

# Um Método para a Refatoração de Software Baseado em Frameworks de Domínio

---

VÍCTOR PEDROSO AMBIEL BARROS

SIMONE NASSER MATOS

DEPARTAMENTO ACADÊMICO DE INFORMÁTICA

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

# Organização da Apresentação

---

- Refatoração de Software.
- Frameworks.
- Método de Refatoração Proposto.
- Resultados.
- Dificuldades.
- Trabalhos Futuros.

# Refatoração de Software

---



Refatoração

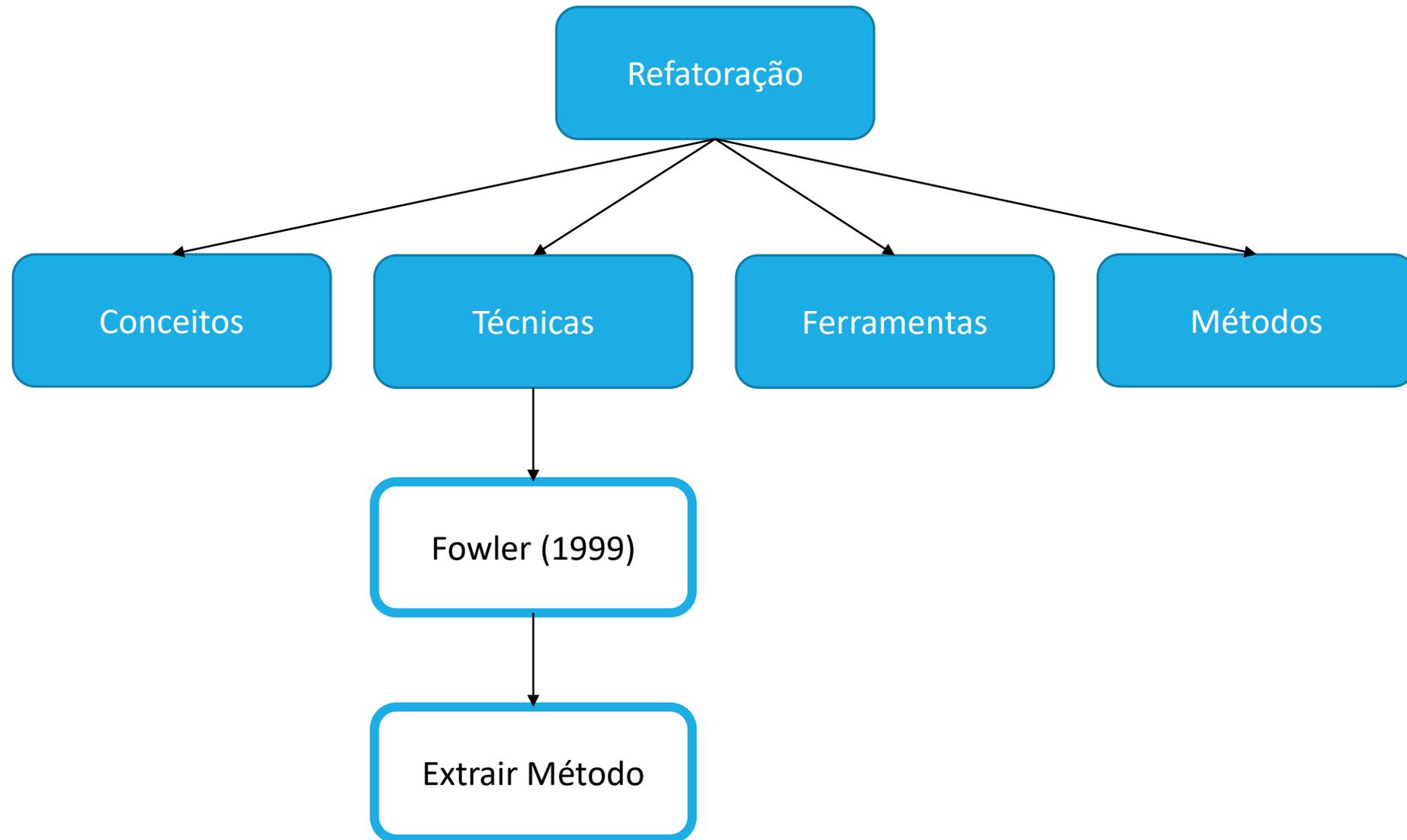
```
graph TD; A[Refatoração] --> B[Conceitos]; A --> C[Técnicas]; A --> D[Ferramentas]; A --> E[Métodos];
```

