***Algav***

2024/2025

Sprint 2

**Turma 3DC, Grupo 17**

Matias Vitorino – 1220727

João Morais – 1221366

Pedro Azevedo – 1221264

Hugo Barros - 1220667

**Índice**

[Introdução 3](#_Toc183191987)

[US 6.3.1 4](#_Toc183191988)

[US 6.3.2 5](#_Toc183191989)

[US 6.3.3 7](#_Toc183191990)

# **Introdução**

Neste relatório, apresentaremos as implementações relacionadas ao módulo de planejamento (Planning Module), descrito na seção 6.3 das User Stories. O foco principal deste módulo é desenvolver um sistema eficiente para o agendamento de cirurgias, garantindo que as restrições, como disponibilidade de equipe e sala, sejam respeitadas.

A solução visa atender diferentes requisitos: desde a obtenção do melhor agendamento possível, até a geração de uma solução "boa" em tempo útil, utilizando métodos heurísticos ou algoritmos informados. Além disso, inclui-se uma análise de complexidade para determinar o limite viável de dimensionamento do problema.

Cada User Story será detalhada, abordando o raciocínio, os métodos aplicados e os resultados alcançados durante o desenvolvimento deste módulo.

**US 6.3.1**

**Objetivo**

A user story 6.3.1 tem como objetivo permitir que um administrador obtenha o melhor agendamento possível para um conjunto de cirurgias numa sala de operação específica, num determinado dia.

Tendo em consideração:

1. Disponibilidade de médicos e outros profissionais.
2. Horários e limitações das salas de operação.
3. Duração e requisitos específicos de cada cirurgia.

**Estrutura do Predicado**

Os predicados usados na solução foram projetados para modelar os seguintes aspetos:

1. Disponibilidade do staff e das salas de operação.
2. Características das cirurgias e suas alocações.
3. Lógica de combinação e agendamento dos recursos.

**1. Predicados Dinâmicos**

Esses predicados são utilizados para armazenar informações durante a execução do programa:

* **agenda\_staff/3**: Representa a agenda de disponibilidade de cada membro do staff.

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

Descrição gerada automaticamente

Figura 1 – predicado agenda\_staff

* **agenda\_operation\_room/3**: Indica a ocupação das salas de operação num dia específico.

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

Descrição gerada automaticamente

Figura 1 – predicado agenda\_operarion\_room

* **surgery/4:** definem o tipo da cirurgia, sua duração e o tempo de preparação e limpeza associados a ela.

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

Descrição gerada automaticamente

Figura 1 – predicado surgery

* **surgery\_id/2:** mapeia o ID de cada cirurgia para o tipo correspondente

Uma imagem com texto, Tipo de letra, branco, algebra

Descrição gerada automaticamente

Figura 1 – predicado surgery\_id

* **assignment\_surgery/2:** faz a associação entre médicos e as cirurgias que devem realizar

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

Descrição gerada automaticamente

Figura 1 – predicado assigment\_surgery

**2. Predicados de Disponibilidade**

* **free\_agenda0/2**: Calcula os períodos livres numa agenda.

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

Descrição gerada automaticamente

Figura 1 – predicado free\_agenda

* **intersect\_all\_agendas/3**: Determina os intervalos comuns de disponibilidade para um conjunto de membros do staff.

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

Descrição gerada automaticamente

Figura 1 – predicado intersect\_all\_agendas

* **adapt\_timetable/4**: Ajusta a agenda de um membro do staff com base no horário de trabalho.



Figura 1 – predicado adapt\_timetable

Esses predicados garantem que todas as restrições de tempo sejam respeitadas.

**3. Predicados de Agendamento**

* **schedule\_all\_surgeries/2**: Principal predicado que agenda todas as cirurgias de uma sala para um dia específico.

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

Descrição gerada automaticamente

Figura 1 – predicado Schedule\_all\_surgeries

* **availability\_operation/5**: Calcula os períodos possíveis para agendar uma cirurgia considerando médicos e salas.

