Uma imagem com Tipo de letra, Gráficos, design gráfico, captura de ecrã

Descrição gerada automaticamente

**Relatório ALGAV Sprint 2&3**

Turma 3DD - Grupo 23

**Aluno:**

Rafael Brandao, 1220879

**Professor:**

Francisco da Silva, FPS

**Unidade Curricular:**

Algoritmia Avançada

Índice

[Sprint 2 3](#_Toc186751418)

[US 6.3.1 3](#_Toc186751419)

[US 6.3.2 3](#_Toc186751421)

[Sprint 3 3](#_Toc186751423)

[US 7.3.1 3](#_Toc186751424)

[US 7.3.2 3](#_Toc186751426)

# Sprint 2

## US 6.3.1

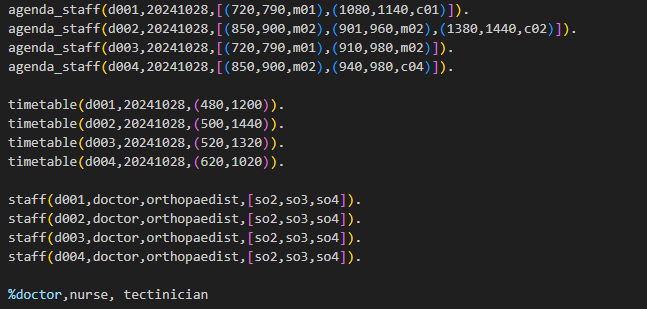
Requisitos: “*As an Admin, I want to obtain the better scheduling of a set of operations (surgeries) in a certain operation room in a specific day.”*

O código fornecido implementa um sistema robusto para o escalonamento de cirurgias em salas de operações, considerando as agendas dos médicos e a disponibilidade das salas de cirurgia.

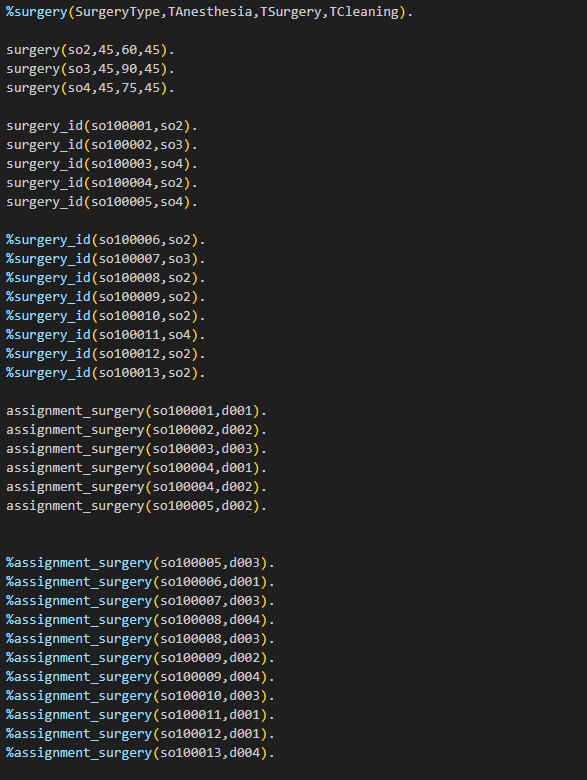
Sendo assim, o funcionamento deste sistema pode ser descrito da seguinte forma:

**Definição dos Dados Base:**

1. **Agendas dos Médicos**:
   * O predicado agenda\_staff/3 especifica os horários ocupados para cada médico em um determinado dia. Cada entrada indica intervalos de tempo já reservados para atividades específicas.
   * O predicado timetable/3 define o horário de trabalho permitido para cada médico em um determinado dia.
   * O predicado staff/3 representa as informações especificas dos trabalhadores do hospital, como a função (doutor, enfermeiro, técnico) e as suas especialidades



1. **Cirurgias e Associações**:
   * O predicado surgery/4 descreve os tempos necessários para anestesia, cirurgia e limpeza para cada tipo de cirurgia.
   * A relação entre tipos de cirurgia e cirurgias específicas é definida pelo predicado surgery\_id/2.
   * O predicado assignment\_surgery/2 associa médicos às cirurgias que podem realizar.

