
RELATÓRIO DE ANÁLISE COCOMO II

Projeto: coreFrontend

Tipo de Análise	INTEGRATED
Caminho do Projeto	/Users/eusener/Projetos/TRASSE/coreFrontend
Data de Geração	16/10/2025 17:36:34

RESUMO EXECUTIVO

Este relatório apresenta uma análise completa do projeto **coreFrontend** utilizando a metodologia COCOMO II integrada com métricas de repositório Git. O projeto possui **80,289 KLOC** (mil linhas de código) e apresenta complexidade **Média**.

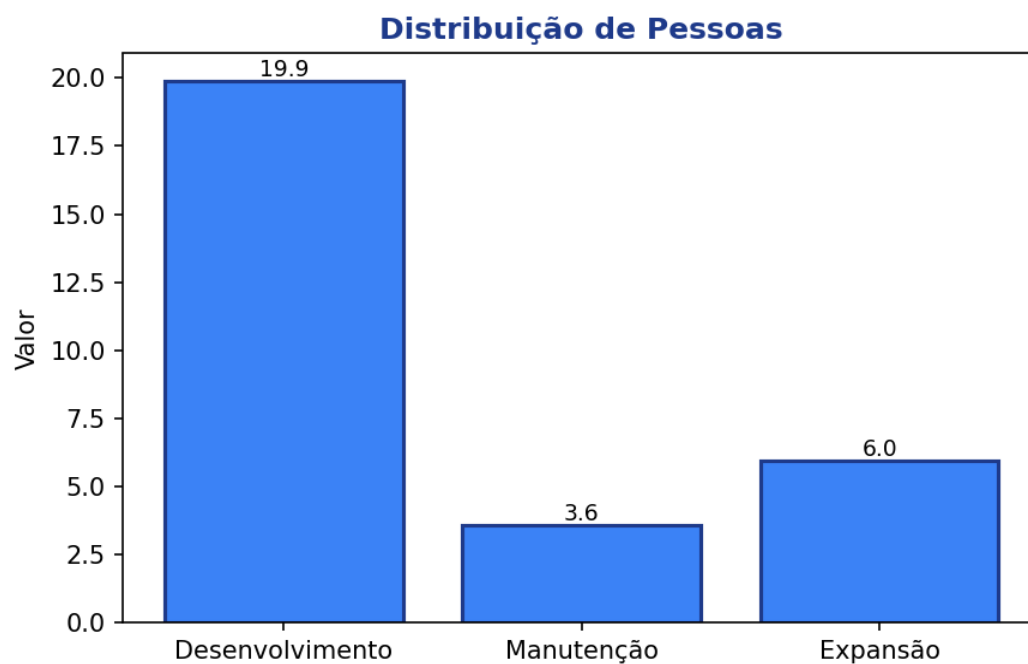
A estimativa de esforço é de **407,70 pessoa-meses**, com duração prevista de **20,49 meses** e necessidade de **19,9 desenvolvedores**. O custo estimado para o projeto é de **R\$ 2.446.173,63**.

O repositório Git contém **180 commits** de **4 autores**, com idade de **225 dias**. O score de produtividade dos desenvolvedores é de **85,24**.



ANÁLISE COCOMO II

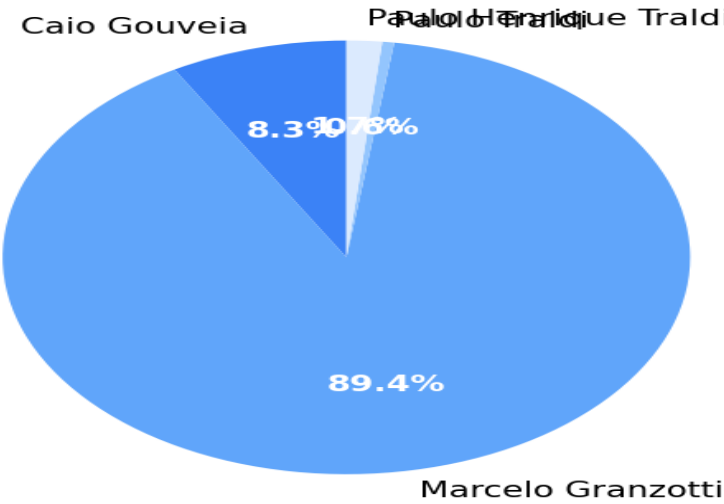
Linhas de Código (KLOC)	80,289 KLOC
Esforço	407,70 pessoa-meses
Tempo de Desenvolvimento	20,49 meses
Pessoas Necessárias	19,90 desenvolvedores
Pessoas para Manutenção	3,58 desenvolvedores
Pessoas para Expansão	5,97 desenvolvedores
Produtividade	196,93 linhas/pessoa-mês
Custo Estimado (BRL)	R\$ 2.446.173,63
Nível de Complexidade	Média



