

# Análise estática para detectar a evolução da linguagem java em projetos open source

Thiago Gomes Cavalcanti  
Vinícius Correa de Almeida



**Universidade de Brasília**

**Orientador: Prof. Dr. Rodrigo Bonifácio de Almeida**

04 de março de 2016

# Roteiro

## Static Analysys

Thiago  
Cavalcanti -  
Vinícius  
Correa

Cenário

Motivação

Revisão  
Teórica

Diretrizes da  
pesquisa

Metodologia

Arquitetura e  
Funciona-  
mento  
Analisador  
Estático

Resultados

Encerramento

- 1 Cenário
- 2 Motivação
- 3 Revisão Teórica
- 4 Diretrizes da pesquisa
- 5 Metodologia
- 6 Arquitetura e Funcionamento Analisador Estático
- 7 Resultados
- 8 Encerramento

# O Software

## Static Analysys

Thiago  
Cavalcanti -  
Vinícius  
Correa

## Cenário

## Motivação

## Revisão Teórica

## Diretrizes da pesquisa

## Metodologia

## Arquitetura e Funciona- mento Analisador Estático

## Resultados

## Encerramento

- Software atual com construções ultrapassadas.
- Código de baixa qualidade e difícil entendimento.
- Dificuldade em efetuar refactoring e testes.
- Desenvolvedores não conhecem de novas features da linguagem.
- A release não é importante no desenvolvimento.
- Time que está ganhando não se mexe!!!

## Fortran

- Artigo Regrowing a Language de Jeffrey L. Overbey e Ralph E. Johnson.
- Remoção do ultrapassado estilo *do loop* caso este terminasse com *continue*, era substituído por construção equivalente *end do*.
- Remoção do *goto* por uma construção *case* equivalente.
- Uso de palavras reservadas *if*, *while*, ..., como variáveis.
- Introdução de OO em Fortran 2003.

# Aspectos analisados

Static  
Analysys

Thiago  
Cavalcanti -  
Vinícius  
Correa

Cenário

Motivação

Revisão  
Teórica

Diretrizes da  
pesquisa

Metodologia

Arquitetura e  
Funciona-  
mento  
Analisador  
Estático

Resultados

Encerramento

## Aspectos Analisados

- Investigação empírica da adoção de features Java.
- Replicação de um estudo existente de Parnin et al.
- Investigou-se as características:
  - Generics.
  - Lambda Expression.
  - Multi-catch.
  - Try-Resource.
  - Switch-String.
- Oportunidade de evolução em:
  - Lambda Expression, `foreach`, `exists`.
  - Multi-catch, `catch(E1 | E2 | ...)`.
  - Switch-String, `if(string.equals())`.

## Referências

- Chris Parnin, Christian Bird, and Emerson Murphy-Hill 2011. **Java generics adoption: How new features are introduced, championed, or ignored.**
- Jefrey L. Overbey and Ralph E. Johnson. **Regrowing a language: refactoring tools allow programming languages to evolve.**
- Dyer, Robert and Rajan, Hridesh and Nguyen, Hoan Anh and Nguyen, Tien N. **Mining billions of AST nodes to study actual and potential usage of Java language features.**

# Problema

Static  
Analysys

Thiago  
Cavalcanti -  
Vinícius  
Correa

Cenário

Motivação

Revisão  
Teórica

Diretrizes da  
pesquisa

Metodologia

Arquitetura e  
Funciona-  
mento  
Analisador  
Estático

Resultados

Encerramento

## Problema

- Desenvolver um analisador estático de código Java.
- Pesquisar características ultrapassadas em projetos atuais.
- Refazer estudo de Generis.

# Hipóteses

Static  
Analysys

Thiago  
Cavalcanti -  
Vinícius  
Correa

Cenário

Motivação

Revisão  
Teórica

Diretrizes da  
pesquisa

Metodologia

Arquitetura e  
Funciona-  
mento  
Analisador  
Estático

Resultados

Encerramento

## Hipótese 1

- Novas características são pouco utilizadas ou ignoradas no desenvolvimento do software.

