“Um Estudo Quantitativo Sobre o Uso de Herança e Interface em Sistemas Java”

Aluno: Carlos Eduardo de Carvalho Dantas
Orientador: Marcelo de Almeida Maia

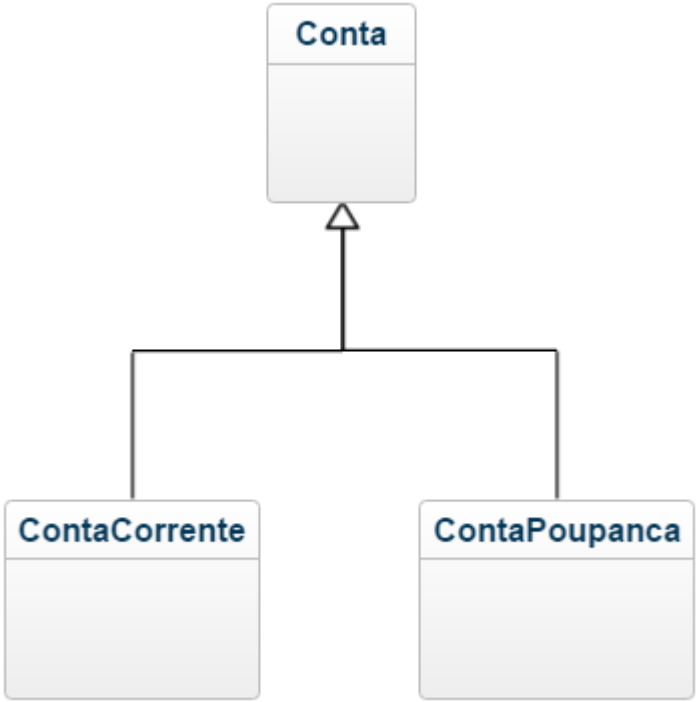
UFU

Fevereiro/2017

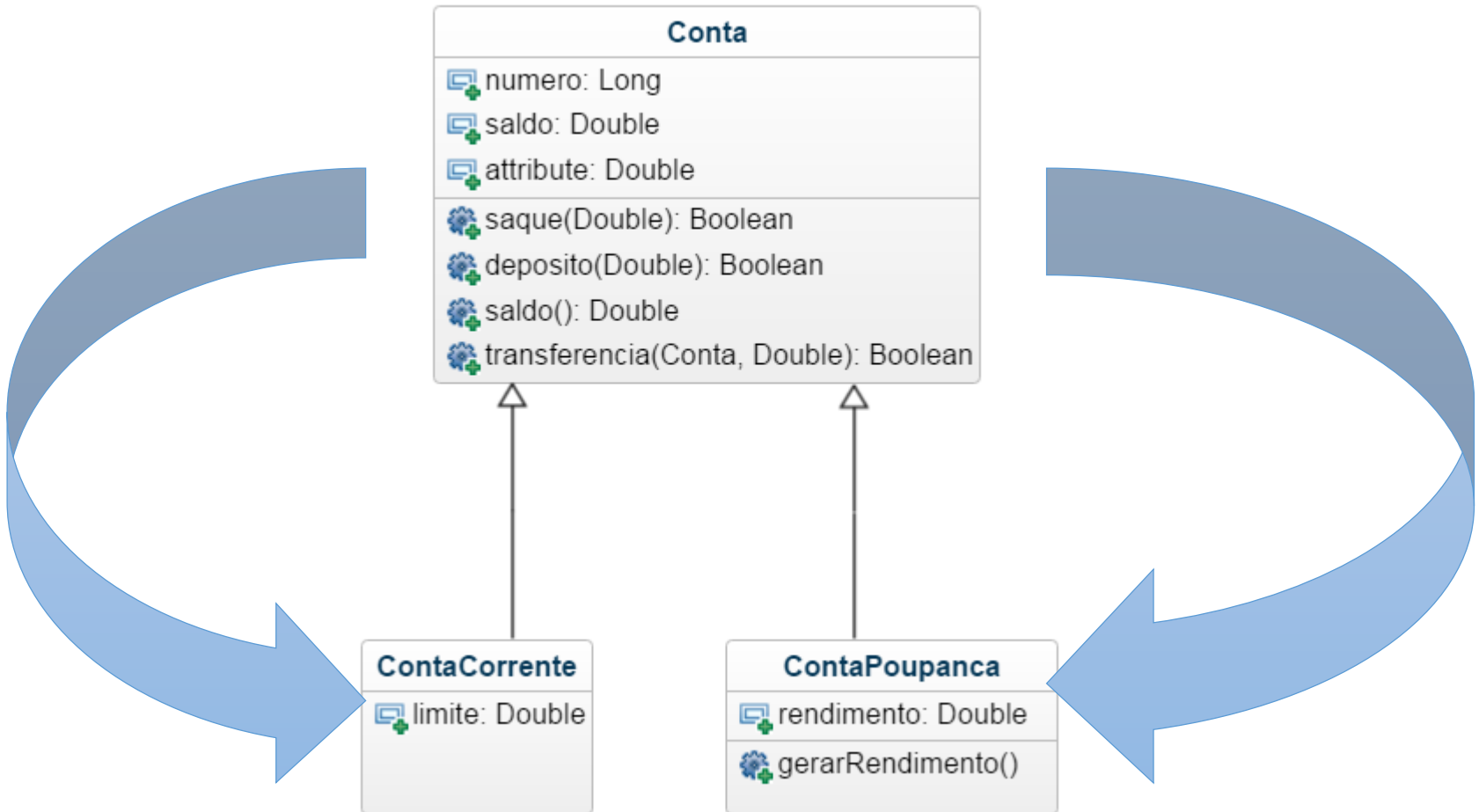
ROTEIRO

1. **Contextualização do Problema**.
2. Objetivos.
3. Metodologia.
4. Resultados.
4. Ameaças à validade.
5. Conclusão.