ANÁLISE DO REPOSITÓRIO GIT

Total de Commits	180
Total de Autores	4
Total de Inserções	170.665
Total de Deleções	59.388
Média de Mudanças por Commit	1.278,07
Média de Arquivos por Commit	8,69
Commits por Dia	0,80
Primeiro Commit	19/02/2025 16:13
Último Commit	03/10/2025 08:56
Idade do Repositório	225 dias

Distribuição de Commits por Autor



ANÁLISE DE SEGURANÇA

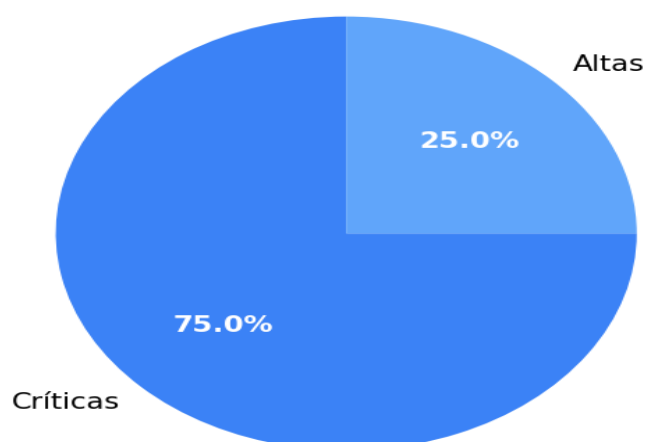
Esta seção apresenta os resultados da análise de segurança realizada com Semgrep, uma ferramenta de análise estática que identifica vulnerabilidades, bad practices e problemas de segurança no código-fonte.

Score de Segurança

64.2/100

Total de Descobertas	4
Críticas	3
Altas	1
Médias	0
Baixas	0
Informativas	0
Problemas de Segurança	4
Best Practices	0
Performance	0
Arquivos Escaneados	577
Tempo de Scan	29.81s

Distribuição de Descobertas por Severidade



Top 10 Arquivos Mais Vulneráveis

...SSE/coreFrontend/src/components/common/CustomButtonMap.ts	1
...etos/TRASSE/coreFrontend/src/types/indicadoresEsgTypes.ts	1
...ASSE/coreFrontend/src/types/rotogramaSimplificadoTypes.ts	1
...reFrontend/src/validations/rotogramaSimplificado/index.ts	1

Principais Descobertas (Críticas/Altas)

1. [CRITICAL] javascript.browser.security.insecure-document-method.insecure-document-method

Arquivo: ...ojetos/TRASSE/coreFrontend/src/components/common/CustomButtonMap.ts:32

Mensagem: User controlled data in methods like `innerHTML`, `outerHTML` or `document.write` is an anti-pattern that can lead to XSS vulnerabilities

2. [CRITICAL]

generic.secrets.security.detected-aws-access-key-id-value.detected-aws-access-key-id-value

Arquivo: ...sener/Projetos/TRASSE/coreFrontend/src/types/indicadoresEsgTypes.ts:127

Mensagem: AWS Access Key ID Value detected. This is a sensitive credential and should not be hardcoded here. Instead, read this value from an environment var...

3. [CRITICAL]

generic.secrets.security.detected-aws-access-key-id-value.detected-aws-access-key-id-value

Arquivo: ...rojetos/TRASSE/coreFrontend/src/types/rotogramaSimplificadoTypes.ts:31

Mensagem: AWS Access Key ID Value detected. This is a sensitive credential and should not be hardcoded here. Instead, read this value from an environment var...

4. [HIGH] javascript.ajv.security.audit.ajv-allerrors-true.ajv-allerrors-true

Arquivo: .../TRASSE/coreFrontend/src/validations/rotogramaSimplificado/index.ts:9

Mensagem: By setting `allErrors: true` in `Ajv` library, all error objects will be allocated without limit. This allows the attacker to produce a huge number...