Conceitos

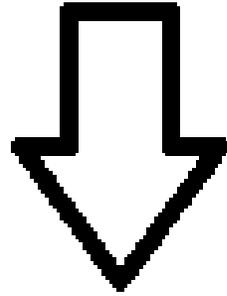
Técnicas

Ferramentas

Métodos



```
public void apresentar(){  
    preco = produtoA + produtoB;  
    System.out.println(preco);  
}
```



```
public void apresentar(){  
    preco = calcularPreco();  
    System.out.println(preco);  
}
```

```
public float calcularPreco(){  
    preco = produtoA + produtoB;  
    return preco;  
}
```

Refatoração

Conceitos

Técnicas

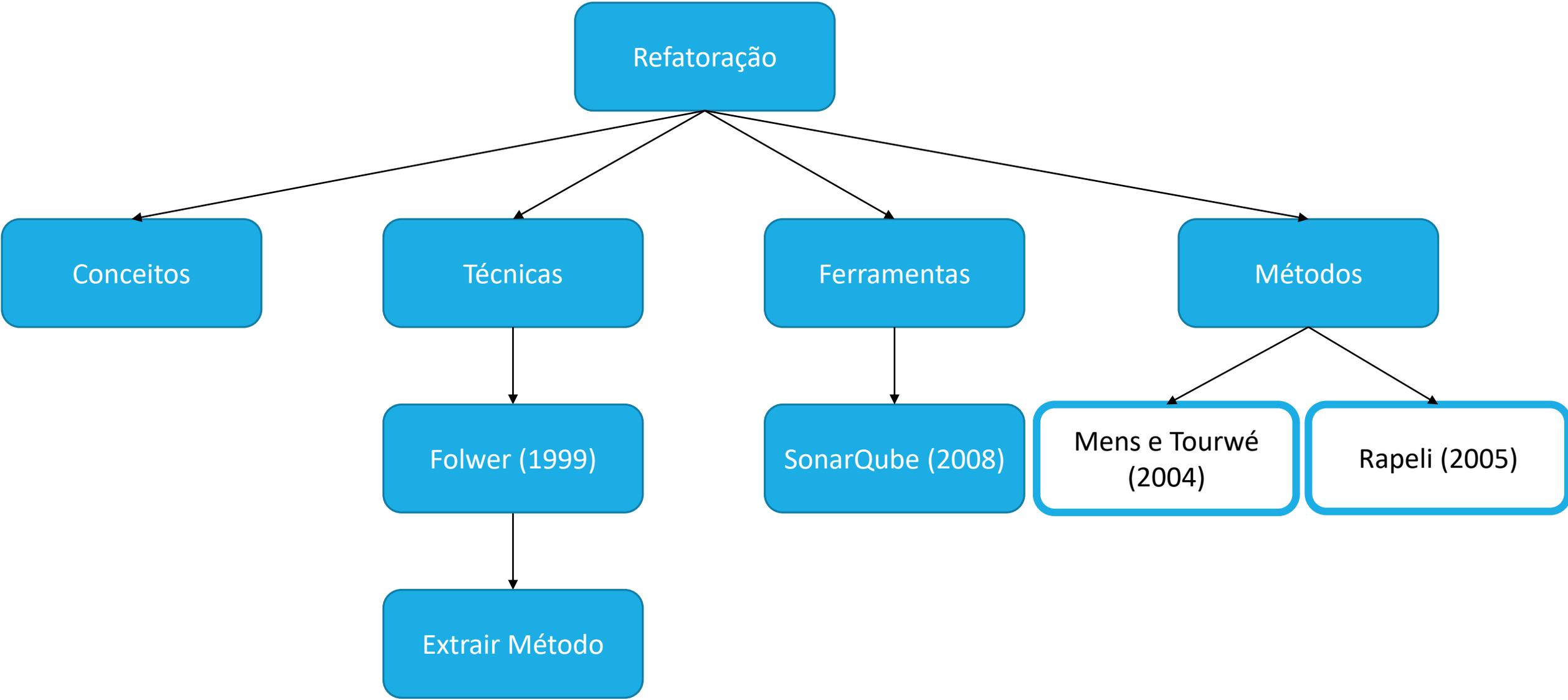
Ferramentas

Métodos

Folwer (1999)

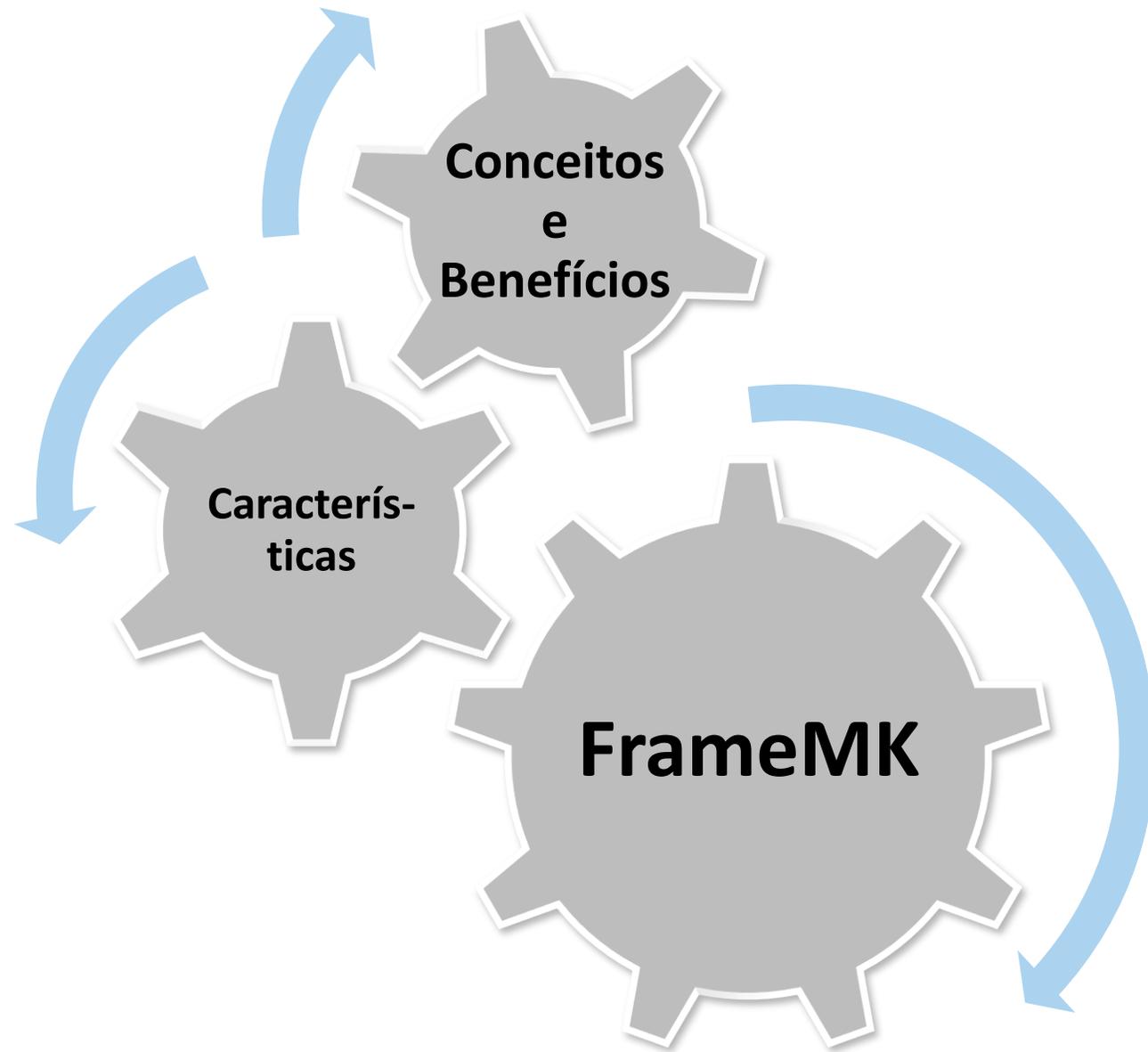
SonarQube (2008)