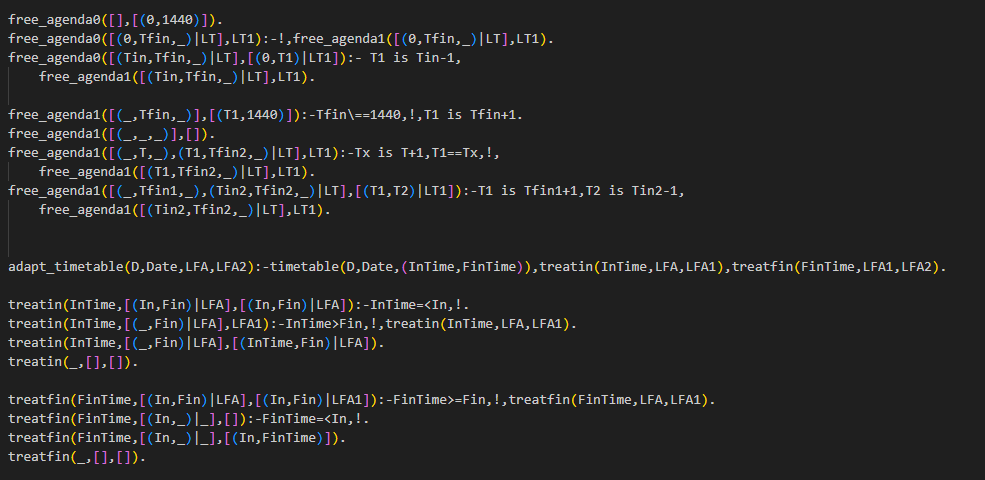


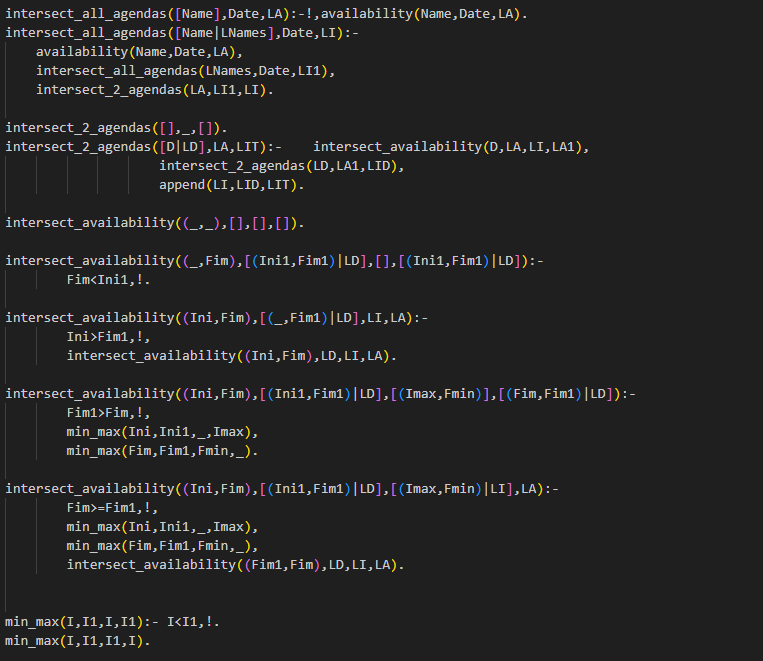
1. **Ocupação das Salas de Operação**:
   * O predicado agenda\_operation\_room/3 lista os horários em que uma sala de operação já está ocupada em um dia específico.



**Gestão das Agendas:**

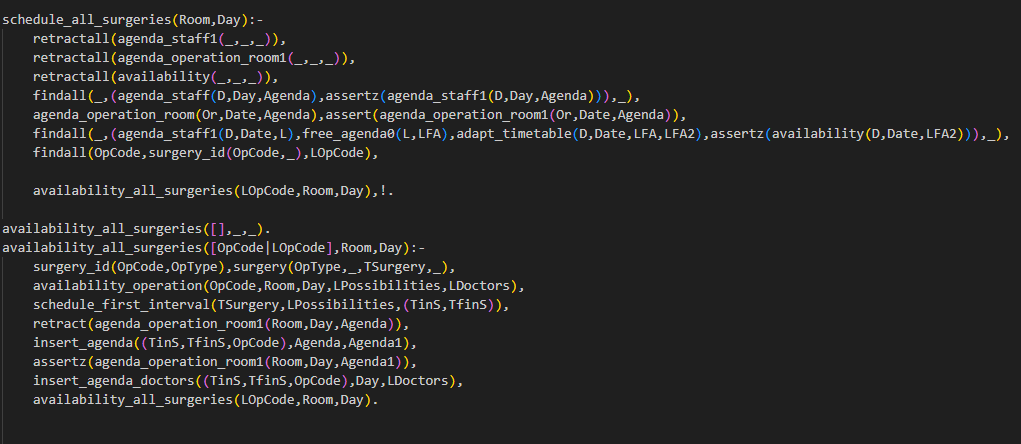
1. **Identificação de Intervalos Livres**:
   * free\_agenda0/2 processa as agendas ocupadas para determinar os intervalos de tempo disponíveis.
   * Caso as agendas não possuam horários é assumido que o horário livre é o dia todo e caso contrário usa free\_agenda1/2 para calcular os horários livres a partir do primeiro horário ocupado.
   * adapt\_timetable/4 ajusta os intervalos livres para coincidir com as agendas de cada médico e operações sem sobreposição. Adicionalmente, usamos treatin/3 e treatfin/3 de modo a verificar se o horário de entrada ou de fim de uma operação também ser colocado na agenda.



