

# **APLICAÇÃO DE TÉCNICAS DE REFACTORING NO DESENVOLVIMENTO DE FRAMEWORKS ORIENTADOS A OBJETOS**



**U F M G**

**FI**Tec  
Inovações Tecnológicas

**Palestrante: Isaac Babsky Neto**  
ATAN Ciência da Informação

**Orientador: Renato Cardoso Mesquita**  
Departamento de Engenharia Elétrica – UFMG

**Co-orientadora: Ana Liddy Cenni de Castro Magalhães**  
FITec Inovações Tecnológicas

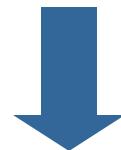
- 1. Objetivo / Justificativa**
- 
- 3. Contextualização**
- 
- 5. Conceitos**
- 
- 7. Desenvolvimento**
- 
- 9. Resultados Obtidos**
- 
- 11. Resultados Futuros**

- **Objetivo Geral:**

- Elaborar um processo de desenvolvimento de **frameworks orientados a objetos** que utilize **refactoring** para alcançar um modelo de melhoria contínua de aspectos relacionados à qualidade interna dos softwares derivados.

- **Justificativa:**

- A **interação** entre o **usuário do framework** e o próprio **framework** deve ser projetada para garantir um esforço mínimo de utilização.



- Visando **melhorar essa interação**, esse projeto buscará, pela aplicação de **técnicas de refactoring**, aprimorar o desenvolvimento de **frameworks** por meio de transformações em seu desenho e código.

- **Produto resultante:**

- Um **modelo de desenvolvimento de frameworks orientados a objetos** que utilize:
  - técnicas de *refactoring*
  - padrões de codificação
  - processos de revisão de código
  - métricas de qualidade
  - testes

- **Relevância:**

- Ao **aprimorar o desenvolvimento** de frameworks orientados a objetos será possível tornar as **organizações** proprietárias desses *frameworks mais competitivas*:
  - permitirá um aumento de **produtividade**
  - garantirá ainda mais a **qualidade final das aplicações** que foram desenvolvidas sobre esses *frameworks*
  - possibilitará uma maior **satisfação de seus clientes** e novas **oportunidades de negócio**

- **Refactoring:**

- (Verbo): **Reestruturar** um software aplicando mudanças em seu desenho, sem alterar o comportamento externo ou observável.
- (Substantivo): Uma mudança feita na estrutura interna de um software para torná-lo **mais fácil e barato de modificar e mais simples de entender**, sem alterar seu comportamento externo ou observável.



Oficina Mecânica  
de Motos  
do João.  
Sou mecânico,  
conserto motos  
e cobro barato