Extrair Método



# Frameworks

---

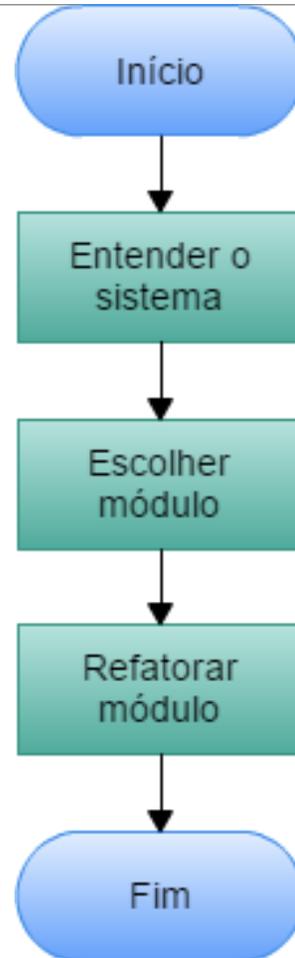


Por que um método  
específico para  
Frameworks de Domínio?

---

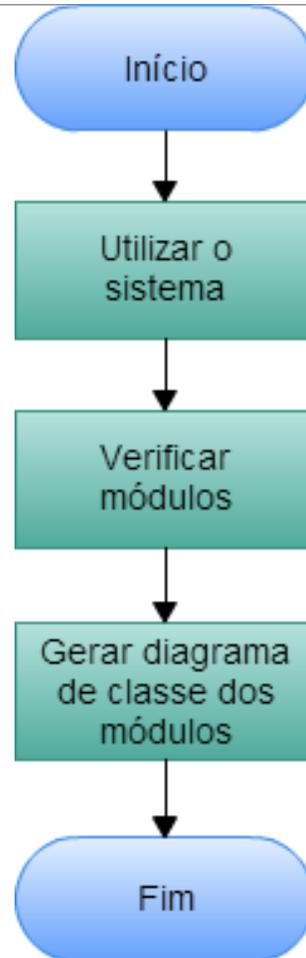
# Método Proposto

---

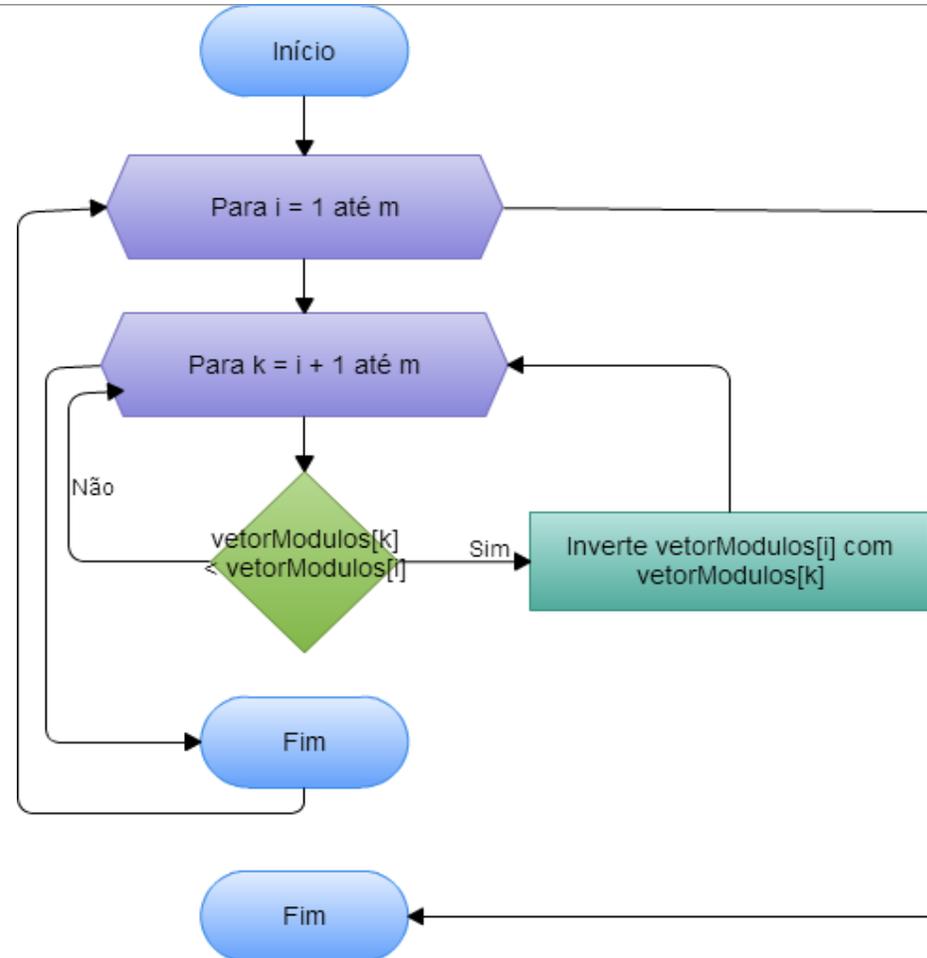


# Entender o Sistema

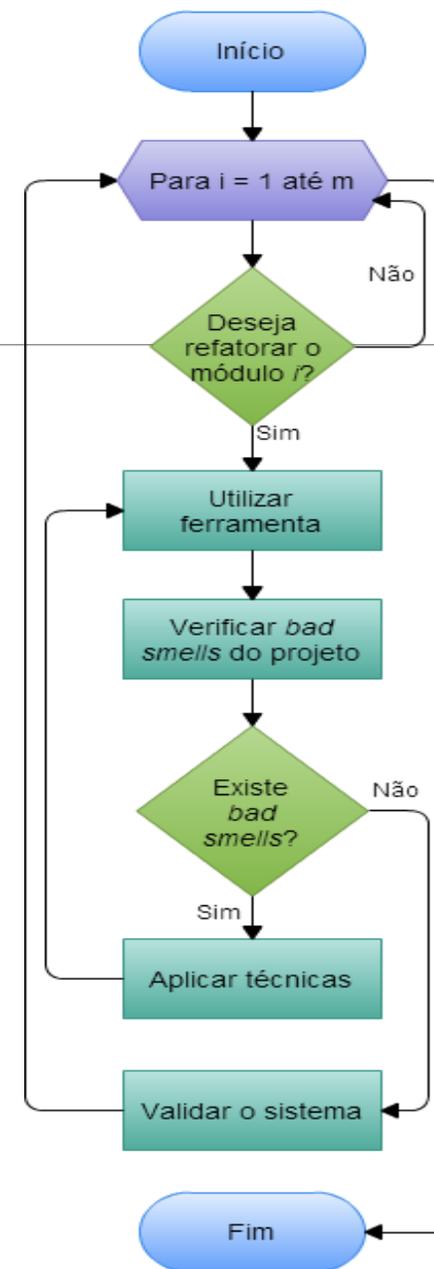
---



# Escolher Módulo



# Refatorar Módulo



# Resultados

---

- 30 classes modificadas no módulo *app*.
- 23 classes modificadas no módulo *Persistence* e *BusinessRule*.
- 5 classes modificadas no módulo *services*.
- Removido 61% dos *bad smells* do módulo *app*.
- Aumento da classificação no *SQALE* de B para A.

# Resultados

---

- 295 refatorações aplicadas:
  - Metapadrão Unification.
  - Técnicas de Fowler.
  - Refatorações a partir do SonarQube.