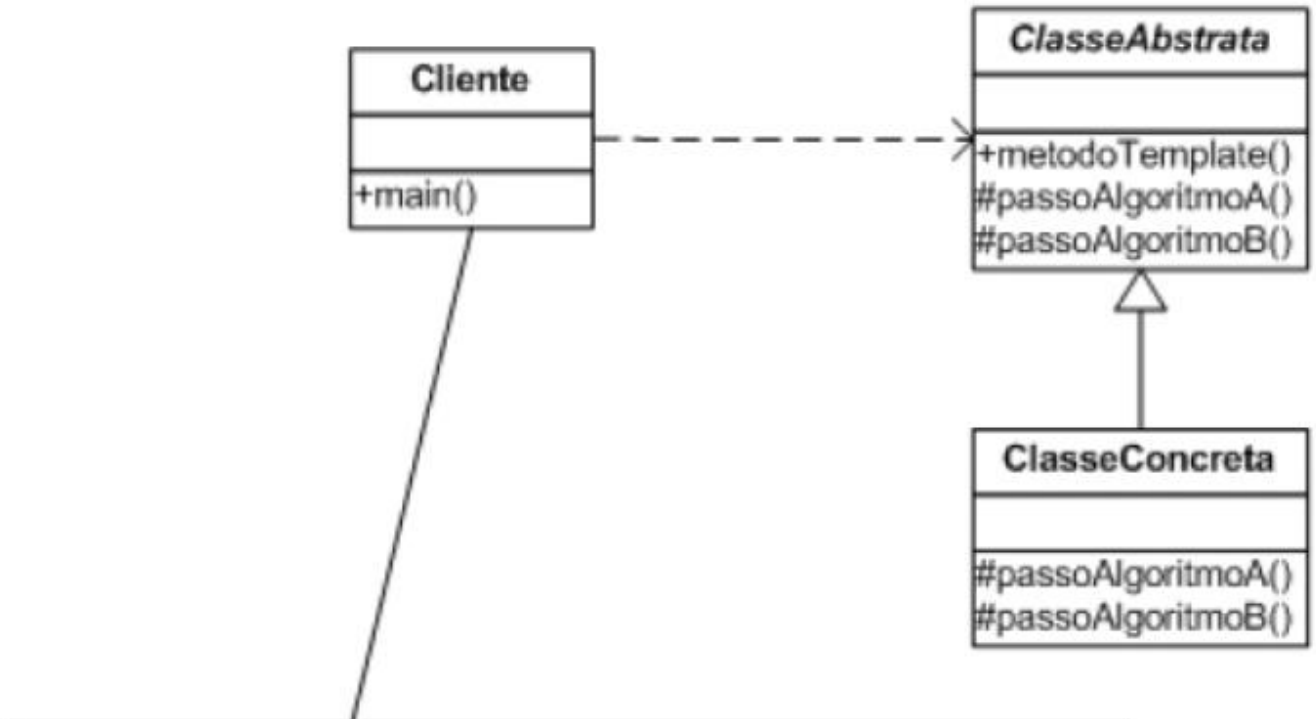
HERANÇA



Reaproveitamento de Código

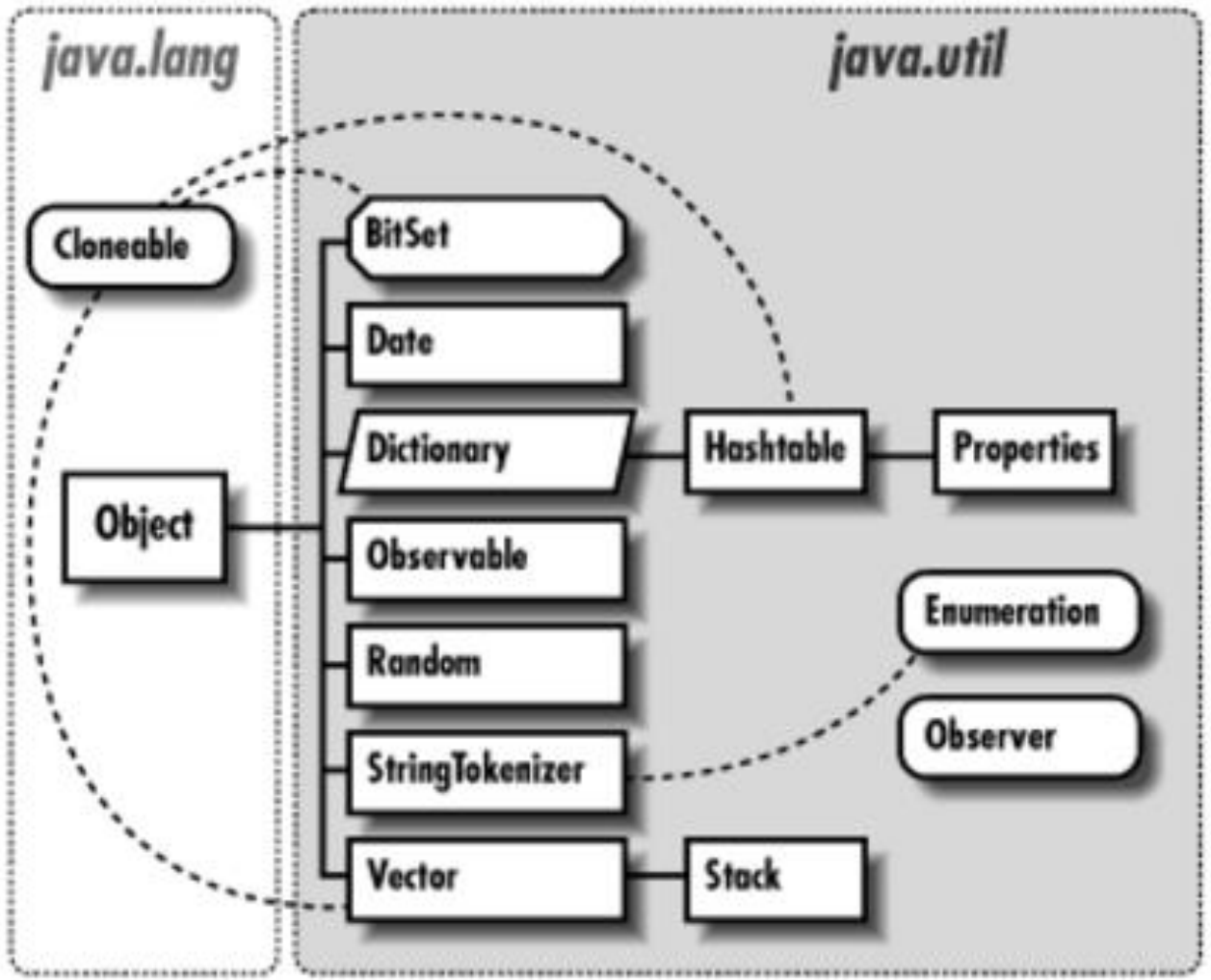


Design Patterns

