

Objetivo

Desenvolver um sistema simples que permita **gerenciar propostas de seguro e efetuar sua contratação**, utilizando:

- Arquitetura Hexagonal (Ports & Adapters)
- Abordagem baseada em **microserviços (APIs, Filas ou BD)**
- Banco de dados relacional ou em dados mocados
- Boas práticas de Clean Code, DDD, SOLID, design partners e testes unitários

Contexto do Sistema

Você está construindo parte de uma plataforma de seguros. O sistema precisa permitir que usuários **criem propostas de seguro**, consultem seu status, e eventualmente **efetuem a contratação da proposta**.

Divida a aplicação em **dois microserviços** principais:

1. PropostaService

Responsável por:

- Criar proposta de seguro
- Listar propostas
- Alterar status da proposta (**Em Análise**, **Aprovada**, **Rejeitada**)
- Expor API REST

2. ContratacaoService

Responsável por:

- Contratar uma proposta (somente se **Aprovada**)
- Armazenar informações da contratação (ID da proposta, data de contratação)
- Comunicar-se com o **PropostaService** para verificar status da proposta
- Expor API REST

Requisitos Técnicos

- Linguagem: **C# com .NET 8 ou superior**
- Banco de dados: **SQL Server, PostgreSQL ou outro a sua escolha**
- Arquitetura: **Hexagonal**
- Comunicação entre microserviços: **HTTP REST, banco de dados ou mensageria (bonus)**
- Padrões de projeto: Clean Architecture, DDD (camadas bem definidas)
- Contêineres: Docker (bonus)
- Testes automatizados: Unitários e/ou de integração

Entregáveis

- Repositório no GitHub com instruções de build e execução (README)
- Dockerfile (se possível)
- Banco de dados versionado com scripts ou migrations
- Testes automatizados
- Diagrama simples da arquitetura (bonus)