Recomendações de Segurança:

- Priorize a correção de problemas CRÍTICOS e ALTOS imediatamente
- Revise os arquivos mais vulneráveis identificados acima
- Implemente testes de segurança automatizados no pipeline CI/CD
- Realize revisões de código com foco em segurança
- Mantenha dependências atualizadas para evitar vulnerabilidades conhecidas
- Considere treinamento em segurança para a equipe de desenvolvimento

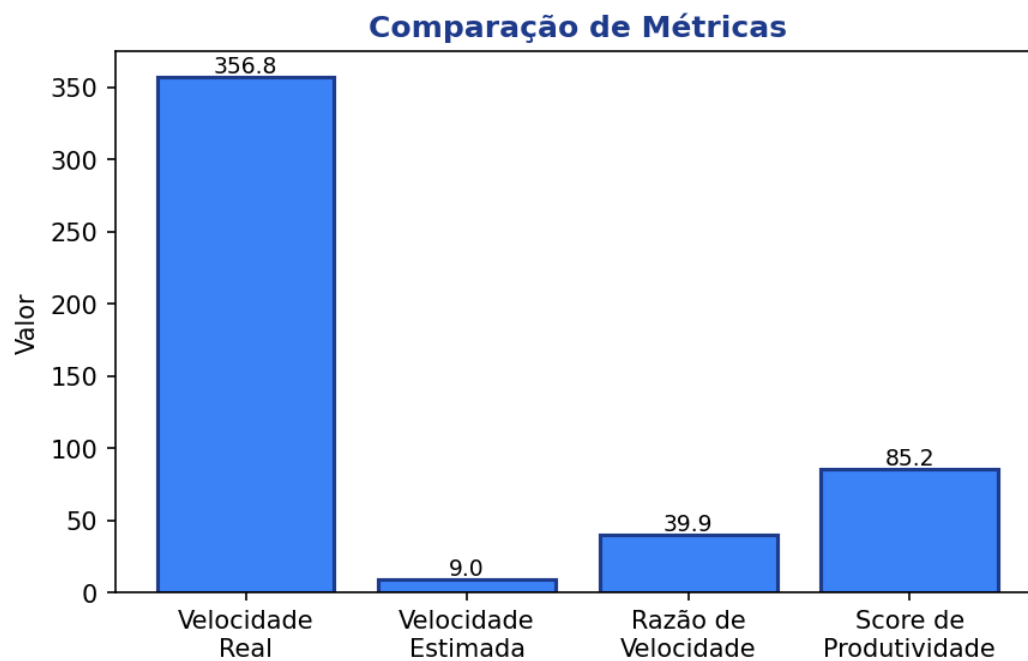
ANÁLISE INTEGRADA

KLOC (COCOMO)	80,289 KLOC
Esforço (COCOMO)	407,70 pessoa-meses
Commits por Mês	24,00
Linhas por Commit	446,05
Commits Necessários para Reconstruir	180
Velocidade Real	356,84
Velocidade Estimada	8,95
Razão de Velocidade	39,86
Eficiência de Commits	3.490,02%
% de Mudança por Commit	1,59%
Score de Produtividade	85,24

Análise de Velocidade:

A razão de velocidade de **39,86** indica que o projeto está sendo desenvolvido **39,86x** mais rápido que o estimado pelo modelo COCOMO II. Isso pode indicar:

- Alta produtividade da equipe
- Uso eficiente de ferramentas e frameworks
- Possível subestimação da complexidade inicial
- Reutilização de código existente



INSIGHTS DE INTELIGÊNCIA ARTIFICIAL

Esta seção apresenta análises avançadas geradas por Inteligência Artificial, fornecendo insights sobre o valor do código, métricas de mercado e recomendações estratégicas baseadas nas métricas coletadas do projeto.

Avaliação Geral:

O código do projeto coreFrontend apresenta uma complexidade média, com um volume considerável de linhas e uma boa taxa de comentários. No entanto, com um número elevado de pessoas necessárias para a manutenção e expansão, é preciso focar na eficiência e qualidade do código.

Pontos Fortes:

- Boa taxa de comentários (5,21%) que facilita a manutenção.
- O projeto possui uma quantidade significativa de código em TypeScript, que aumenta a segurança e a legibilidade.
- A produtividade média é razoável (196,93 KLOC/Pessoa-Mês).

Áreas de Melhoria:

- Reduzir o número de pessoas necessárias para a manutenção e expansão para aumentar a eficiência.
- Melhorar a eficiência dos commits e a velocidade de desenvolvimento.
- Monitorar e aprimorar a qualidade de código para reduzir o número de findings críticos.

Valor de Mercado Estimado: Considerando o custo estimado de aproximadamente R\$ 2.446.173,63, o projeto poderia ser avaliado em um mercado de software com potencial de monetização dependendo da sua aplicação e adoção.

MÉTRICAS DE MERCADO

Comparação com a Indústria:

O projeto apresenta métricas próximas às médias do setor, mas a necessidade de quase 20 pessoas para desenvolvimento é acima da média, o que pode ser um indicativo de ineficiência.

Maturidade do Projeto:

Em desenvolvimento, com uma estrutura de código sólida mas com necessidade de melhorias significativas.

Qualidade do Código:

A qualidade do código é média, com vulnerabilidades que precisam de atenção e uma taxa de comentários que, embora boa, ainda pode ser melhorada.