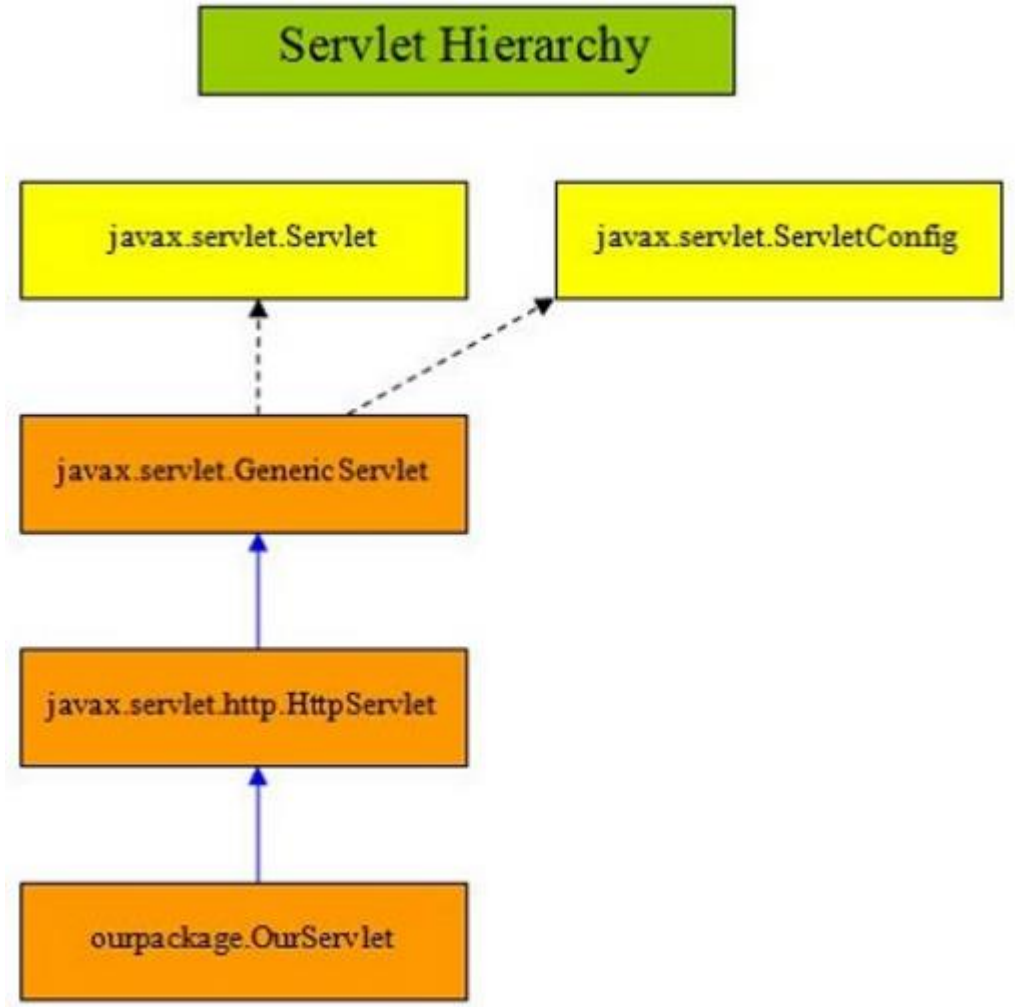



```
public static void main(String[] args){
   ClasseAbstrata c = new ClasseConcreta();
   c.metodoTemplate();
}
```

