surgery Using Heuristic (minutes)** | **Time to generate the best solution (s)** | **Time to generate the heuristic solution (s)** | **Solution with the Heuristic 2** |
| 3 | [(480, 539, so100001), (540, 629, so100002), (630, 704, so100003)] | 704 | 704 | 0.07 | 0.0064 | [(480,539,so100001),(540,629,so100002),(630,704,so100003)] |
| 4 | [(480,539,so100001),(540,614,so100003),  (615,674,so100004),(675,764,so100002)] | 764 | 1020 | 0.17 | 0.0028 | [(480,539,so100001),(540,629,so100002),(630,704,so100003),(961,1020,so100004)] |
| 5 | [(500, 559, so100004), (560, 634, so100003), (635, 709, so100005), (710, 799, so100002), (800, 859, so100001)] | 859 | 1095 | 0.93 | 0.015 | [(480,539,so100001),(540,629,so100002),(630,704,so100003),(961,1020,so100004),(1021,1095,so100005)] |
| 6 | [(500,559,so100004),(560,634,so100003),(635,709,  so100005),(710,799,so100002),(800,859,so100001),  (860,919,so100006)] | 919 | 1095 | 5.45 | 0.089 | [(480,539,so100001),(540,629,so100002),(630,704,so100003),(791,850,so100006),(961,1020,so100004),(1021,1095,so100005)] |
| 7 | [(500,559,so100004),(560,634,so100003),(635,709,  so100005),(710,799,so100002),(800,889,so100007),  (890,949,so100001),(950,1009,so100006)] | 1009 | 1185 | 34.87 | 0.633 | [(480,539,so100001),(540,629,so100002),(630,704,so100003),(791,850,so100006),(961,1020,so100004),(1021,1095,so100005),(1096,1185,so100007)] |
| 8 | [(500,559,so100004),(560,649,so100007),(650,709,  so100008),(710,799,so100002),(800,874,so100003),  (875,934,so100001),(935,994,so100006),  (995,1069,so100005)] | 1069 | 1185 | 76.94 | 4.483 | [(480,539,so100001),(540,629,so100002),(630,689,so100008),(791,865,so100003),(866,925,so100006),(961,1020,so100004),(1021,1095,so100005),(1096,1185,so100007)] |
| 9 | [(480,539,so100001),(540,614,so100005),(620,679,  so100008),(680,769,so100002),(770,829,so100009),  (830,904,so100003),(905,964,so100006),(965,1024,  so100004),(1025,1114,so100007)] | 1114 | 1185 | 381.64 | 39.445 | [(480,539,so100001),(540,629,so100002),(630,689,so100008),(690,749,so100009),(791,865,so100003),(866,925,so100006),(961,1020,so100004),(1021,1095,so100005),(1096,1185,so100007)] |
| 10 | [(480,539,so100001),(540,629,so100007),(630,689,  so100008),(690,779,so100002),(780,839,so100009),  (840,899,so100010),(900,959,so100006),(961,1020,  so100004),(1021,1095,so100005),(1096,1170,so100003)] | 1170 | 1245 | 1254.77 | 431.21 | [(480,539,so100001),(540,629,so100002),(630,689,so100008),(690,749,so100009),(791,865,so100003),(866,925,so100006),(961,1020,so100004),(1021,1095,so100005),(1096,1185,so100007),(1186,1245,so100010)] |
| 11 | [(480,539,so100001),(540,629,so100007),(630,689,  so100008),(690,779,so100002),780,839,so100009),  (840,899,so100006),(900,974  ,so100011),(975,1034,so100004),  (1035,1109,so100003),(1110,1184,so100005),  (1185,1244,so100010)] | 1244 | 1285 | 8009.08 | 4982.3 | [(480,539,so100001),(540,629,so100002),(630,689,so100008),(690,749,so100009),(791,865,so100003),(866,925,so100006),(926,1000,so100011),(1001,1060,so100004),(1061,1135,so100005),(1136,1225,so100007),(1226,1285,so100010)] |

Vendo e comparando os valores da tabela conseguimos perceber que a heurística que fizemos gera uma solução mais rapidamente do que a “optimal solution”.

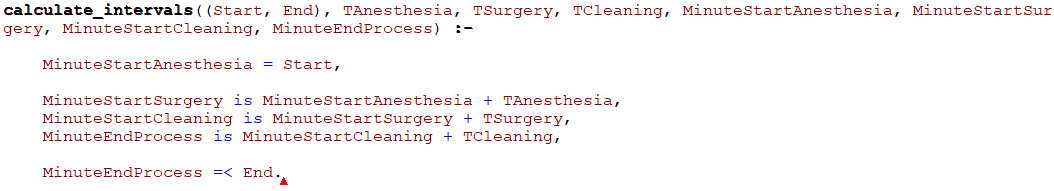
# **4 – Adaptação para todo o staff e fases da operação**

Para a inclusão de todo o staff teremos que considerar que, por exemplo, num intervalo necessário de uma cirurgua *Hip Replacement* de 165 mins, teremos uma primeira fase de 45 mins de duração de anestesia e preparação do paciente, em que será necessário 1 Dr. Anestesista e 1 enfermeiro anestesista. Na segunda fase, a cirurgia, os mesmos dois membros do staff da primeria fase estarão presentes acompanhados também de 2 Dr. Ortopedia, 1 enfermeiro instrumentador e 1 enfermeiro circulante. Passando esta fase, após 120 mins do início do procedimento, teremos a fase final de limpeza e desinfetação do bloco operatório, em que apenas está presente 1 assistente médico.