Velocidade de Desenvolvimento:

A velocidade de desenvolvimento é baixa quando comparada à estimativa de velocidade, indicando possíveis gargalos no fluxo de trabalho.

Análise de Custo-Benefício:

O custo-benefício é questionável com a atual equipe, sugerindo que uma reavaliação da estrutura da equipe e dos processos de desenvolvimento possa ser necessária.

INDICADORES CHAVE

ROI Estimado	O ROI pode ser considerado médio a alto, dependendo das estratégias de monetização.
Time to Market	O tempo até o lançamento é estimado em cerca de 20 meses, o que pode ser um desafio.
Risco Técnico	MÉDIO, devido ao número elevado de vulnerabilidades críticas que podem impactar a segurança.
Escalabilidade	O projeto apresenta potencial de escalabilidade, porém requer refinamento nas práticas de desenvolvimento.
Sustentabilidade	A sustentabilidade a longo prazo será desafiada sem melhorias significativas nas práticas de desenvolvimento.

RECOMENDAÇÕES ESTRATÉGICAS

Curto Prazo (1-3 meses):

- Revisar e otimizar a estrutura da equipe para reduzir a quantidade de pessoas necessárias para manutenção e expansão.
- Realizar uma análise de código focada para identificar e corrigir as vulnerabilidades críticas.
- Implementar práticas de integração contínua para melhorar a eficiência dos commits.

Médio Prazo (3-6 meses):

- Treinar a equipe em boas práticas de desenvolvimento e segurança.
- Focar na melhoria da documentação e dos comentários no código.
- Estabelecer métricas de desempenho para monitorar a produtividade da equipe.

Longo Prazo (6+ meses):

- Investir em automação de testes e CI/CD para aumentar a eficiência do desenvolvimento.
- Desenvolver um plano de escalabilidade para suportar o crescimento do projeto.

-
- Explorar novas oportunidades de monetização e desenvolvimento de features.

OPORTUNIDADES

Monetização:

- Desenvolver um modelo de assinatura para acesso a funcionalidades premium.
- Oferecer serviços de personalização do software.
- Explorar parcerias com outras empresas para integração de serviços.

Expansão:

- Expandir para mercados latino-americanos com necessidades semelhantes.
- Desenvolver funcionalidades adicionais que atendam a um público mais amplo.
- Considerar a internacionalização do software com suporte a múltiplas línguas.

Otimização:

- Melhorar a eficiência do código através de revisões e refatorações.
- Aumentar a cobertura de testes para reduzir o número de bugs em produção.
- Implementar práticas de DevOps para otimizar o fluxo de trabalho.

ANÁLISE DE SEGURANÇA

Avaliação Geral:

O estado de segurança do projeto é preocupante, com um número significativo de findings críticos que precisam ser abordados imediatamente.

Nível de Risco: ALTO, uma vez que a presença de vulnerabilidades críticas como XSS e hard-coded credentials pode levar a sérias consequências para a segurança do sistema.

Vulnerabilidades Críticas:

- CWE-798: Uso de credenciais hardcoded.
- CWE-79: Falha de neutralização inadequada de entrada durante a geração de página da web.
- CWE-400: Consumo de recursos não controlado.

Prioridades de Correção:

-
1. Corrigir as credenciais hardcoded imediatamente.
 2. Mitigar a vulnerabilidade XSS.
 3. Implementar controles para evitar consumo não controlado de recursos.

Impacto no Negócio:

As vulnerabilidades podem comprometer a integridade e a reputação do negócio, resultando em perdas financeiras e de confiança por parte dos usuários.

Compliance:

É fundamental garantir que o projeto esteja em conformidade com as regulamentações de segurança de dados, como a LGPD no Brasil.

Recomendações Urgentes:

- Implementar uma auditoria de segurança completa.
- Treinar a equipe sobre práticas de segurança de desenvolvimento.
- Estabelecer um ciclo regular de revisão de segurança e testes de penetração.

CONCLUSÃO

O projeto **coreFrontend** demonstra indicadores positivos de desenvolvimento:

Pontos Fortes:

- Score de produtividade elevado (85,24)
- Eficiência de commits de 3.490,02%
- Velocidade de desenvolvimento 39,86x acima do estimado
- Complexidade classificada como Média
- Score de Segurança: 64.2/100 (bom)
- Total de Descobertas de Segurança: 4

Estimativas Finais:

- Custo Total: R\$ 2.446.173,63
- Duração: 20,49 meses
- Equipe: 19,9 desenvolvedores
- Manutenção: 3,6 desenvolvedores

Este relatório fornece uma base sólida para tomada de decisões estratégicas sobre alocação de recursos, planejamento de releases, estimativas de custos e segurança do projeto.