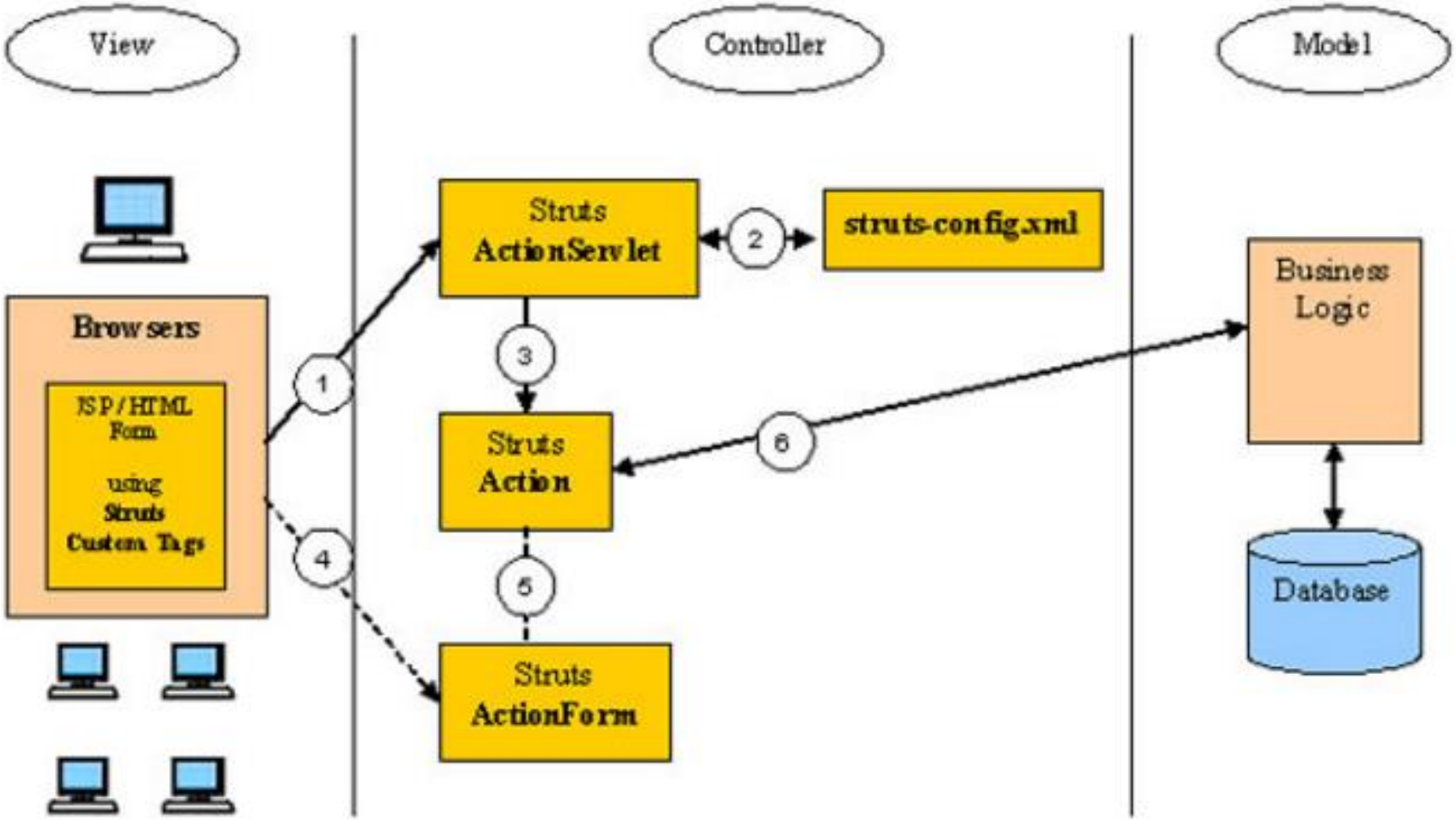
Sintaxe da linguagem Java



APIs


Servlet Hierarchy
javax.servlet.Servlet
javax.servlet.ServletConfig
javax.servlet.GenericServlet
javax.servlet.http.HttpServlet
ourpackage.OurServlet

Frameworks


View
Controller
Model
Browsers
JSP/HTML Form
using Struts Custom Tags
Struts ActionServlet
struts-config.xml
Struts Action
Struts ActionForm
Business Logic
Database
1
2
3
4
5
6

Entretanto...

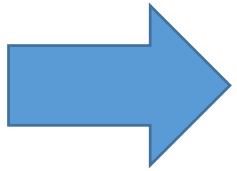
“The introduction of inheritance severely compromises the benefits of this encapsulation”

1986

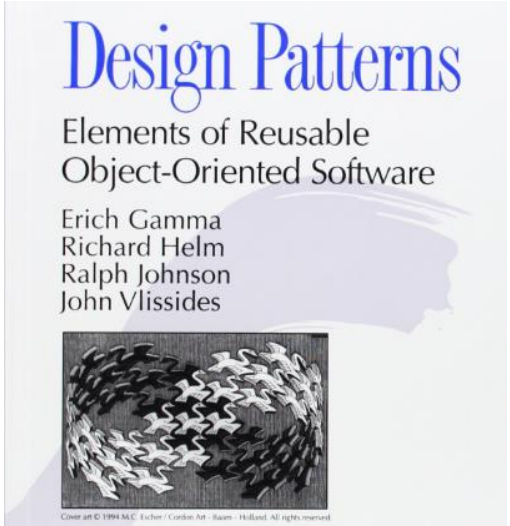
Encapsulation and Inheritance in Object-Oriented Programming Languages

Alan Snyder
Software Technology Laboratory
Hewlett-Packard Laboratories
P.O. Box 10490
Palo Alto CA 94303-0971
(415) 857-8764

“Favor object composition over class inheritance”



1995



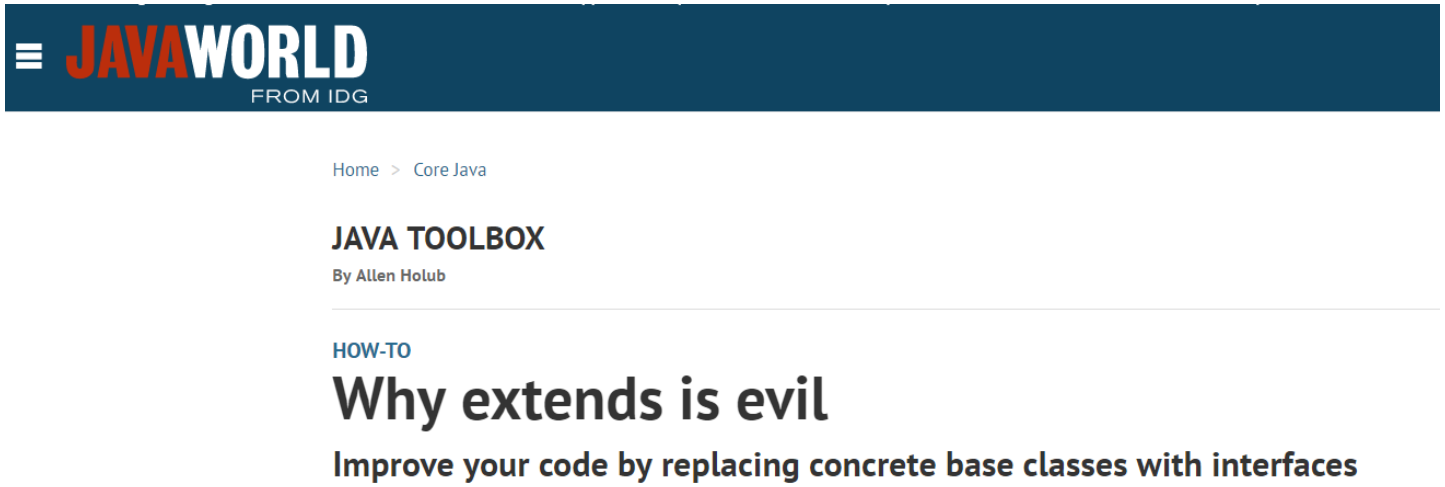
“Systems without inheritance were easier to modify than the corresponding systems containing three or five levels of inheritance”

2000

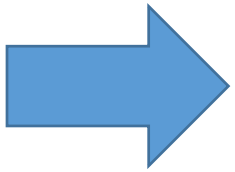
Experimental Assessment of the Effect of Inheritance on the Maintainability of Object-Oriented Systems

R. Harrison, S. Counsell, R. Nithi
Department of Electronics and Computer Science,
Mountbatten Building,
University of Southampton, Southampton, SO17 1BJ, U.K.
Tel. +44 (0) 1703 593249, Fax. +44 (0) 1703 593045.
email: rh@ecs.soton.ac.uk

