

# Refatoração Baseada em Padrões de Projeto Usando Agentes

Thyago Henrique Pacher, Simone Nasser Matos

Departamento Acadêmico de Informática

Universidade Tecnológica Federal do Paraná (UTFPR)

Av. Monteiro Lobato, s/n – KM 04, CEP 84016-210 – Ponta Grossa – PR –  
Brasil

[thyago.pacher@gmail.com](mailto:thyago.pacher@gmail.com), [snasser@utfpr.edu.br](mailto:snasser@utfpr.edu.br)

# O que é refatorar?

- Melhorar (FOWLER, 1999), (KERIEVSKY, 2009), (MARTIN, 2009), TRIFU (2004), (MALL, 2018):
  - Manutenção;
  - Legibilidade;
  - Flexibilidade.
- Pode ser feita com:
  - Catálogo de técnicas de Fowler (1999);
  - Catálogo de padrões de projeto Gamma (1995).

# O que são padrões de projeto?

- Reuso de uma experiência já desenvolvida por outra pessoa sobre um mesmo problema (FREEMAN *et al.*, 2009).

# O que são Agentes?

- Sistema projetado para realizar tarefas em nome do ser humano (COPPIN, 2015).
- Autônomo ou semiautônomo.
- Pode ter inteligência para tomar decisões.
- Reage de acordo com seu ambiente.
- Arquitetura: BDI (*Belief – Desire – Intention*)

# Arquitetura para agente - BDI

- BDI é composto por (FAGUNDES, 2007):
  - Estados mentais tais como crenças, desejos e intenções
  - Raciocínio pois pode ter deliberação verificando os desejos possíveis de serem aplicados.

# Qual a objetivo geral do trabalho?

- Criar uma abordagem fundamentada em agentes que possa detectar e inserir padrões de projeto em código-fonte OO (Orientado a Objetos) usando interação com usuário e métodos da literatura que identificam a aplicação de padrões de projeto.

# Mapeamento sistemático

- Método:
  - Kitchenham e Charters (2007)
- Aplicação
  - Período: 2008 a 2019
  - Repositórios: 7
  - Questões

# Mapeamento sistemático - contexto

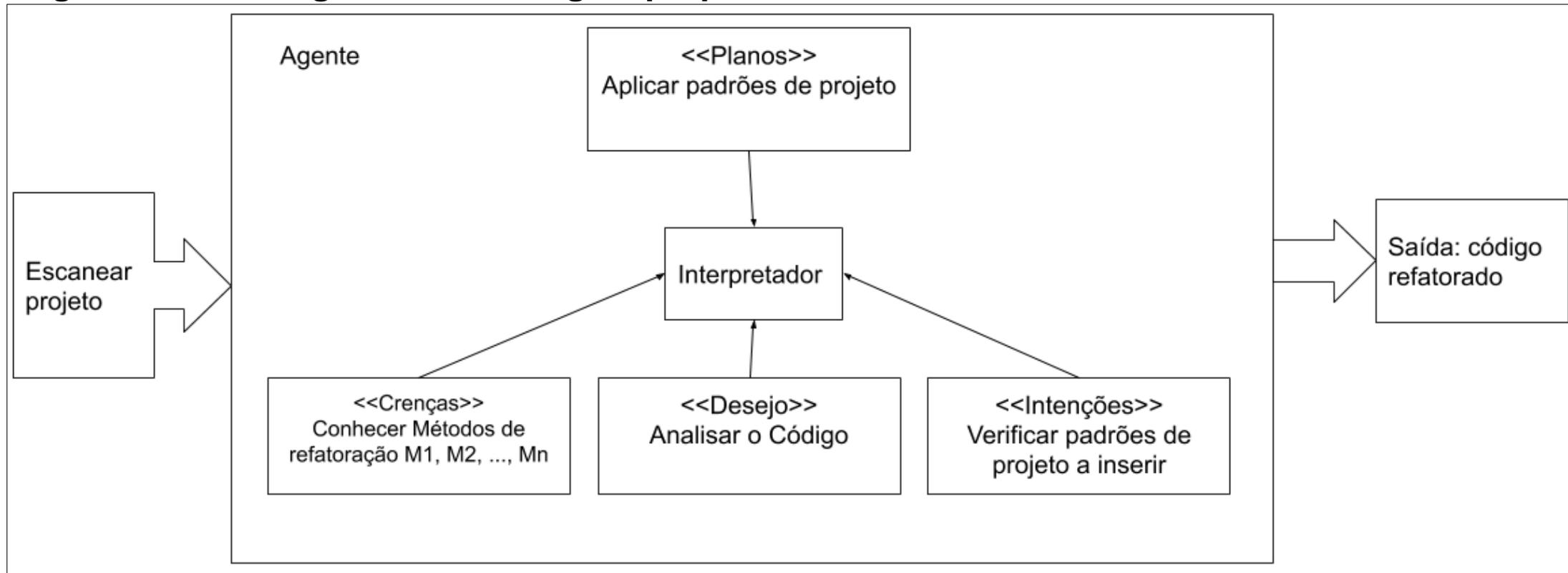
- Encontrados 14 trabalhos na área.
- Início dos estudos em 2008.
- Estudo final em 2016.
- Conclusão do mapeamento: existem abordagens que trabalham com refatoração e agentes, aplicadas somente às técnicas de refatoração.