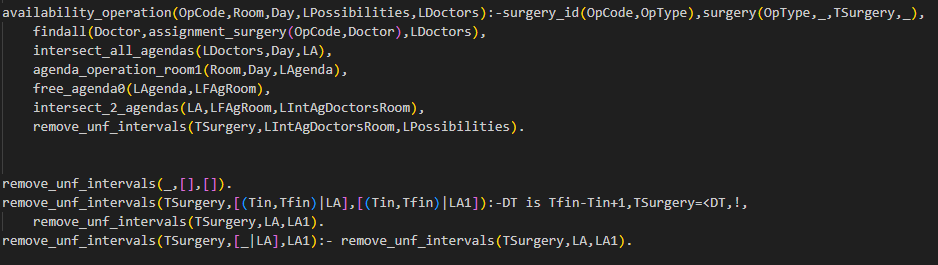
1. **Interseção de Agendas**:
   * intersect\_all\_agendas/3 combina as disponibilidades de diferentes médicos para encontrar horários em comum.
   * intersect\_2\_agendas/3 calcula a interseção entre duas agendas específicas, refinando os horários disponíveis.

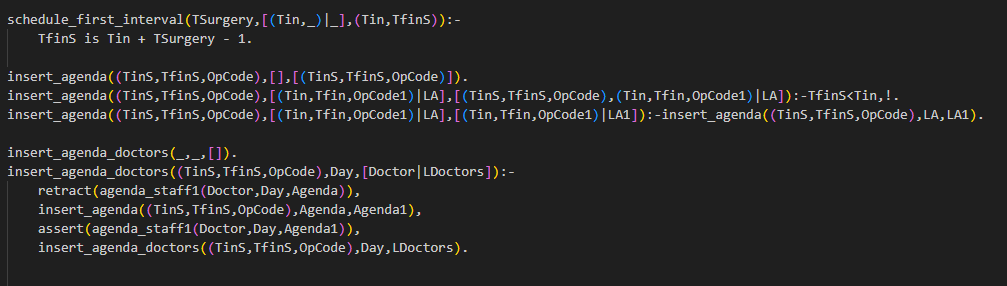
**Agendamento de Cirurgias:**

1. **Marcação de Cirurgias**:
   * O predicado principal, schedule\_all\_surgeries/2, realiza o escalonamento de todas as cirurgias para uma sala de operação num dia específico.
   * Este predicado copia as agendas iniciais, calcula os intervalos livres para médicos e salas (availability\_all\_surgeries/3), e insere as cirurgias em horários válidos.



1. **Determinação de Horários Disponíveis**:
   * availability\_operation/5 avalia os horários possíveis para uma cirurgia, considerando a disponibilidade de médicos/salas.



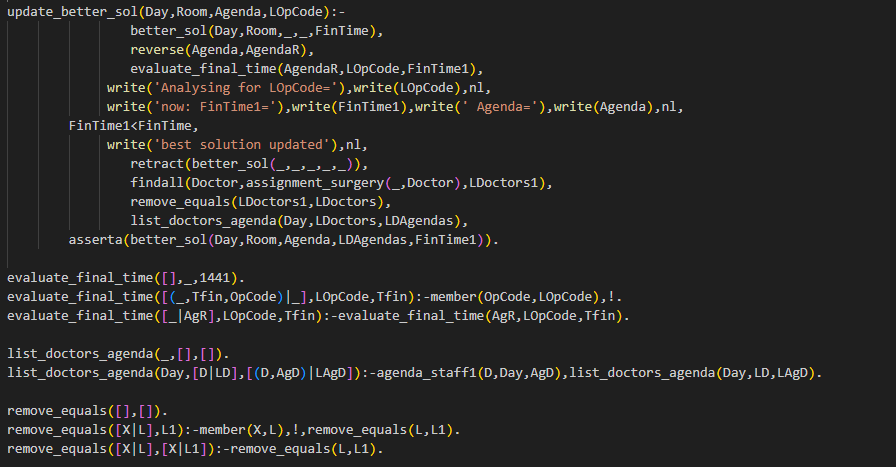
1. **Atualização das Agendas**:
   * insert\_agenda/3 insere uma cirurgia nas agendas dos médicos e/ou salas.
   * insert\_agenda\_doctors/3 insere o horário da mesma cirurgia nas agendas dos médicos lhe designados.