2003

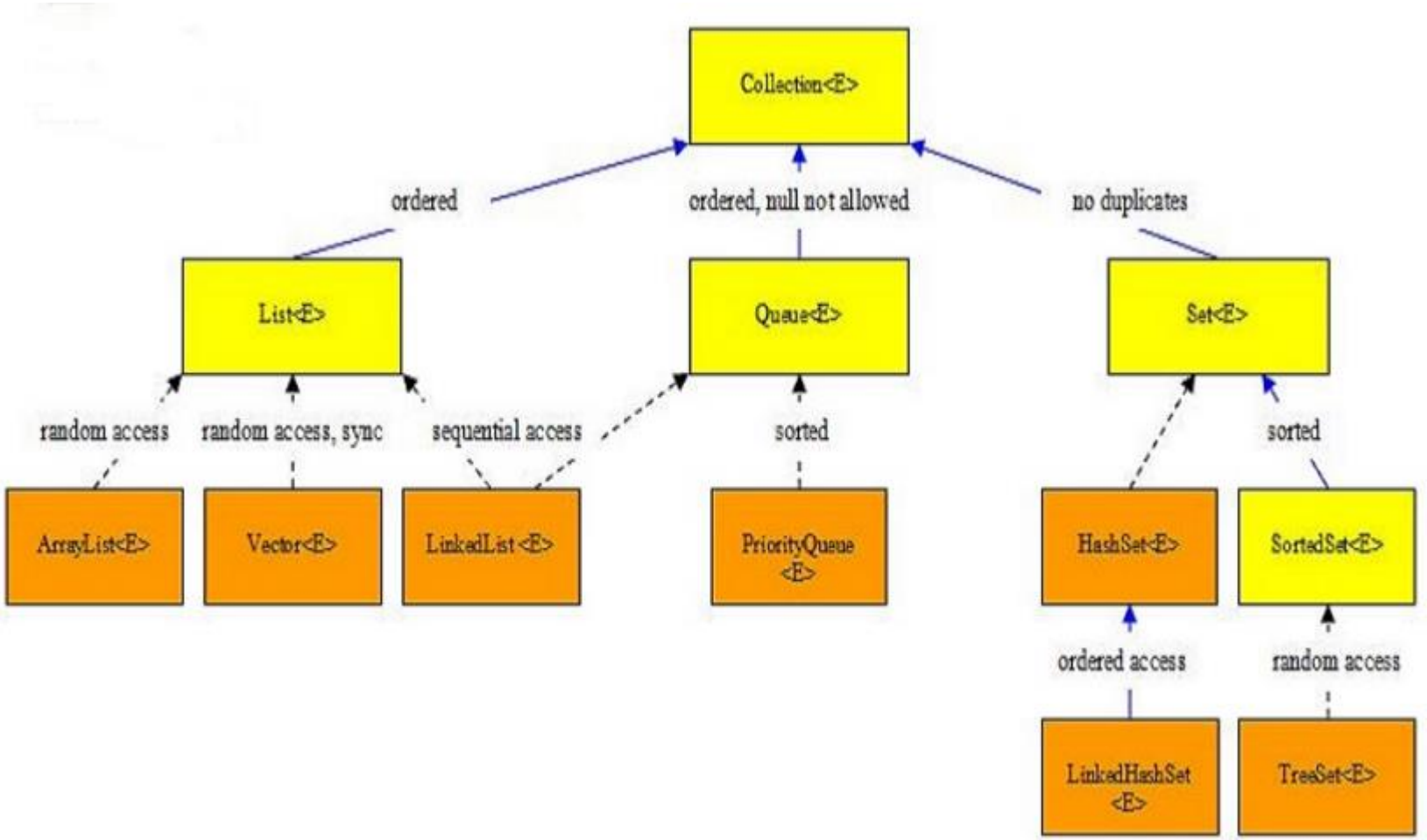

JAVAWORLD
FROM IDG
Home > Core Java
JAVA TOOLBOX
By Allen Holub
HOW-TO
Why extends is evil
Improve your code by replacing concrete base classes with interfaces

“Used inappropriately, inheritance leads to fragile software”