## Hipótese 2

- Código obsoleto é mantido em releases atuais.



# Objetivos

## Static Analysys

Thiago  
Cavalcanti -  
Vinícius  
Correa

Cenário

Motivação

Revisão  
Teórica

Diretrizes da  
pesquisa

Metodologia

Arquitetura e  
Funciona-  
mento  
Analisador  
Estático

Resultados

Encerramento

## Geral

- Identificar de forma eficiente construções na linguagem Java.

## Específicos

- Elaborar um analisador estático flexível.

## Ordem das tarefas

- Revisão Bibliográfica
  - Seleção do projetos.
  - Escolha de projetos opensource.
- Implementação do analisador estático.
  - Definir arquitetura.
  - Implementação.
  - Coleta de dados.
- Refazer os passos do artigo de Generics(escrever artigo).
- Contribuições adicionais.

# Analizador Estático

Static  
Analysys

Thiago  
Cavalcanti -  
Vinícius  
Correa

Cenário

Motivação

Revisão  
Teórica

Diretrizes da  
pesquisa

Metodologia

Arquitetura e  
Funciona-  
mento  
Analizador  
Estático

Resultados

Encerramento

## Classes Utilitárias

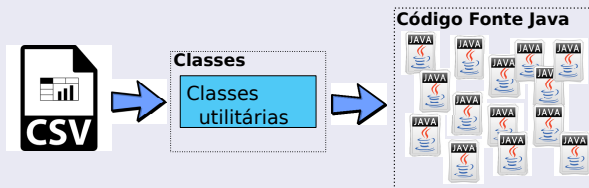


Figura: Classes Utilitárias.



# Resultados

Static  
Analysys

Thiago  
Cavalcanti -  
Vinícius  
Correa

Cenário

Motivação

Revisão  
Teórica

Diretrizes da  
pesquisa

Metodologia

Arquitetura e  
Funciona-  
mento  
Analisador  
Estático

Resultados

Encerramento

## Generics

- 16% dos projetos não declaram Generics.
- *Commons Collections* possui 75% dos tipos genéricos.
- De 925.925 atributos e variáveis apenas 9.1% são tipos genéricos (84880).
- `List<String>` possui quase 25% de todos os genéricos conforme Parnis.
- Apenas 0.84% de 730.720 métodos parametrizados.
- Considerando o tipo e idade da aplicação Generics não exerce influencia significativa.

# Resultados

Static  
Analisis

Thiago  
Cavalcanti -  
Vinícius  
Correa

Cenário

Motivação

Revisão  
Teórica

Diretrizes da  
pesquisa

Metodologia

Arquitetura e  
Funciona-  
mento  
Analisador  
Estático

Resultados

Encerramento

## Generics

**Tabela:** Resumo dos tipos agrupados por idade e do tipo dos projetos.

Tipo de Projeto	Antes Java SE 5.0	Tipo	Tipo Genérico	(%)
Aplication	Yes	18168	177	0.97
Aplication	No	16148	744	4.61
Library	Yes	21537	1198	5.56
Library	No	22639	947	4.18
Server/Database	Yes	18038	552	3.10
Server/Database	No	11790	760	6.45

# Resultados

Static  
Analisis

Thiago  
Cavalcanti -  
Vinícius  
Correa

Cenário

Motivação

Revisão  
Teórica

Diretrizes da  
pesquisa

Metodologia

Arquitetura e  
Funciona-  
mento  
Analisador  
Estático

Resultados

Encerramento

## Generics

**Tabela:** Tipo declarado X Número de instância

Tipo	Número de Instância
List<String>	4993
Class<?>	3033
Set<String>	2872
Map<String,String>	2294
Map<String,Object>	1554

# Resultados

Static  
Analysys

Thiago  
Cavalcanti -  
Vinícius  
Correa

Cenário

Motivação

Revisão  
Teórica

Diretrizes da  
pesquisa

Metodologia

Arquitetura e  
Funciona-  
mento  
Analisador  
Estático

Resultados

Encerramento

## Generics

- Foram encontrados 4355 que fazem polimorfismo parametrizado.
- 30% do polimorfismo pesquisado possui Super, extends ou ?.
- Todos os tipos que utilizam ? ou Super, utilizam o extends.
- De 925.925 atributos e variáveis apenas 9.1% são tipos genéricos (84880).
- Também foi descoberto que quase 10% de todos os atributos e variáveis são tipos genéricos.
- Considerando o tipo e idade da aplicação Generics não exerce influencia significativa.



