Documento de Arquitetura e Proposta Técnica

Case: Cadastro de Clientes e Geração de Crédito Resiliente

Autor: Arthur Thomé

Cargo: Desenvolvedor Backend Pl (c# .NET)

Data: Outubro/2025

Sumário

1. Introdução e Arquitetura

O objetivo deste documento é descrever de forma prática a arquitetura do sistema responsável pelo fluxo de cadastro de clientes, análise de crédito e emissão de cartões.
A ideia é garantir resiliência, rastreabilidade e autonomia entre serviços, mantendo o controle completo do status de cada cliente em todo o processo.

A solução adota uma arquitetura de microsserviços orientada a eventos, seguindo o padrão SAGA Coreografada.

Nesse modelo, não existe um orquestrador central, cada microsserviço executa sua parte, publica um evento e aciona o próximo serviço do fluxo.
Isso traz baixo acoplamento e alta escalabilidade, além de facilitar a manutenção e evolução do sistema.

A mensageria será feita via RabbitMQ, que ficará responsável pelas filas, roteamento de mensagens, mecanismos de retry e gerenciamento das Dead Letter Queues.

Tecnologias principais:
- .NET 8.0 para o desenvolvimento dos microsserviços
- RabbitMQ como barramento de mensagens

2. Microsserviços Envolvidos

Clientes: Recebe o cadastro via API REST, persiste os dados, inicia o fluxo de eventos e consome os eventos de finalização para atualizar o status do cliente.

Proposta: Consome o evento de cliente cadastrado, executa a análise de crédito e publica o evento de aprovação ou reprovação.

Cartão: Consome o evento de proposta aprovada, realiza a emissão e publica o evento final de sucesso ou falha.

3. Fluxo de Eventos e Comunicação (SAGA Coreografada)

O processo completo (**Cadastro → Proposta → Cartão)** é todo assíncrono e baseado em mensagens publicadas e consumidas entre os microsserviços.

3.1. Eventos Principais

ClienteCadastradoEvent

Publicado por: Clientes
Consumido por: Proposta
Finalidade: Inicia a análise de crédito.

PropostaAprovadaEvent

Publicado por: Proposta
Consumido por: Cartão
Finalidade: Indica que a emissão do cartão pode prosseguir

PropstaReprovadaEvent

Publicada por: Proposta
Consumida por: Clientes
Finalidade: Finaliza a SAGA com status ‘Reprovado’.

CartaoEmitidoEvent

Publicado por: Cartão
Consumido por: Clientes
Finalidade: Finaliza a SAGA com status ‘Aprovado’.

EmissaoCartaoFalhouEvent

Publicado por: Cartão
Consumido por: Clientes
Finalidade: Finaliza a SAGA com falha técnica ou lógica.

4. Detalhamento do Fluxo

1. 4.1. Início da SAGA – MS Clientes

- Recebe o cadastro via POST /clientes, salva os dados e publica o evento ClienteCadastradoEvent.

2. 4.2. Análise de Crédito – MS Proposta

- Consome o evento ClienteCadastradoEvent, executa as regras de negócio e publica o evento de aprovação ou reprovação.

3. 4.3. Emissão de Cartão – MS Cartão

- Consome o evento PropostaAprovadaEvent, tenta emitir o cartão e publica o resultado final.

4. 4.4. Sinalização Final – MS Clientes

- Consome todos os eventos de encerramento e atualiza o status final no banco (Aprovado, Reprovado, Falha na Emissão).

5. Resiliência e Tratamento de Falhas

Como o sistema é totalmente assíncrono, aplicamos dois mecanismos principais: Retry e Dead Letter Queue (DLQ).

5.1. Retry

Caso um evento gere erro técnico (ex: timeout, falha de banco, exceção não tratada), a mensagem não é descartada.
O sistema faz novas tentativas de processamento com backoff exponencial (exemplo: 1min → 5min → 15min).

5.2. Dead Letter Queue (DLQ)

Se todas as tentativas falharem, a mensagem é movida para a DLQ correspondente.
Essas filas permitem auditoria e garantem que nenhuma mensagem seja perdida.