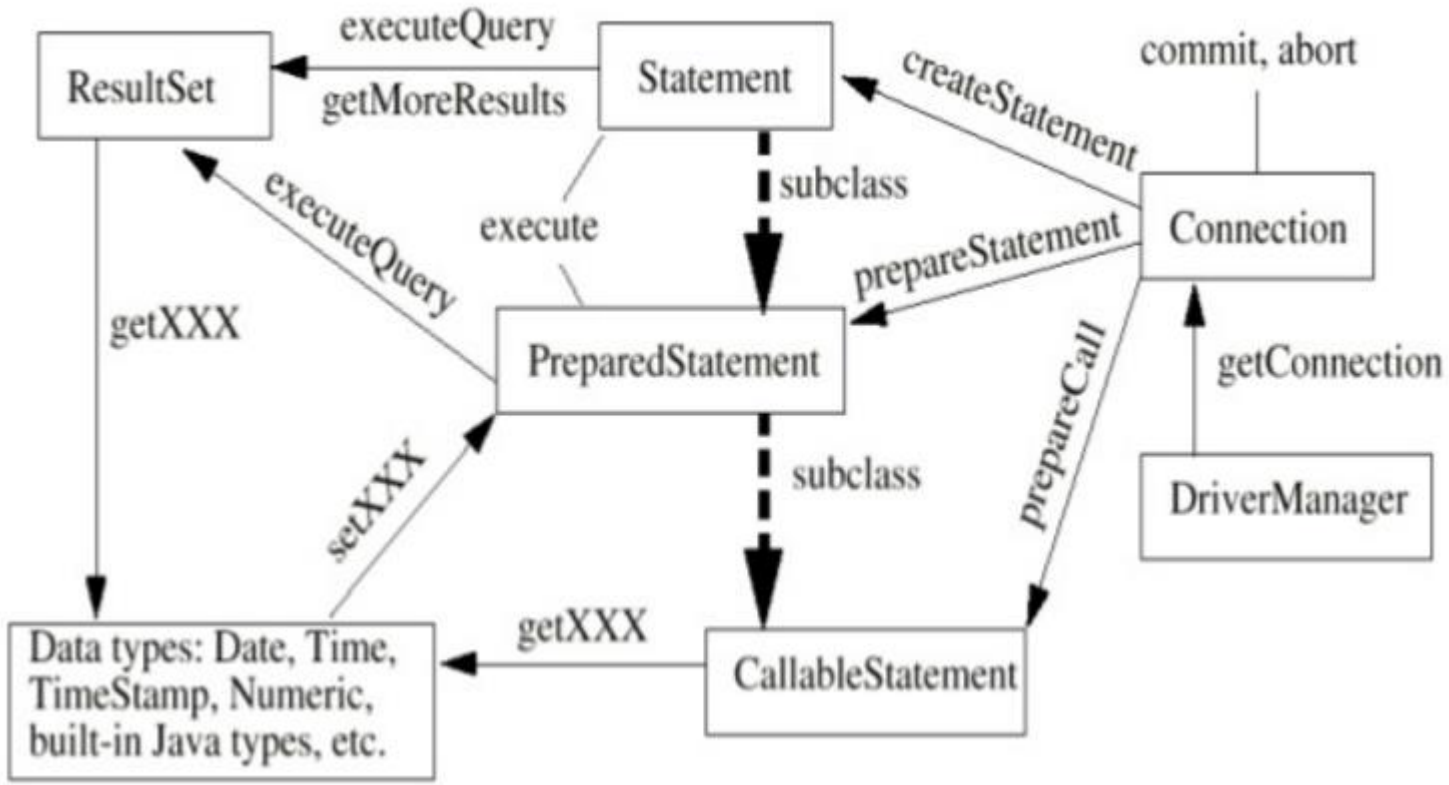


2008

Collections Framework



JDBC API


executeQuery
ResultSet
getMoreResults
Statement
createStatement
commit, abort
subclass
executeQuery
execute
Connection
prepareStatement
getXXX
PreparedStatement
getConnection
prepareCall
setXXX
subclass
DriverManager
getXXX
Data types: Date, Time,
TimeStamp, Numeric,
built-in Java types, etc.
CallableStatement

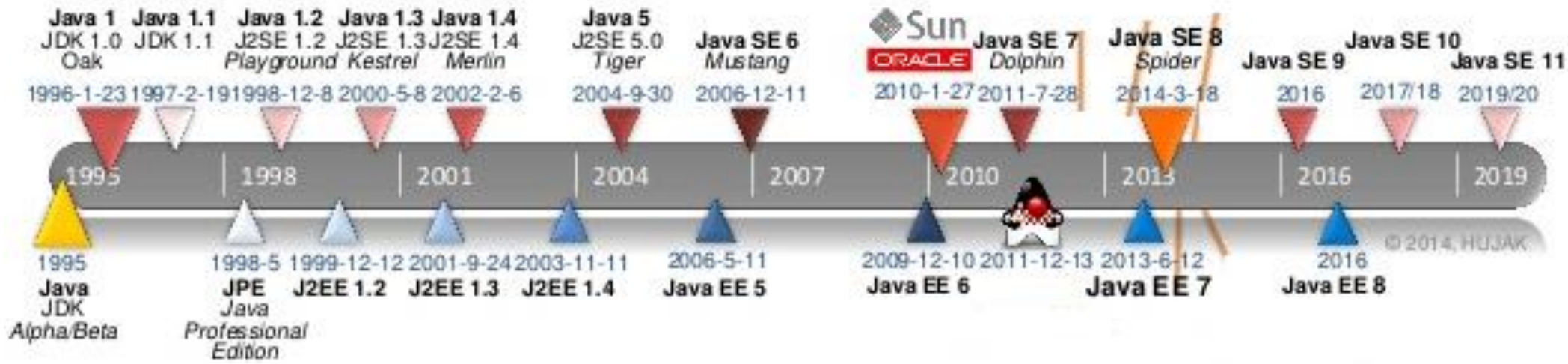
Estas recomendações exerceram alguma influência nos desenvolvedores em geral?

ROTEIRO

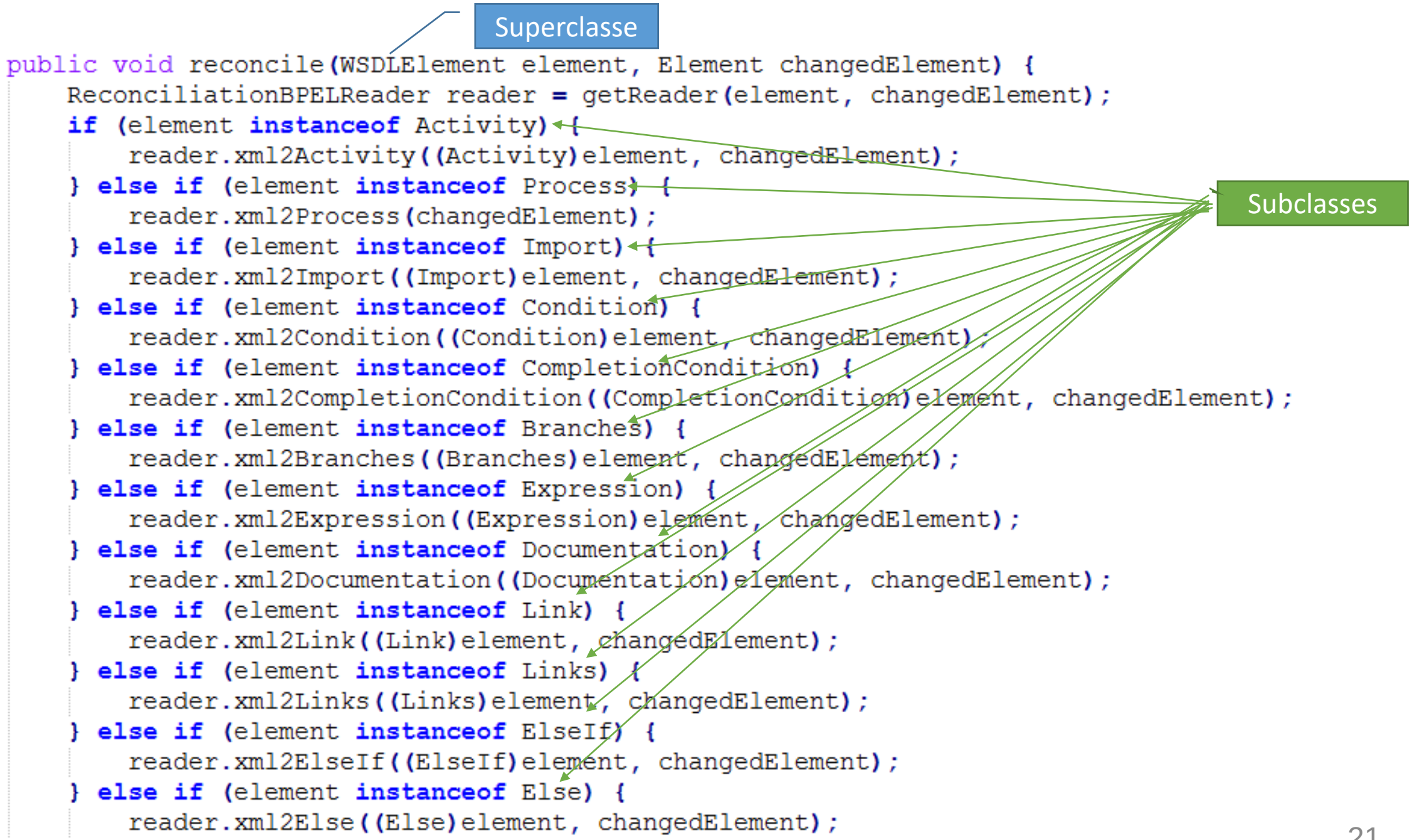
1. Contextualização do Problema.
2. **Objetivos.**
3. Metodologia.
4. Resultados.
4. Ameaças à validade.
5. Conclusão.