# Resultados

Static  
Analysys

Thiago  
Cavalcanti -  
Vinícius  
Correa

Cenário

Motivação

Revisão  
Teórica

Diretrizes da  
pesquisa

Metodologia

Arquitetura e  
Funciona-  
mento  
Analisador  
Estático

Resultados

Encerramento

## Lambda

- Apenas 5% faz uso de Lamba Expression.
- Utilização apenas em Teste Unitários.
- Esforços individuais para incorporar Lambda.

**Tabela:** Ocorrências de Expressões Lambda.

Sistema	Ocorrências Expressões Lambda
Hibernate	168
Jetty	2
Lucene	11
Spark	77
Spring-framework	121

# Resultados

Static  
Analysys

Thiago  
Cavalcanti -  
Vinícius  
Correa

Cenário

Motivação

Revisão  
Teórica

Diretrizes da  
pesquisa

Metodologia

Arquitetura e  
Funciona-  
mento  
Analisador  
Estático

Resultados

Encerramento

## Oportunidade de aplicar Lambda

- EXIST PATTERN, FILTER PATTERN e MAP PATTERN.
- Foram encontrados 2496 oportunidades.
- 2190 correspondem a MAP PATTERN, 87%.

## Multi-catch

- Representam 1% do mecanismo de exceção.
- 95% dos projetos com oportunidades reais.
- 1474 Try com catch similares repetidos.

**Tabela:** Oportunidades de multi-catch por tipo do sistema.

Natureza	Ocorrências
Application	551
Library	464
Servers - Database	459

# Resultados

## Static Analysis

Thiago Cavalcanti -  
Vinícius Correa

Cenário

Motivação

Revisão Teórica

Diretrizes da pesquisa

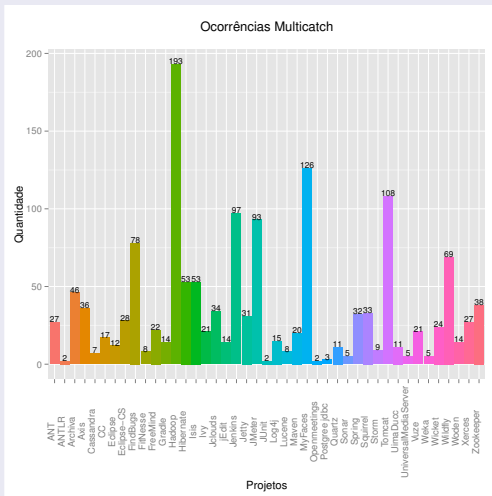
Metodologia

Arquitetura e Funcionamento  
Analisador Estático

Resultados

Encerramento

## Multi-catch



# Resultados

Static  
Analysys

Thiago  
Cavalcanti -  
Vinícius  
Correa

Cenário

Motivação

Revisão  
Teórica

Diretrizes da  
pesquisa

Metodologia

Arquitetura e  
Funciona-  
mento  
Analisador  
Estático

**Resultados**

Encerramento

## Try-Resource

- Representam 1.15% de todos os try dos sistemas.
- Encontrados 1616 ocorrências em 13 Projetos.
- Server-Database concentram 91% das ocorrências.