Oficina Mecânica  
de Motos  
do João.  
Conserto motos  
e cobro barato



Oficina Mecânica  
de Motos  
do João.  
Cobro barato



Oficina Mecânica  
de Motos  
do João

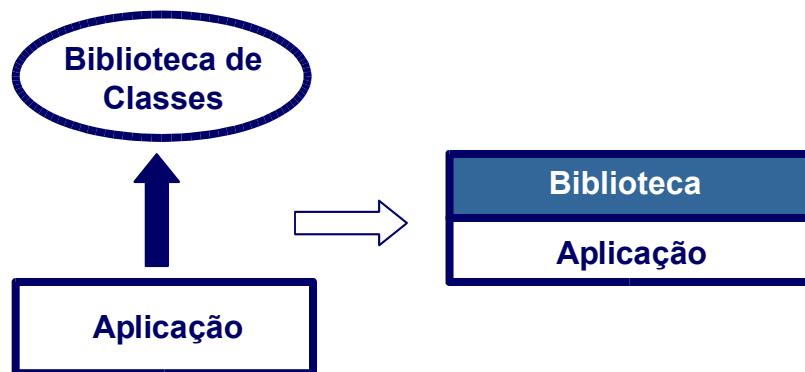


Oficina do João

## ■ **Framework Orientado a Objetos:**

### Orientação a Objetos

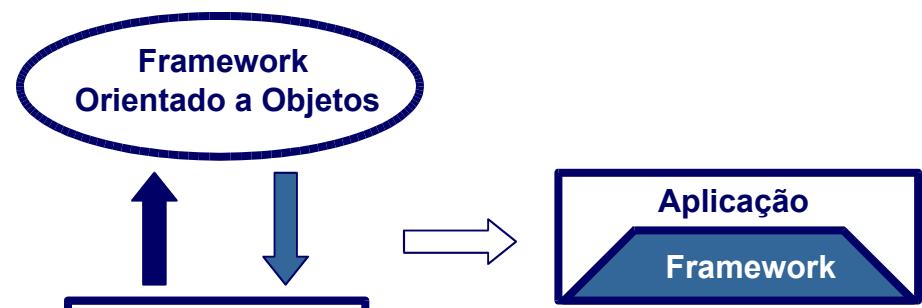
**Reutilização de Código:** Integração de dados e operações em classes (comumente reutilizadas por meio de bibliotecas).



### Framework Orientado a Objetos

**Reutilização de Código e Desenho:**

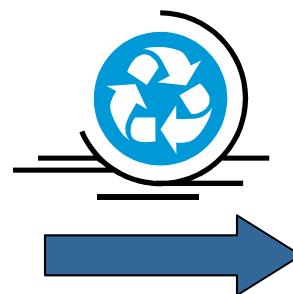
Arquitetura principal para uma aplicação ser construída, sendo responsável pelo controle do fluxo dessa aplicação (comumente reutilizado através de herança).



1. Seleção de dois *frameworks* a serem utilizados
2. Seleção e estudo de métricas de qualidade interna e externa definidas para software (normas ISO/IEC 9126 e 14598)
3. Elaboração de mecanismos para aplicação dessas métricas
4. Experimentação criteriosa de técnicas de *refactoring*
5. Introdução do uso de padrões de codificação
6. Identificação de um conjunto de regras a serem seguidas para aplicação de técnicas de *refactoring* durante o desenvolvimento de *frameworks* orientados a objetos
7. Sistematização do acompanhamento do desenvolvimento e elaboração preliminar do processo/modelo
8. Geração de produtos de software de apoio, se aplicável
9. Elaboração final do processo/modelo

# RESULTADOS OBTIDOS

- Aplicação de *Refactoring* Contínuo e bem estruturado em alguns projetos da ATAN (“Dividir para Conquistar”)
  - Estudo das Técnicas de *Refactoring*
    - ➔ Definição de Etapas de *Refactoring*
    - ➔ Proposta inicial de um processo/modelo
    - ∴ Organização, Limpeza e Aprimoramento
- Solução para o “eterno problema da pia”
  - ➔ Manutenção preventiva e melhoria contínua do processo
  - ➔ “Não esperar quebrar para consertar”



## Modelo testado em equipe inexperiente (em treinamento):

- Ganhos pessoais do desenvolvedor:
  - Melhor entendimento do sistema  
→ Aumento da confiança e da produtividade
  - Maior comprometimento quanto à qualidade  
→ Maior conscientização em relação à melhoria continua
  - Melhor interação entre os membros  
→ Aumento da comunicação
- Ganhos técnicos para a equipe de desenvolvimento:
  - Produtos mais organizados e padronizados  
→ Códigos mais enxutos, mais simples e mais claros
  - Diminuição do tempo de manutenção  
→ Ganho de tempo para a melhoria do sistema

- Comprovação dos resultados obtidos por meio de testes que utilizem métricas de qualidade
- Aprimoramento do modelo inicial
- Aplicação do modelo em equipes mais experientes
- Adaptação do modelo para equipes responsáveis pelo desenvolvimento dos dois *frameworks* selecionados
- Aumento da qualidade dos *frameworks* e da produtividade das equipes de desenvolvimento
- Definição de um modelo consistente que possa ser aplicado a uma equipe qualquer de desenvolvimento de *frameworks* orientados a objetos
- Desenvolvimento de ferramentas para auxiliar na aplicação do modelo

# Dúvidas ou sugestões ?

## Obrigado!



[isaac@atan.com.br](mailto:isaac@atan.com.br)

**(31) 3261-5079**