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

Descrição gerada automaticamente

Figura 1 – predicado availability\_operation

* **schedule\_first\_interval/3**: Seleciona o primeiro intervalo disponível para uma cirurgia.



Figura 1 – predicado achedule\_first\_interval

Esses predicados organizam as cirurgias, associando-as aos identificadores e atribuindo médicos responsáveis.

**4. Predicado para Solução Otimizada**

* **obtain\_better\_sol/5**: Avalia todas as permutações possíveis das cirurgias para encontrar a melhor solução (menor tempo total).



Figura 1 – predicado obtain\_better\_sol

* **update\_better\_sol/5**: Atualiza a melhor solução encontrada até o momento.

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

Descrição gerada automaticamente

Figura 1 – predicado update\_better\_sol

Esses predicados procuram e garantem que a solução encontrada para o agendamento seja otimizada, respeitando as restrições de tempo e disponibilidade.

**Resultados Obtidos**

Após a execução do programa, os resultados são apresentados em três elementos principais:

1. **Agenda da Sala de Operação:** Intervalos ocupados com cirurgias e os seus códigos.
2. **Agendas dos Médicos:** Disponibilidades atualizadas após o agendamento.
3. **Tempo Final das Cirurgias:** Indica o tempo necessário para concluir todas as operações.

**Importância dos resultados:**

* A **agenda da sala** confirma a sequência e distribuição das cirurgias.
* As **agendas dos médicos** verificam se o uso do staff é eficiente.
* O **tempo final** avalia o sucesso do objetivo principal da user story.

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

Descrição gerada automaticamente

Figura 1 – Resultados pretendidos (com 5 Ids de cirurgias)

### **US 6.3.2**

O objetivo principal desta User Story é determinar até que dimensão (número de cirurgias) é viável solicitar a melhor solução para o agendamento de cirurgias em bloco operatório. Isto envolve uma análise de complexidade do problema para entender os limites práticos do sistema.

1. **Dimensão do Problema:**

* Foram testados casos de 1 a 10 cirurgias
* O número de soluções possíveis varia de 1 a 3628800

1. **Métricas Importantes:**

* **N. of Surgeries:** Número de cirurgias a agendar
* **N. of solutions:** Número de soluções possíveis
* **Best Schedule of activities (including surgeries) of the operation room:** Melhor Agenda de atividades (incluindo cirurgias) do bloco operatório
* **Final Time for the last Surgery(m):** Tempo final da última cirurgia em minutos
* **Time to generate the solution(s):** Tempo necessário para gerar a solução em segundos