Avaliar se o recurso de herança vêm sendo empregado de maneira diferente em sistemas mais recentes

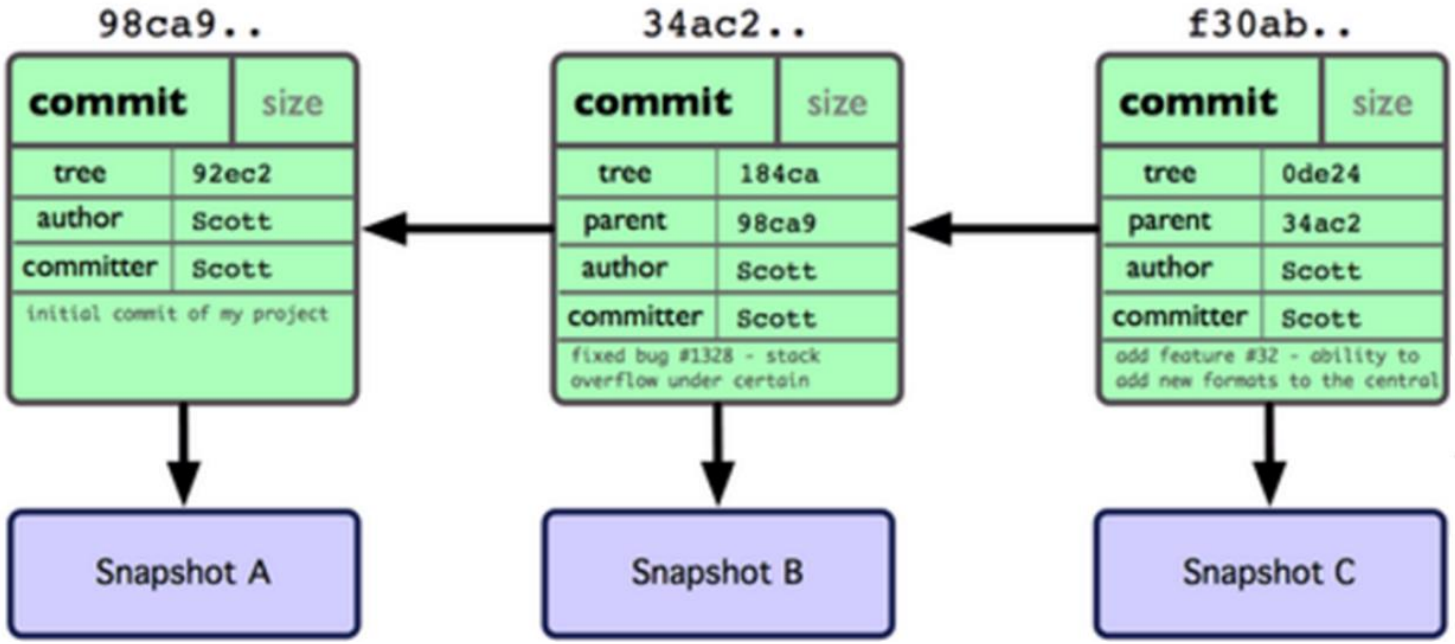
RQ #1) A época em que o sistema foi construído exerce alguma influência sobre a frequência no uso de herança ou implementação de interfaces?



RQ #2: A época em que o sistema foi construído exerce alguma influência sobre a quantidade das quebras de encapsulamento por instanceof?


Superclasse
Subclasses
public void reconcile(WSDLElement element, Element changedElement) {
ReconciliationBPELReader reader = getReader(element, changedElement);
if (element instanceof Activity) {
reader.xml2Activity((Activity)element, changedElement);
} else if (element instanceof Process) {
reader.xml2Process(changedElement);
} else if (element instanceof Import) {
reader.xml2Import((Import)element, changedElement);
} else if (element instanceof Condition) {
reader.xml2Condition((Condition)element, changedElement);
} else if (element instanceof CompletionCondition) {
reader.xml2CompletionCondition((CompletionCondition)element, changedElement);
} else if (element instanceof Branches) {
reader.xml2Branches((Branches)element, changedElement);
} else if (element instanceof Expression) {
reader.xml2Expression((Expression)element, changedElement);
} else if (element instanceof Documentation) {
reader.xml2Documentation((Documentation)element, changedElement);
} else if (element instanceof Link) {
reader.xml2Link((Link)element, changedElement);
} else if (element instanceof Links) {
reader.xml2Links((Links)element, changedElement);
} else if (element instanceof ElseIf) {
reader.xml2ElseIf((ElseIf)element, changedElement);
} else if (element instanceof Else) {
reader.xml2Else((Else)element, changedElement);

RQ #3: A época em que o sistema foi construído exerce alguma influência sobre a quantidade de alterações corretivas em classes com herança e interface?



