DESAFIO TÉCNICO - SISTEMA DE ASSINATURAS

Contexto

Você foi contratado para desenvolver um sistema de gestão de assinaturas para um serviço de streaming. Os usuários podem assinar planos mensais e a cobrança deve ocorrer automaticamente.

Requisitos

1. Cadastro e Gerenciamento de Assinaturas

- Criar uma API para **cadastrar usuários e criar uma assinatura**.
- Um usuário pode ter **apenas uma assinatura ativa** por vez.
- A assinatura contém:
  {
  "id":"uuid",
  "usuarioId":"uuid",
  "plano":"PREMIUM",
  "dataInicio":"2025-03-10",
  "dataExpiracao":"2025-04-10",
  "status":"ATIVA"
  }
- O plano pode ser:
  - BASICO - R$ 19,90/mês
  - PREMIUM - R$ 39,90/mês
  - FAMILIA - R$ 59,90/mês

2. Renovação Automática de Assinaturas

- Implementar um **agendador** que renove assinaturas automaticamente no dia do vencimento.
- Caso a renovação falhe (exemplo: erro de pagamento), a assinatura deve ser suspensa após **três tentativas**.

3. Cancelamento de Assinatura

- Criar um endpoint para **cancelar a assinatura** de um usuário.

- Se cancelada antes da expiração, o usuário pode continuar usando o serviço até o fim do ciclo.

Regras de Negócio

✔ Um usuário pode assinar apenas **um plano por vez**.
✔ O sistema deve **gerenciar pagamentos e falhas**, suspendendo assinaturas após 3 tentativas de renovação falhadas.
✔ A renovação ocorre automaticamente **exatamente no dia de vencimento**.

Diferenciais (Não obrigatórios)

- Uso de **Spring Boot** para estruturar a aplicação.
- Persistência com **PostgreSQL ou MongoDB**.
- Implementação de **eventos assíncronos (RabbitMQ/Kafka)** para processar pagamentos.
- Uso de **cache (Redis)** para otimizar consultas de assinaturas ativas.
- Implementação de **testes automatizados** com JUnit + Mockito.
- Deploy em **Docker** ou integração com **AWS/GCP**.

Critérios de Avaliação

✅ **Código bem estruturado** - boas práticas, SOLID, padrões de projeto
✅ **Tratamento de concorrência** - evitar problemas na renovação das assinaturas
✅ **Desempenho** - uso eficiente de banco, cache e concorrência
✅ **Testabilidade** - cobertura de testes unitários e de integração
✅ **Criatividade e melhorias extras**