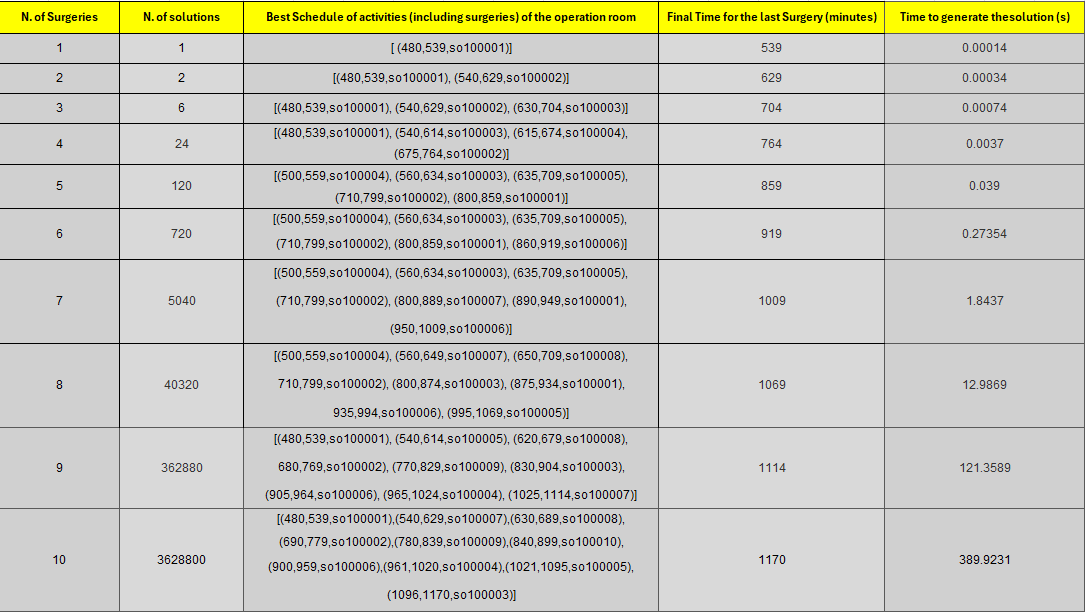


Figura 1 – tabela de complexidade

**Conclusões sobre a Complexidade com o Aumento do Número de Cirurgias**

A análise da tabela evidencia como o aumento do número de cirurgias afeta diretamente a complexidade computacional e os tempos necessários para gerar uma solução. A seguir estão as principais observações:

1. **Crescimento Exponencial do Número de Soluções**  
   O número de soluções possíveis cresce de forma fatorial com o aumento do número de cirurgias. Para um número “n” de cirurgias, o número total de combinações é “n!”, o que resulta numa complexidade O(n!). Isso significa que cada nova cirurgia adicionada multiplica significativamente o espaço de busca.
2. **Aumento do Tempo de Geração das Soluções**  
   Os tempos necessários para calcular a melhor solução aumentam drasticamente conforme cresce o número de cirurgias. Para 1 cirurgia, o tempo é praticamente insignificante (0.00014 segundos), mas para 10 cirurgias, o tempo ultrapassa 389 segundos. Isso reflete o impacto do crescimento exponencial no esforço computacional necessário.
3. **Impacto no Tempo Final da Última Cirurgia**  
   À medida que o número de cirurgias aumenta, o tempo final para a realização da última cirurgia também cresce. Este aumento está relacionado à necessidade de acomodar mais atividades na sala de operações, reforçando a importância de uma eficiente alocação de recursos.

### **US 6.3.3**

O objetivo desta tarefa é gerar um "bom" agendamento para as cirurgias em salas de operações de hospitais, priorizando o tempo de execução em detrimento da obtenção da solução absolutamente ótima. O objetivo é fornecer uma solução num intervalo de tempo razoável (por exemplo, menos de 30 segundos), utilizando heurísticas. Isso garante que o sistema seja eficiente e prático para uso em cenários reais.

Para alcançar este objetivo, foram implementadas duas heurísticas:

1. **Heurística da Disponibilidade Mais Cedo**.
2. **Heurística do Tempo Mais Ocupado**.

Os agendamentos gerados foram comparados com uma solução ótima, que avalia exaustivamente todas as permutações possíveis. A comparação foi realizada com base no tempo de execução e no tempo final da última cirurgia do agendamento.

**Descrição das Heurísticas**

1. Heurística da Disponibilidade Mais Cedo

Objetivo: Agendar cirurgias priorizando os slots de tempo mais cedo disponíveis para a sala de operações e os membros do staff necessários.

A função “availability\_all\_surgeries/3” é responsável por percorrer a lista de códigos de operação (OpCode) e identificar intervalos de tempo livres para cada cirurgia. Os passos são:

1. Identificação dos requisitos da cirurgia: Obtém os tempos necessários para preparação, cirurgia e limpeza (TPrep, TSurgery, TCleaning) e calcula o tempo total (TTotal).
2. Busca pelos intervalos disponíveis: A função “availability\_operation/5” realiza a interseção entre a disponibilidade da sala e dos membros do staff, retornando os intervalos viáveis.
3. Agendamento no primeiro intervalo livre: A cirurgia é programada no primeiro slot que atende aos requisitos, e os horários das três fases (preparação, cirurgia e limpeza) são atribuídos à sala e à agenda dos membros do staff.
4. Continuação para outras cirurgias: A função recursivamente processa todas as operações na lista.

