

Análise de Migração COBOL para .NET

Sistema SUSEP Circular 360 - Apuração de Prêmios

Caixa Seguradora

Versão 1.0.0
24/10/2025

CONFIDENCIAL

1. Resumo Executivo

1.1 Contexto do Projeto

Este documento apresenta a análise completa da migração do sistema COBOL RG1866B (SUSEP Circular 360 - Sistema de Apuração de Prêmios) para uma arquitetura moderna baseada em .NET 9 e React 18.

O sistema legado RG1866B é responsável por:

- Processamento de 687 itens de dados de prêmios de seguros
- Consulta a 26+ tabelas/views DB2
- Geração de arquivos regulatórios PREMIT.TXT e PREMCED.TXT
- Cálculos complexos de cosseguro conforme regulamentação SUSEP

O programa possui aproximadamente 5.000 linhas de código COBOL e processa transações críticas para conformidade regulatória.

1.2 Objetivos da Migração

1. **Modernização Tecnológica:** Substituir COBOL por stack moderno (.NET 9 + React 18)
2. **Conformidade Byte-for-Byte:** Manter compatibilidade exata com saída COBOL
3. **Arquitetura Limpa:** Implementar Clean Architecture com 3 camadas
4. **Interface Web:** Criar dashboard React para consultas e relatórios
5. **Qualidade:** Atingir 90%+ cobertura de testes

1.3 Investimento

Item	Valor (BRL)
Desenvolvimento (313 FP x R\$ 750)	R\$ 234.750,00
Infraestrutura Azure (12 meses)	R\$ 18.000,00
Licenças e ferramentas	R\$ 15.000,00
Contingência (15%)	R\$ 40.162,50
TOTAL	R\$ 307.912,50

2. Análise do Sistema COBOL Legado

2.1 Visão Geral do Programa RG1866B

O programa RG1866B foi desenvolvido em COBOL e executa processamento batch para apuração de prêmios conforme SUSEP Circular 360.

Métricas Principais:

- Linhas de código: ~5.000 LOC
- Itens de dados: 687 data items
- Tabelas acessadas: 26+ views/tables
- Arquivos de saída: 2 (PREMIT.TXT, PREMCED.TXT)
- Complexidade ciclomática: Alta (cálculos de cosseguro)

2.2 Principais Componentes

Seção de Inicialização (R0100-R0400):

- Abertura de arquivos
- Inicialização de variáveis
- Validação de parâmetros de execução
- Setup de cursores DB2

Processamento Principal (R0500-R0600):

- Lê registros da view V0PREMIOS
- Aplica regras de negócio
- Calcula prêmios e comissões
- Acumula totalizadores

Cálculos de Cosseguro (R3000-R5500):

- Distribuição proporcional de prêmios entre cosseguradoras
- Cálculo de participações conforme tabela GE399
- Validação de limites regulatórios SUSEP
- Geração de totalizadores por cosseguradora

3. Arquitetura de Migração

3.1 Clean Architecture - 3 Camadas

A arquitetura target segue os princípios de Clean Architecture com separação clara de responsabilidades:

Camada 1: CaixaSeguradora.Api (Presentation)

- ASP.NET Core 9 Web API
- Controllers para endpoints REST e SOAP
- Validação de requisições
- Swagger/OpenAPI documentation

Camada 2: CaixaSeguradora.Core (Domain)

- Entidades de domínio (Premium, Policy, Cossurance)
- Interfaces de repositório e serviço
- Regras de negócio (cálculos financeiros)
- Zero dependências externas

Camada 3: CaixaSeguradora.Infrastructure (Data Access)

- Entity Framework Core 9
- Repositórios concretos
- Mapeamento para 26 views DB2
- FixedWidthFormatter para arquivos COBOL

3.2 Stack Tecnológico

Componente	Tecnologia
Backend Framework	.NET 9.0
Linguagem	C# 13.0
ORM	Entity Framework Core 9.0
API	ASP.NET Core Web API
Database	SQLite (dev), DB2 (prod)
Frontend	React 18.3 + TypeScript 5.5
Build Tool	Vite 5.3
Styling	TailwindCSS 3.4
Testing Backend	xUnit 2.9 + FluentAssertions
Testing Frontend	Vitest 2.0 + Playwright

4. Análise de Pontos de Função

4.1 Metodologia IFPUG 4.3.1

A análise de pontos de função foi realizada seguindo IFPUG (International Function Point Users Group) versão 4.3.1.

4.2 Breakdown por Tipo de Função

Tipo	Nome	Complexidade	Peso	Total
EO	Gerar PREMIT.TXT	Alta	7	7
EO	Gerar PREMCED.TXT	Alta	7	7
EO	API Relatórios	Média	5	30
EI	Consulta Prêmios	Média	4	8
EI	Filtros Dashboard	Baixa	3	12
EQ	Query Builder	Alta	6	12
ILF	V0PREMIOS	Alta	15	15
ILF	V0APOLICE	Alta	15	15
ILF	GE399 (Cosseguro)	Média	10	10
EIF	Demais 23 tabelas	Média	7	161
			UFP	277
			VAF	1.13
			AFP	313

Observação: O VAF de 1.13 reflete alta complexidade de processamento (cálculos cosseguro), performance crítica (< 2s resposta), conformidade byte-for-byte com COBOL, e reusabilidade de componentes (90%+).

5. Cronograma do Projeto

5.1 Visão Geral - 14 Semanas

Fase	Atividades	Duração
Fase 0	Análise e planejamento	1 semana
Fase 1	Setup projeto, CI/CD, DB	2 semanas
Fase 2	Modelos domínio, testes base	2 semanas
Fase 3	Lógica negócio, cálculos	2 semanas
Fase 4	API REST/SOAP, integração	1 semana
Fase 5	Frontend React, dashboard	2 semanas
Fase 6	Homologação, documentação	4 semanas
TOTAL		14 semanas

5.2 Milestones Principais

1. **M1 (Semana 1):** Aprovação do projeto e orçamento
2. **M2 (Semana 3):** Ambiente de desenvolvimento configurado
3. **M3 (Semana 5):** Testes de comparação COBOL passando
4. **M4 (Semana 7):** API backend completa
5. **M5 (Semana 9):** Dashboard frontend funcional
6. **M6 (Semana 10):** Início da homologação
7. **M7 (Semana 13):** Aprovação SUSEP
8. **M8 (Semana 14):** Go-live em produção

6. Apêndices

6.1 Glossário

AFP - Adjusted Function Points (Pontos de função ajustados)

COBOL - Common Business-Oriented Language

DB2 - Sistema de gerenciamento de banco de dados IBM

EF Core - Entity Framework Core (ORM da Microsoft)

IFPUG - International Function Point Users Group

PREMIT - Arquivo de premissões conforme SUSEP

PREMCED - Arquivo de prêmios cedidos (cosseguro)

SUSEP - Superintendência de Seguros Privados

UFP - Unadjusted Function Points

VAF - Value Adjustment Factor

6.2 Referências

1. SUSEP Circular SUSEP 360/2007 - Normas para apuração de prêmios
2. IFPUG Function Point Counting Practices Manual, Release 4.3.1
3. Microsoft .NET 9 Documentation
4. Clean Architecture - Robert C. Martin
5. Domain-Driven Design - Eric Evans

DOCUMENTO CONFIDENCIAL

Caixa Seguradora S.A. - Todos os direitos reservados