	Antes	Depois
Quantidade de Classes	37	39
Quantidade de bad smells	477	182
Complexidade	531	236
Duplicações	39,2%	35,8%
Linhas de código	2797	2921
Quantidade de funções	144	151
SQALE rating	B	A

\* Estatísticas referentes ao módulo *app*

# Dificuldades na Aplicação do Método

---

- **Entender o Sistema:** gerar o diagrama de classe do FrameMK, que contém 344 classes.
- **Escolher Módulo:** não houve grandes dificuldades. Algoritmo implementado foi o *Insertion Sort*.
- **Refatorar Módulo:** maior dificuldade em comparação com as outras etapas. Conhecer bem todas as técnicas de refatoração.

# Trabalhos Relacionados

---

- Semelhanças com o método de Mens e Tourwé (2004).
- Semelhanças com o método de Rapeli (2005).
- Diferenças entre os métodos.

<b>Características</b>	<b>Rapeli (2005)</b>	<b>Mens e Tourwé (2004)</b>	<b>Método Proposto</b>
<b>Compreender a funcionalidade do sistema.</b>	X	X	X
<b>Gerar diagrama de classe.</b>	X		X
<b>Aplicar padrões de projeto de forma específica.</b>	X		X
<b>Testar sistema refatorado.</b>	X	X	
<b>Refatorar para qualquer linguagem.</b>		X	
<b>Analisar código para refatorar.</b>	X	X	X
<b>Identificar tipo de refatoração a serem aplicadas.</b>		X	X
<b>Utilizar ferramentas automatizadas para análise do código.</b>			X
<b>Aplicar Inversão de Controle</b>			X
<b>Aplicar Metapadrões</b>			X

# Conclusão

---

- Criado um novo método de refatoração.
- Guia de Metapadrões e Inversão de Controle (Barros, 2015).
- Foi possível a aplicação do método proposto.
- Resultados obtidos com a aplicação do método.
- Comparado com outros métodos presentes na literatura.

# Trabalhos Futuros

---

- A identificação de novas características na construção de frameworks de domínio que devem ser levados em consideração no processo de refatoração.
- A aplicação de métricas para avaliar quantitativamente a contribuição do método na refatoração de frameworks.
- Aplicar o método proposto em outros estudos de caso ou módulos.
- Automatizar o processo de detecção de metapadrões em código fonte.
- Refinar o modelo de classes do FrameMK de modo a facilitar a engenharia reversa.
- Definir as etapas que devem ser realizadas na fase de validação do framework.

# Referências

---

FOWLER, Martin. **Refactoring**: Improving the Design of Existing Code. Boston: Addison-wesley Professional, 1999. 464 p.

**FrameMK**. Disponível em: <<http://gpes.pg.utfpr.edu.br/framek/>>. Acesso em: 5 set. 2016.

GPSI. Grupo de Pesquisa em Sistemas de Informação: GPSI. Disponível em: <<http://gpes.pg.utfpr.edu.br/gpes/>>. Acesso em: 5 set. 2016.

MENS, Tom. TOURWÉ, Tom. A Survey of Software Refactoring. **IEEE Transactions on Software Engineering**, v. 30, n. 2, 2004.

M. Fayad, D. C. Schmidt, and R. E. Johnson. **Building application frameworks: object-oriented foundations of framework design**. 1999.

# Referências

---

**RAPELI, Leide R. Refatoração de sistemas Java utilizando padrões de projeto: um estudo de caso.** 2005. 127 f. Dissertação (Mestrado, Universidade Federal de São Carlos. São Paulo, 2005).

**SQALE.** Disponível em: <<http://www.sqale.org/wp-content/uploads/2016/08/SQALE-Method-EN-V1-1.pdf>>

**SONARQUBE 5.1.1.** Disponível em: <<http://www.sonarqube.org/>>. Acesso em 5 set. 2016.

# Obrigado!

---



[victorambiel@outlook.com](mailto:victorambiel@outlook.com)



[facebook.com/victorambiel](https://facebook.com/victorambiel)



(42) 9128-9407



[linkedin.com/in/victorambiel](https://linkedin.com/in/victorambiel)



Laboratório de Estudos do PPGCC

## Perguntas?