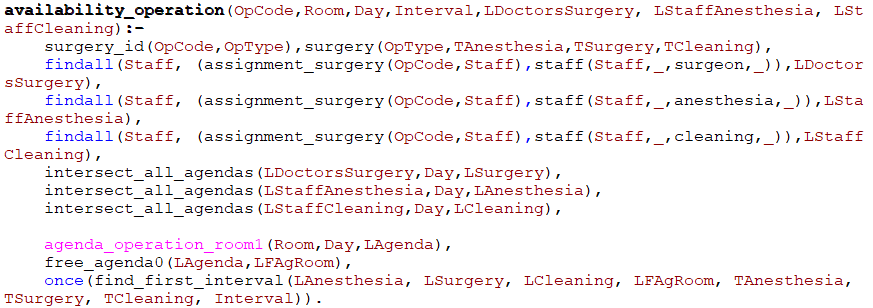


Passando aos predicados, primeiramente no ***availability\_all\_surgeries/3***, para poder incluir estes novos membros do staff modificamos/adicionamos também 2 predicados auxiliares, ***availability\_operation/7*** e ***calculate\_intervals/8*** para adaptar às adicionais fases da operação e staff necessário com a mesma funcionalidade inicial: encontrar o primeiro intervalo disponível para a realização de todo o procedimento de operação.

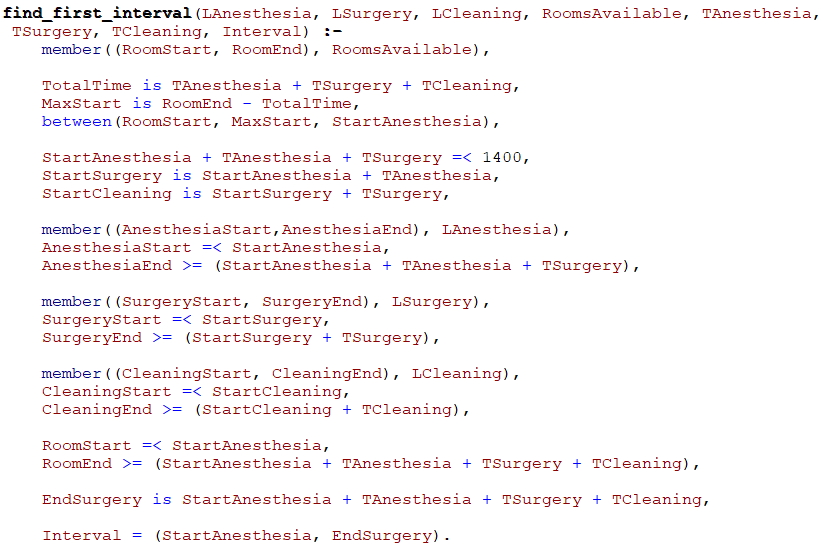
O predicado auxiliar ***calculate\_intervals/8*** foi criado para fins de organização do código e que serve para definir os instantes de início de cada fase da operação.



Alteramos também o predicado ***availability\_operation/7*** para proceder à inclusão do staff, começando por definir as listas de staff requerido e usando agora 3 instâncias do predicado ***insert\_all\_agendas/3*** para cada tipo de staff das 3 fases.



Também definimos a disponibilidade da sala passada como argumento e utilizamos o predicado auxiliar ***find\_first\_interval/8*** para encontrar o primeiro intervalo de tempo válido para a realização da operação. Como este predicado pode ter várias soluções, o ***availability\_operation/7*** usa o *once* para retornar a primeira instância de sucesso do predicado.



Neste predicado ***find\_first\_interval/8*** vamos encontrar um intervalo em que a operação pode decorrer.

Aqui, definimos o tempo total da operação e verificamos se esse tempo total não se estende para um intervalo em que a sala esteja indisponível seja por ter outra operação agendada ou por ser fora do horário de trabalho, indo fazer o mesmo para as 3 fases da operação. Além disso, ainda fazemos uma verificação da disponibilidade de cada tipo de staff necessário aos procedimentos. Após verificarmos se a sala tem disponibilidade considerando o instante de início dos procedimentos, retornamos o intervalo de tempo válido.

# **5. Conclusões**

Foi desenvolvida uma solução para a marcação de operações considerando inicialmente apenas médicos, evoluindo depois para incluir todo o staff necessário e as fases de cada tipo de operação.Estudou-se também a complexidade do problema de agendamento (apenas com a solução a considerar médicos) e propuseram-se duas heurísticas para melhorá-lo, comparando-as com a solução ótima.

A solução inicial revelou-se eficaz para médicos, mas a inclusão de outras restrições (como todo o pessoal necessário, e fases) fez nos reescrever/modificar o código. As heurísticas propostas demonstraram-se capazes de encontrar soluções próximas da ótima em menor tempo, destacando a sua aplicabilidade em contextos reais onde o tempo de resposta é crítico.

A adaptação para incluir todos os profissionais de saúde e fases do processo mostrou-se desafiadora, mas permitiu uma abordagem mais realista e abrangente ao problema de agendamento.