

Documento de Considerações Finais – Projeto de Gestão de Coworking

Equipe: Arthur Andrade de Oliveira, Lucas Borges Silva, Raphaela Tamietto Rios, Gabriel Vilhena Magri, Ian Faria Chamone, Guilherme Pereira Carneiro, Iyan Lucas Duarte Marques.

1. Introdução

Este documento apresenta as considerações finais referentes ao desenvolvimento do sistema distribuído para gestão de coworking em Belo Horizonte. Concebido para solucionar os desafios de agendamento em ambientes colaborativos, o projeto resultou na criação de uma plataforma funcional, composta por um backend responsável pela lógica de negócio, uma aplicação web e um aplicativo mobile.

Ao longo do semestre, a equipe dedicou-se a transformar os objetivos propostos em uma solução viável, superando as restrições de prazo e orçamento. O documento a seguir detalha a avaliação crítica das tecnologias e da arquitetura implementadas, uma análise do processo de desenvolvimento adotado, o panorama da gestão do projeto e, por fim, as atribuições e responsabilidades de cada membro da equipe.

2. Avaliação dos Frameworks e Tecnologias Utilizadas

A seleção tecnológica foi um pilar estratégico, guiada pela necessidade de construir uma aplicação multiplataforma, robusta e escalável, respeitando as restrições do projeto. A seguir, avaliamos as principais ferramentas utilizadas.

2.1. Backend: Java com Spring Boot

- **Pontos Positivos:** O Spring Boot forneceu um ecossistema completo e maduro, acelerando o desenvolvimento da API REST. A injeção de dependências e a vasta quantidade de módulos, como o Spring Security e o Spring Data, simplificaram a implementação de funcionalidades complexas como autenticação (RF-001) e a persistência de dados com o PostgreSQL.
- **Pontos Negativos e Desafios:** O consumo de memória da JVM foi um ponto de atenção, exigindo uma configuração cuidadosa para operar em ambientes com recursos limitados.
- **Conclusão da Escolha:** A escolha se provou acertada pela segurança e escalabilidade que o ecossistema Spring oferece, garantindo uma base sólida e confiável para os serviços essenciais do sistema.

2.2. Frontend: JavaScript com React

- **Pontos Positivos:** A arquitetura baseada em componentes do React permitiu a criação de uma interface de usuário modular e reutilizável, o que foi crucial para desenvolver a funcionalidade de agendamento (RF-002) e garantir uma experiência responsiva (RNF-001).
- **Pontos Negativos e Desafios:** O gerenciamento de estado entre múltiplos componentes tornou-se um desafio à medida que a aplicação crescia. A equipe separou em dois repositórios, e depois tivemos o retrabalho de integrar os dois em apenas um só, para fazer a entrega..
- **Conclusão da Escolha:** React foi fundamental para entregar uma interface moderna e interativa, cumprindo com os requisitos de usabilidade e responsividade do projeto.