# Requisitos básicos para funcionamento da abordagem

- Identificação de métodos da literatura para detectar e inserir de padrões de projeto
- Criação de um agente autônomo que realize a inserção e detecção de padrões de projeto usando uma arquitetura BDI

# Visão geral da abordagem

Figura 01 – Visão geral da abordagem proposta



Fonte: Autoria própria

# O que o agente proposto pode fazer?

- Escanear projeto para verificar quais arquivos são .java.
- Interpretar o código de acordo com a AST.
- Analisar o código para ver se tem algum padrão de projeto aplicável aquela classe lida.

# Como foi implementado?

- Usando *Framework* para agentes chamado *Jadex*.
- *IDE Eclipse Oxygen 3ª*.
- Usou o modelo arquitetural BDI (*Belief – Desire – Intention*).

# Como foi avaliado?

- 25 projetos open-source retirados do GitHub.
- Escolhidos projetos somente implementados com a linguagem de programação Java.
- Ordenados de forma decrescente por popularidade (“starts”).

# Padrões detectados x Qtd. de projetos

Tabela 1 – Resultados agrupados por padrões detectados e qtd de projetos

Patterns detectados	Qtd. de Projetos
Singleton	11
Singleton, Factory, Strategy	7
Singleton, Factory, Strategy, Null Object	1
Singleton, Strategy	1
Singleton, Strategy, Null Object	2
Singleton, Strategy, Null Object	1
Strategy	1
(vazio)	1

Fonte: Autoria própria

# Referências

- COPPIN, B. Inteligência artificial. 1ª. ed. Rio de Janeiro: Grupo Editorial Nacional, 2015.
- FOWLER, M. Refactoring: improving the design of existing code. [S.I.]: Pearson Education India, 1999.
- FAGUNDES, Moser Silva. **Integrating BDI model and Bayesian networks**. 2007. 104f. Dissertação (Mestrado em Ciência da Computação) – Universidade Federado do Rio Grande do Sul. Porto Alegre, 2007.
- FREEMAN, E. et al. Use a Cabeça ! Padrões de Projetos (Design Patterns). 2º. ed. [S.I.]: [s.n.], 2009.
- GAMMA, Erich. **Design patterns: elements of reusable object-oriented software**. Pearson Education India, 1995.
- KERIEVSKY, J. Refatoração para padrões. [S.I.]: Bookman Editora, 2008.

# Referências

- MALL, Rajib. Fundamentals of software engineering. PHI Learning Pvt. Ltd., 2018.
- MARTIN, R. C. Clean code: a handbook of agile software craftsmanship. [S.l.]: Pearson Education, 2009.
- TRIFU, Adrian; SENG, Olaf; GENSSLER, Thomas. Automated design flaw correction in object-oriented systems. In: Eighth European Conference on Software Maintenance and Reengineering, 2004. CSMR 2004. Proceedings. IEEE, 2004. p. 174-183.