# Resultados

## Static Analysis

Thiago  
Cavalcanti -  
Vinícius  
Correa

Cenário

Motivação

Revisão  
Teórica

Diretrizes da  
pesquisa

Metodologia

Arquitetura e  
Funciona-  
mento  
Analisador  
Estático

Resultados

Encerramento

## Try-Resource

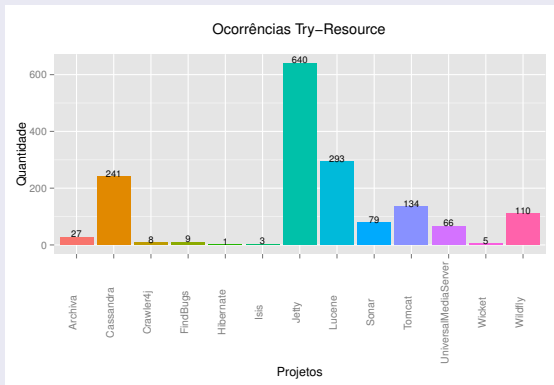


Figura: Adoção de Try-Resource nos projetos.

# Resultados

Static  
Analysys

Thiago  
Cavalcanti -  
Vinícius  
Correa

Cenário

Motivação

Revisão  
Teórica

Diretrizes da  
pesquisa

Metodologia

Arquitetura e  
Funciona-  
mento  
Analisador  
Estático

Resultados

Encerramento

## Switch-String

Menos de 1% dos 6827 Switch usam String.

**Tabela:** Adoção Switch String por tipo do sistema.

Sistema	Ocorrências
Cassandra	14
FindBugs	3
Jetty	16
Lucene	2
Sonar	1
Spring	2
Tomcat	8
UniversalMediaServer	18
Wicket	1
Wildfly	1
Total	66



## Switch-String

- Pesquisa por `if(String.equals())`.
- Foram encontrados 4940 em 45 dos 46 sistemas.
- Switch-String mais eficiente que `if`.

**Tabela:** Oportunidade de aplicar switch por tipo de sistema.

Natureza	Ocorrências
Application	1773
Library	1881
Servers - Database	1286
Total	4940



# Resultados

Static  
Analysis

Thiago  
Cavalcanti -  
Vinícius  
Correa

Cenário

Motivação

Revisão  
Teórica

Diretrizes da  
pesquisa

Metodologia

Arquitetura e  
Funciona-  
mento  
Analisador  
Estático

Resultados

Encerramento

## Switch-String

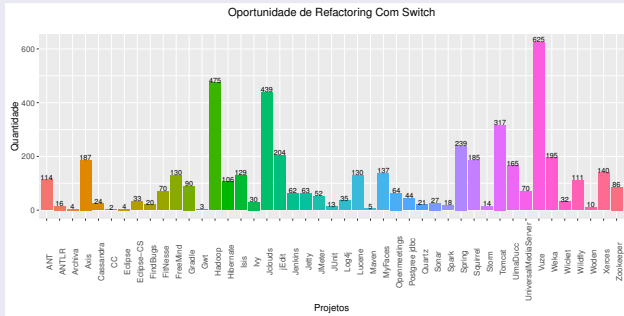


Figura: Oportunidades de *refactoring* em *if-then-else* por sistema.

# Conclusão

Static  
Analysys

Thiago  
Cavalcanti -  
Vinícius  
Correa

Cenário

Motivação

Revisão  
Teórica

Diretrizes da  
pesquisa

Metodologia

Arquitetura e  
Funciona-  
mento  
Analisador  
Estático

Resultados

Encerramento

## Conclusão

- Código obsoleto é mantido em versões atuais.
- Apesar da grande expectativa sobre Lambda, não existe um entusiasmo para adoção.
- A realease da linguagem não é importante no desenvolvimento.
- Temos um analisador flexível e estável para tentar automatizar refactoring.

# Encerramento

Static  
Analisis

Thiago  
Cavalcanti -  
Vinícius  
Correa

Cenário

Motivação

Revisão  
Teórica

Diretrizes da  
pesquisa

Metodologia

Arquitetura e  
Funciona-  
mento  
Analisador  
Estático

Resultados

Encerramento

## Perguntas

