

Relatório de Produção Individual

Sistema de Agendamento para Salão de Beleza

Guilherme G. Silvério

22/07/2025

1. Atribuição de Cargo e Tarefas

Por decisão minha, resolvi fazer o projeto final da matéria sozinho então tive e terei vários papéis diferentes para garantir não apenas a funcionalidade, mas também a qualidade e a robustez da arquitetura.

Os papéis exercidos até agora foram:

- Responsável por desenhar a arquitetura do sistema. Isso incluiu a decisão inicial de desacoplar os componentes, a escolha das tecnologias (mudando um pouco dos exemplos passados em sala de aula), e a modelagem do diagrama de classes e dos diagramas de componentes e de sequência com PlantUML para visualizar o fluxo de dados.
- Responsável pela codificação da aplicação principal (o lado ORM), utilizando Spring Boot. Inicialmente era pra eu ter usado o padrão MVC porém decidi utilizar o padrão em camadas, mas ainda assim vou implementar a camada de VIEW no projeto.
- Realizei a modelagem do banco de dados relacional, optando pelo PostgreSQL por questão de familiaridade.
- Estruturei um cronograma detalhado para garantir que todas as entregas necessárias (ODM, Middleware, UI, Testes, etc.)

2. Contribuição de Acordo com a Atribuição

Nesta primeira fase do projeto, meu foco foi estabelecer uma base sólida e funcional para a aplicação principal (realização do agendamento de serviços) para dar um foco maior na persistência dos dados que é o foco da matéria.

O que foi cumprido:

- **Definição completa da Arquitetura:** O design de um sistema de sincronização orientado a eventos foi finalizado, com a escolha das ferramentas e a definição de como os componentes se comunicam.
- **Implementação da Camada ORM:** Toda a parte referente ao banco de dados relacional (PostgreSQL) foi implementada, incluindo:
 - Entidades JPA que mapeiam o diagrama de classes.

- Repositórios para acesso aos dados.
- Serviços com a lógica de negócio principal para o agendamento.
- DTOs para a troca de dados entre as camadas.

3. Principais Dificuldades:

- **Escolha da Ferramenta de Mensageria:** A decisão inicial de usar Redis foi questionada porque ele não teria suporte de manter as mensagens na fila caso o listener estivesse offline. Isso levou a minha decisão do uso do RabbitMQ.
- Pensar na desnormalização e normalização também está sendo um desafio para o que me propus solucionar.

4. Contribuição Além do Atribuído

Como único membro do projeto, "ajudar a equipe" se traduziu em ir além do requisito básico e aprofundar em conceitos teóricos para construir uma solução mais profissional e resiliente.

- **Estudo de Padrões de Integração:** Pesquisei sobre "Enterprise Integration Patterns" para entender formalmente os papéis dos meus componentes. Identifiquei que meu middleware atua como um *Content Enricher* e *Message Translator*, o que clareou minhas ideias de como implementá-lo.
- Buscar conhecimento em como implementar mensageria no sistema também é um ponto importante a ser citado pois seria a primeira vez que eu desenvolveria e implantaria a função de fato em uma aplicação, por mais que eu já tinha conhecimento de sua arquitetura.

5. Considerações Gerais

- **O que aprendi:** Esta primeira fase foi um aprendizado imenso sobre a importância do design de arquitetura. Aprendi na prática as diferenças filosóficas entre bancos de dados SQL e NoSQL, o poder do desacoplamento através de mensageria assíncrona.
- **Trabalhos Futuros Pendentes:** Conforme o cronograma estabelecido, as próximas etapas para a finalização do projeto são:
 - Implementação da camada ODM com MongoDB.
 - Desenvolvimento completo do Middleware Integrador (com a lógica de transformação).
 - Criação da Interface Gráfica em JavaFX.

- Elaboração dos testes unitários e de integração.
 - Finalização da documentação do projeto.
- **Conclusões:** A base construída até agora é sólida e bem planejada. As decisões arquiteturais tomadas, como o uso de RabbitMQ e a estratégia de desnormalização, preparam o terreno para uma aplicação final robusta e escalável. Embora o desafio de desenvolver tudo sozinho seja grande, acredito que o trabalho esteja proporcionando uma visão completa e aprofundada nas diferentes formas que um sistema pode ter de persistir seus dados.