PROJETO A3

REFATORAÇÃO DE CÓDIGO · CLEAN CODE · PADRÕES DE PROJETO

UNIVERSAL CONVERTER

INTEGRANTES:

- LEONARDO MELO PELICANO
- CARLOS EDUARDO DOS SANTOS JUNIOR
- PEDRO HENRIQUE HÔRTENCIO DE OLIVEIRA

INTRODUÇÃO

O que é o UniversalConverter?

- Um sistema que realiza conversões de unidades:
  - Temperatura (Celsius, Fahrenheit, Kelvin)
  - Comprimento (metro, centímetro, quilômetro)
  - Peso (quilograma, grama, libra)

Objetivo do projeto A3:

- Criar um código legado simples, porém funcional.
- Refatorar completamente aplicando boas práticas.
- Transformar um script desorganizado em um sistema estruturado.

Por que isso importa?

- Simula um cenário real de manutenção de software.

O CÓDIGO ORIGINAL

Resumo:
Funcionava, mas era frágil, confuso e pouco profissional.

O CÓDIGO LEGADO FAZIA O QUÊ?

- Recebia um valor numérico.
- Recebia a unidade de origem e destino.
- Usava vários IFs para decidir qual cálculo aplicar.
- Imprimia o resultado direto no terminal.

PROBLEMAS DO CÓDIGO ORIGINAL:

- Arquivo único e difícil de entender.
- Lógica repetida para diferentes unidades.
- Condicionais gigantes.
- Difícil de testar e evoluir.
- Sem estrutura para expansão.

PROBLEMAS TÉCNICOS IDENTIFICADOS

- FALTA DE MODULARIZAÇÃO E RESPONSABILIDADES MISTURADAS.
- DUPLICAÇÃO DE CÁLCULOS E REGRAS DE CONVERSÃO.
- AUSÊNCIA TOTAL DE TESTES.
- ALTO ACOPLAMENTO (TUDO DEPENDIA DE TUDO).
- CÓDIGO DIFÍCIL DE MANTER.

IMPACTO REAL

Qualquer pequena mudança quebrava outra parte.

Adicionar novas unidades era praticamente impossível.

IMPLEMENTAMOS

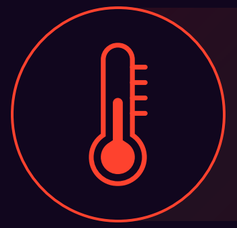
REFACTOR: ARQUITETURA FINAL

STRATEGY	Cada conversor tem sua estratégia.
FACTORY	Seleciona automaticamente o conversor correto.
SERVIÇO CENTRAL	Orquestra as conversões.
CLI	Facilita rodar no terminal.

BENEFÍCIOS:

- Código limpo, modular, organizado e expansível.

TESTES E QUALIDADE



Temperatura (C ↔ F ↔ K)

Comprimento (m ↔ cm ↔ km)

Peso (kg ↔ g ↔ lb)

Casos inválidos (unidade desconhecida, valor inválido)

Por que isso é importante?

- Garante que o comportamento original foi mantido.
- Evita regressões.
- Dá segurança para evoluir o sistema.
- Mostra maturidade no desenvolvimento.

RESULTADOS E CONCLUSÃO

Resultados alcançados:

- Código totalmente modular e bem estruturado.
- Arquitetura profissional (Strategy + Factory).
- Testes cobrindo cenários reais.
- Sistema pronto para receber novas unidades.
- Documentação completa e clara.

Conclusão do grupo:

- Refatorar melhora drasticamente qualidade do software.
- Padrões de projeto tornam o código profissional e escalável.
- Testes garantem confiança no que foi entregue.
- O projeto demonstra aprendizado real em engenharia de software.

OBRIGADO