**Relatório da Solução: Modernização de Sistema para Microserviços em Java**

1. Introdução: Este relatório descreve a solução proposta para a modernização do sistema legado mainframe por meio da implementação de microserviços em Java, utilizando o framework Spring Boot. A arquitetura adotada visa melhorar a flexibilidade, escalabilidade e manutenibilidade do sistema.

2. Arquitetura de Microserviços: A solução consiste em quatro microserviços principais, cada um focado em uma área específica de funcionalidade:

* **Microserviço - Pagamento:** Gerência ação de pagamento, comunica com outros sistemas para gerar débitos ou boletos para o cliente.
* **\* Microserviço - Renegociação:** Responsável por operações de acordo e geração de contratos.
* **Microserviço – Negativação:** Gera ações para birôs de crédito como Serasa, SPC.
* **Microserviço – Cobrança:** Responsável por gerar a cobrança interna e externa.

(\*) Implementação realizada como exemplo para simulação na entrevista.

Cada microserviço é independente, possui seu próprio banco de dados e expõe endpoints HTTP para interação.

# 3. Comunicação entre Microserviços:

* **API Gateway (APIG):** Utilizado para direcionar as requisições dos clientes para os microserviços apropriados.
* **Kafka:** Facilita a comunicação assíncrona entre os microserviços e expõem dados a fim de auditoria e comunicação para os sistemas legados

# 4. Integração com Canais Externos:

* **Birôs de Crédito:** Notificação para comunicar a negativação da dívida ou a quitação.
* **Canais de Cobrança Externa:** Notificação para reporte empresas externas de cobrança .

# 5. Canais de Acesso ao Cliente:

* **Web App (Web):** Consumidor da API Gateway para acesso via aplicação web.
* **Mobile App (Mobile):** Consumidor da API Gateway para acesso via aplicação móvel.

# 6. Gerenciador do Cluster:

* Utilização do Istio-ingress para balanceamento interno dos pods no cluster EKS
* Utilização do Istio-engress para acesso aos Canais externos

# 7. Tecnologias Utilizadas:

• Linguagem: Java 17

• Framework: Spring Boot WebFlux (Reativa)

• Banco de Dados: NoSQL

# 8. Conclusão:

Realizei a implementação de uma poc colocando algumas ideias para o fluxo de negociação para expressar de forma simplicista uma simulação de acordo e contrato.

Publicação:  
github: <https://github.com/andersoncsprime/entrevista.git>