

Número do ADR: 003

Título: Escolha da Arquitetura de Microservices e Princípios da Clean Architecture

Data: 2024-07-16

Contexto

No início do projeto de desenvolvimento do sistema a **Health&Med**, uma startup inovadora no setor de saúde. Ao receber um aporte financeiro, a empresa decidiu desenvolver um sistema proprietário para melhorar a qualidade dos serviços, garantir a segurança dos dados dos pacientes e reduzir custos operacionais.

Nesse cenário, foi desenvolvido um DDD desempenhou um papel crucial, uma vez que através do mesmo foi possível:

1. Identificação de Domínios:

- a) Agendamento de Consultas: Responsável por gerenciar os horários disponíveis e as marcações de consultas e realização de Consultas Online: Envolve a comunicação entre médicos e pacientes por meio de videochamadas.
- b) Gerenciamento de agenda: gerenciar os horários disponíveis e as marcações de consultas e realização de Consultas Online: Envolve a comunicação entre médicos e pacientes por meio de videochamadas.
- c) Gerenciamento de cadastros: Gerenciamento das especificidades de cliente e medico e gerenciamento de autenticação e visualizações diferente entre ambos.

2. Contextos Delimitados:

Cada domínio foi tratado como um contexto delimitado, com suas próprias regras de negócio e terminologia específica.

Essa delimitação permitiu que a equipe foque em resolver problemas específicos dentro de cada domínio.

3. Comunicação entre Domínios:

A arquitetura de microservices foi escolhida para lidar com os diferentes domínios.

Cada microservice representa um contexto delimitado, permitindo escalabilidade independente e manutenção facilitada.

Em resumo, o DDD ajudou a Health&Med a entender profundamente o negócio, a delimitar contextos e a criar uma arquitetura que refletisse esses domínios de forma clara e eficiente. As escolhas arquiteturais, como a adoção de microservices e os princípios da Clean Architecture, foram motivadas por essa análise cuidadosa do domínio.

Decisão

Optamos por adotar a arquitetura de microservices, seguindo os princípios da Clean Architecture. Essa decisão foi tomada com base nas seguintes considerações:

1. Separação de Responsabilidades:

A Clean Architecture enfatiza a separação clara entre as camadas do sistema: domínio, aplicação e infraestrutura.

Os Microservices também promovem a separação de funcionalidades em serviços independentes.

Essa combinação permite que cada Microservice tenha sua própria lógica de negócios e seja facilmente substituível ou escalável.

2. Independência Tecnológica:

A Clean Architecture garante que o núcleo do sistema (domínio) não dependa de detalhes de infraestrutura.

Os Microservices podem ser desenvolvidos em diferentes linguagens, frameworks e tecnologias, desde que sigam os contratos definidos pelas interfaces.

3. Escalabilidade e Manutenção:

Os Microservices podem ser escalados individualmente, permitindo que áreas específicas do sistema lidem com diferentes cargas de trabalho.

A manutenção é facilitada, pois cada Microservice é autônomo e pode ser atualizado sem afetar outros.

Justificativa

A escolha da arquitetura de Microservices com base nos princípios da Clean Architecture foi motivada pelas seguintes razões:

1. **Flexibilidade e Evolução:** A separação de camadas e a modularização dos serviços permitem que o sistema evolua de forma mais ágil, adicionando ou substituindo Microservices conforme necessário.
2. **Testabilidade:** Cada Microservice pode ser testado independentemente, garantindo a qualidade do software e praticidade na execução dos testes de forma independente.
3. **Futuro-Proofing:** A independência tecnológica e a clareza das responsabilidades tornam o sistema mais resiliente a mudanças futuras.

Alternativas Consideradas Foram consideradas as seguintes alternativas:

Outras opções:

Monolítico com Camadas: Essa abordagem poderia dificultar a escalabilidade e a manutenção à medida que o sistema crescesse.

Arquitetura de Microsserviços sem Clean Architecture: Poderíamos ter optado por uma arquitetura de microsserviços sem seguir os princípios da Clean Architecture. No entanto, isso poderia levar a inconsistências na estrutura e dificultar a manutenção.

Arquitetura de Microsserviços com hexagonal: não é naturalmente orientada à escalabilidade horizontal e em projetos pequenos ou com requisitos simples, a arquitetura hexagonal pode ser excessiva.