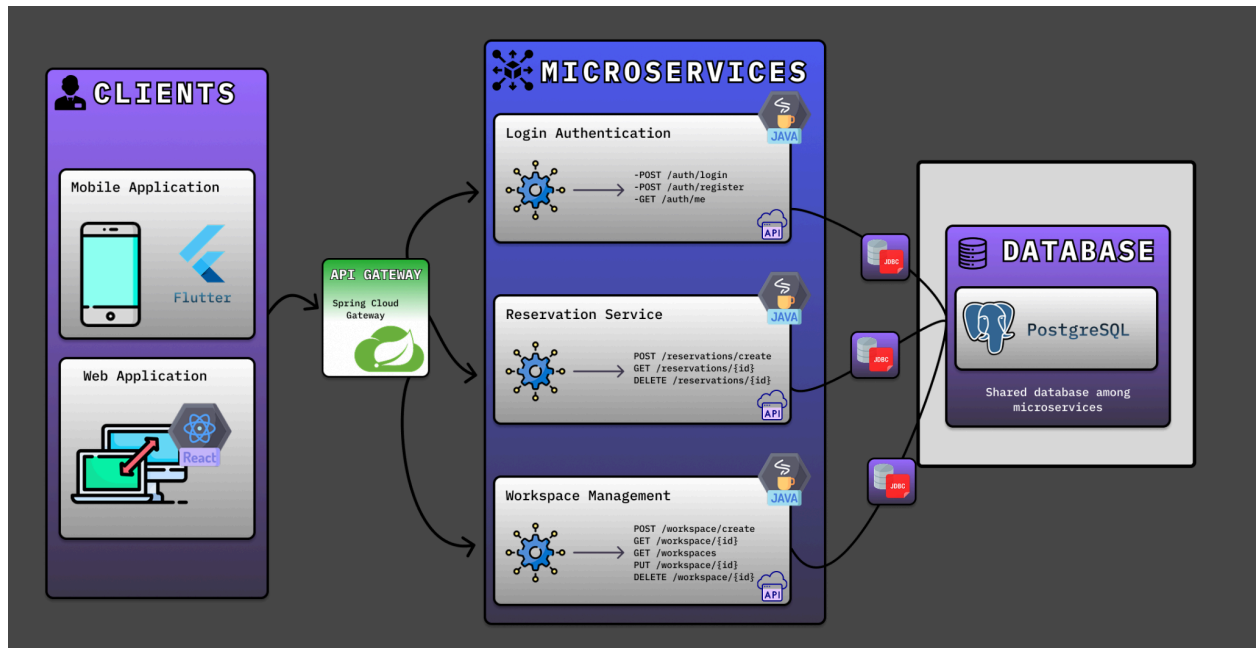
2.3. Mobile: Dart com Flutter

- **Pontos Positivos:** A principal vantagem do Flutter foi a capacidade de desenvolver para múltiplas plataformas (iOS e Android) a partir de um único código-base, otimizando o tempo da equipe e garantindo consistência visual. A funcionalidade de *Hot Reload* foi extremamente útil para agilizar o ciclo de desenvolvimento e teste da interface.
- **Pontos Negativos e Desafios:** O Flutter e a linguagem Dart, possuem um grau de dificuldade maior, comparado com o React utilizado no Frontend. E a dificuldade de instalação, e a mudança constante nas versões, impactou em alguns momentos.
- **Conclusão da Escolha:** Flutter foi uma escolha estratégica que viabilizou a entrega de um aplicativo mobile funcional dentro das restrições do projeto, garantindo uma experiência de usuário coesa com a aplicação web.

3. Análise Crítica e Proposta de Melhorias da Arquitetura

3.1. Visão Geral da Arquitetura Adotada

A arquitetura do projeto foi estruturada em três camadas distintas: um Backend monolítico desenvolvido em Spring Boot, que provê uma API REST; um Frontend web em React; e um aplicativo Mobile em Flutter. Os clientes (web e mobile) consomem a API do backend para todas as operações, e o banco de dados serve como uma camada de persistência.



3.2. Propostas de Melhorias

1. **Agendamento por Horário:** Hoje o serviço do nosso coworking, é apenas por day use, seria interessante, em novas versões construir a funcionalidade de poder reservar horário, para se tornar uma aplicação mais completa e versátil.
2. **Serviço de Logs e Auditoria:** Poderia ser construído, um módulo de relatório e auditoria, para facilitar na extração de logs e realizar análises de dados de workspaces, clientes, etc.
3. **Pipeline de CI/CD:** Implementar um pipeline de Integração e Entrega Contínua para automatizar os testes e o deploy das aplicações. Isso reduziria o risco de erros manuais e aceleraria a entrega de novas funcionalidades.

4. Análise do Processo de Desenvolvimento

O processo de desenvolvimento foi fortemente influenciado pelas restrições do projeto, principalmente o prazo de entrega e a ausência de orçamento.

- **Metodologia:** Adotamos uma abordagem simples e bem comunicativa, utilizamos discord e whatsapp para comunicação principal. A divisão clara de responsabilidades (backend, frontend, etc.) foi definida desde o início para otimizar o fluxo de trabalho.
- **O que Funcionou Bem:** A clara divisão de tarefas por especialidade permitiu que os membros focassem em suas áreas de maior conhecimento, aumentando a produtividade. O uso exclusivo de tecnologias de código aberto foi uma resposta bem-sucedida à restrição de orçamento.

5. Responsabilidades e Atribuições Detalhadas da Equipe

A tabela a seguir detalha as atribuições e o foco principal de cada membro durante o ciclo de vida do projeto.

Responsável	Atividade Principal	Detalhes das Contribuições
Arthur Andrade de Oliveira	Líder	Responsável pela coordenação geral das atividades, facilitação da comunicação, gerenciamento do cronograma e mediação da tomada de decisões técnicas.
Lucas Borges Silva	Backend e Frontend	Atuou no desenvolvimento do backend e contribuiu na construção de componentes da interface web com React.
Raphaela Tamietto Rios	Backend e Documentação	Focada na documentação de endpoints da API, modelagem do banco de dados e na elaboração da documentação técnica e funcional do projeto.
Gabriel Vilhena Magri	Frontend e Mobile	Responsável pelo desenvolvimento da aplicação web em React e contribuiu significativamente na construção da aplicação mobile em Flutter.
Ian Faria Chamone	Mobile	Focado no desenvolvimento da aplicação mobile com Flutter, implementando as interfaces e a lógica de consumo no ambiente mobile.
Guilherme Pereira Carneiro	Testing (QA)	Responsável pela estratégia e execução dos testes, garantindo a qualidade dos requisitos funcionais e não funcionais em todas as plataformas.
Iyan Lucas Duarte Marques	Interfaces (UI/UX)	Responsável pelo design das interfaces e pela experiência do usuário (UX), criando os protótipos e guiando o desenvolvimento visual do frontend e mobile.