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

Descrição gerada automaticamente

Vantagens:

* Execução rápida.
* Minimiza atrasos no início das cirurgias.

Limitações:

* Pode deixar espaços não utilizados no agendamento, reduzindo a utilização geral de recursos.

2. Heurística do Tempo Mais Ocupado

Objetivo: Agendar cirurgias de forma a maximizar a percentagem de tempo em que os membros do staff estão ocupados, reduzindo períodos de inatividade.

A função “obtain\_better\_sol/5” avalia várias permutações possíveis da lista de cirurgias e seleciona a sequência que gera o melhor agendamento. Os passos incluem:

1. Geração de Permutações: A função “permutation/2” cria todas as combinações possíveis da lista de operações.
2. Simulação de Agendamento: Para cada permutação, as cirurgias são agendadas sequencialmente, utilizando “availability\_all\_surgeries/3”.
3. Avaliação do Resultado: O tempo final do último procedimento (TFinOp) e a utilização dos recursos são calculados.
4. Seleção da Melhor Solução: A função “update\_better\_sol/4” atualiza a solução ótima se a permutação atual resultar em um tempo de término menor ou melhor ocupação.

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

Descrição gerada automaticamente

Vantagens:

* Melhora a utilização geral dos recursos.
* Reduz o tempo ocioso dos membros do staff mais ocupados.

Limitações:

* Pode atrasar algumas cirurgias se as percentagens de ocupação forem mal equilibradas.

**Implementação dos Testes para Comparação de Heurísticas**

Para validar e comparar as duas heurísticas — Heurística 1 (Disponibilidade Mais Cedo) e heurística 2 (Tempo Mais Ocupado) — foi desenvolvido um caso de teste.

Este caso simula o agendamento de cirurgias numa sala de operações num dia específico, comparando os agendamentos gerados por ambas as heurísticas e selecionando a melhor.

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

Descrição gerada automaticamente

O objetivo do código de teste é:

1. **Simular o Agendamento**:
   * Testar o agendamento de uma lista de cirurgias (OpCodeList) numa sala de operações (Room) num dia específico (Day).
2. **Aplicar as Heurísticas**:
   * **Heurística 1**: Prioriza o primeiro intervalo de tempo disponível (schedule\_all\_surgeries).
   * **Heurística 2**: Prioriza a utilização máxima dos recursos (obtain\_better\_sol).
3. **Avaliar os Resultados**:
   * Calcula o tempo de término da última cirurgia com o predicado evaluate\_schedule.
   * Compara os agendamentos gerados pelas duas heurísticas e seleciona o melhor.

**Conclusões**

As heurísticas implementadas geraram agendamentos "bons" dentro de um intervalo de tempo razoável, cumprindo os requisitos da história de utilizador 6.3.3.

1. A Heurística da Disponibilidade Mais Cedo prioriza agendamentos rápidos com atrasos mínimos, mas pode não utilizar os recursos de forma ideal.
2. A Heurística do Tempo Mais Ocupado equilibra melhor a utilização dos recursos, embora tenha um tempo de execução ligeiramente superior.

Concluindo, a Heurística 2 (Tempo Mais Ocupado) é mais vantajosa em termos de eficiência global e utilização dos recursos. No entanto, isso depende do equilíbrio entre a necessidade de eficiência e a urgência de realização das cirurgias.

Se o contexto do hospital prioriza:

* Rapidez no início das cirurgias: A Heurística 1 seria melhor.
* Eficiência geral e alta carga operacional: A Heurística 2 seria melhor.

Portanto, a escolha depende do cenário e da prioridade do hospital (por exemplo, atendimento emergencial ou otimização dos recursos).