**Busca pela Melhor Solução:**

1. **Minimização do Tempo Total**:
   * O predicado obtain\_better\_sol/5 encontra a melhor sequência de cirurgias de modo a minimizar o tempo total necessário para as realizar.
   * permutation/2 é usado para gerar todas as possíveis ordens de execução das cirurgias.
   * evaluate\_final\_time/3 calcula o final da última cirurgia em cada sequência, o que permite a comparação de soluções.



1. **Atualização da Melhor Solução**:
   * O predicado update\_better\_sol/5 armazena a solução atual caso seja melhor do que a anterior, tendo em consideração o menor tempo total para as operações.



## US 6.3.2

Requisitos: “*As an Admin, I want to know till what dimension in terms of number of surgeries is possible to ask for the better solution.”*

Resultados de uma análise de complexidade do código base:

|  |  |  |  |
| --- | --- | --- | --- |
| Nº de cirurgias | Melhor Resultado | Tempo Final (min) | Tempo de geração |
| 1 | [(520, 579, so100000), (580, 639, so100001), (1000, 1059, so099999)] | 639 | 0.012403964 |
| 2 | [(520, 579, so100000), (580, 639, so100001), (640, 729, so100002), (1000, 1059, so099999)] | 729 | 0.020236968 |
| 3 | [(520, 579, so100000), (580, 639, so100001), (640, 714, so100003), (715, 804, so100002), (1000, 1059, so099999)] | 804 | 0.044281005 |
| 4 | [(520, 579, so100000), (580, 654, so100003), (655, 714, so100004), (715, 804, so100002), (805, 864, so100001), (1000, 1059, so099999)] | 864 | 0.290410995 |
| 5 | [(520, 579, so100000), (580, 639, so100004), (640, 714, so100005), (715, 804, so100002), (805, 879, so100003), (880, 939, so100001), (1000, 1059, so099999)] | 939 | 0.349401951 |
| 6 | [(520, 579, so100000), (580, 639, so100004), (640, 714, so100005), (715, 804, so100002), (805, 879, so100003), (880, 939, so100001), (940, 999, so100006), (1000, ..., ...)] | 999 | 2.22795701 |
| 7 | [(520, 579, so100000), (580, 639, so100004), (640, 714, so100005), (715, 804, so100002), (805, 879, so100003), (880, 939, so100001), (940, 999, so100006), (1000, ..., ...), (..., ...)] | 1149 | 6.533479915 |
| 8 | [(520, 579, so100000), (580, 639, so100004), (640, 699, so100008), (700, 789, so100002), (791, 865, so100003), (866, 925, so100001), (926, 985, so100006), (1000, ..., ...), (..., ...)|...] | 1224 | 13.66150604 |
| 9 | [(520, 579, so100000), (580, 639, so100004), (640, 699, so100008), (700, 789, so100002), (790, 849, so100009), (850, 909, so100001), (910, 969, so100006), (1000, ..., ...), (..., ...)|...] | 1299 | 48.07397785 |
| 10 | [(520, 579, so100000), (580, 639, so100004), (640, 699, so100008), (700, 759, so100009), (791, 865, so100003), (866, 925, so100001), (926, 985, so100006), (1000, ..., ...), (..., ...)|...] | 1374 | 388.55271291 |

**Análise da Complexidade**

1. **Tempo Final:**
   * O **tempo final** representa a eficiência do algoritmo a ordenar as cirurgias de forma a minimizar o tempo total necessário para as completar.
   * Observa-se um **crescimento linear** do tempo final, ou seja, o algoritmo consegue gerar soluções eficientes mesmo com o aumento do número de cirurgias.
2. **Tempo de Geração da Solução:**
   * O **tempo de geração** dita o tempo necessário para encontrar a melhor solução.
   * Aqui, o crescimento é **exponencial**, o que é consistente com a abordagem do algoritmo, pois utiliza permutações para avaliar todas as possíveis ordens das cirurgias.
   * Podemos concluir que a complexidade do algoritmo é O(N!×N), onde N! é o número de permutações para N cirurgias N, onde se calcula o o tempo total de cada cirurgia.

Assim sendo, conseguimos perceber que a partir de 10 cirurgias o tempo de processamento torna-se excessivamente grande, o que remove a viabilidade deste algoritmo.

# Sprint 3

## US 7.3.1

Requisitos: “*As an Admin, I want an automatic method to assign a set of operations (surgeries) to several operation rooms (assign is just to decide in which operation room the surgery will be done).”*

## US 7.3.2

## Requisitos: *“As an Admin, I want to be able to schedule surgeries to several operations rooms using Genetic Algorithms (Genetic Algorithm parameters need to be tuned according to conditions like number of genes, desired time for solution, etc.)”*