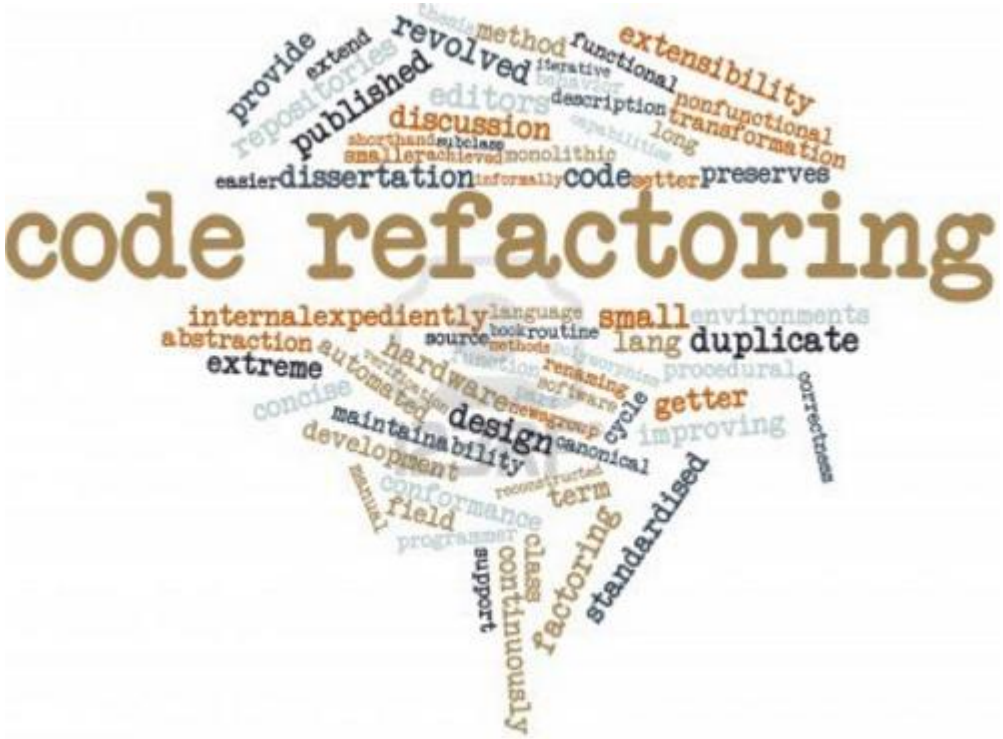
RQ #4: Classes com herança ou implementação de interface possuem níveis adequados de coesão e acoplamento?

CBO	ELOC
LCOM	NOM
RFC	WMC

RQ #5: Quais Code Smells ocorrem predominantemente em classes com herança ou implementação de interface?

Class Data Should be Private
Complex Class
Functional Decomposition
God Class
Lazy Class
Long Method
Spagheti Code

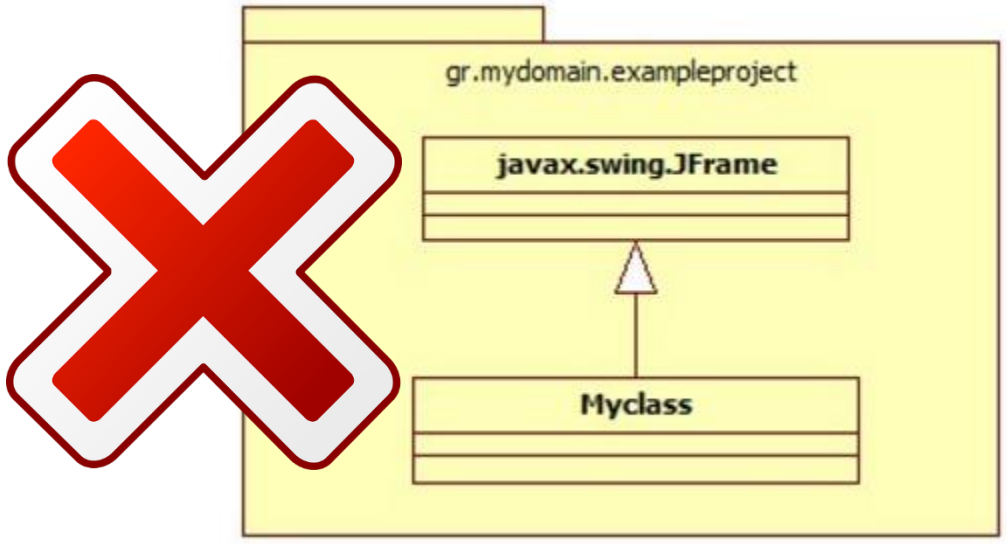
RQ #6: Com qual frequência ocorre adição ou remoção de herança e implementação de interfaces sobre as classes? E por quais razões estas operações são realizadas?

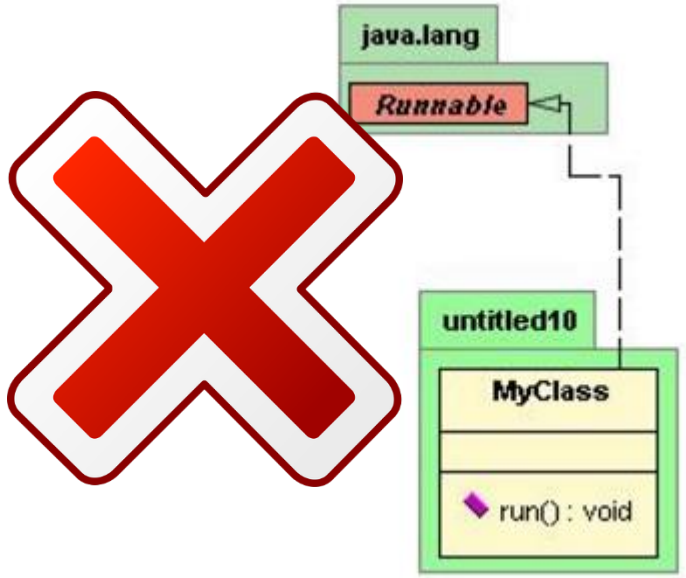


ROTEIRO

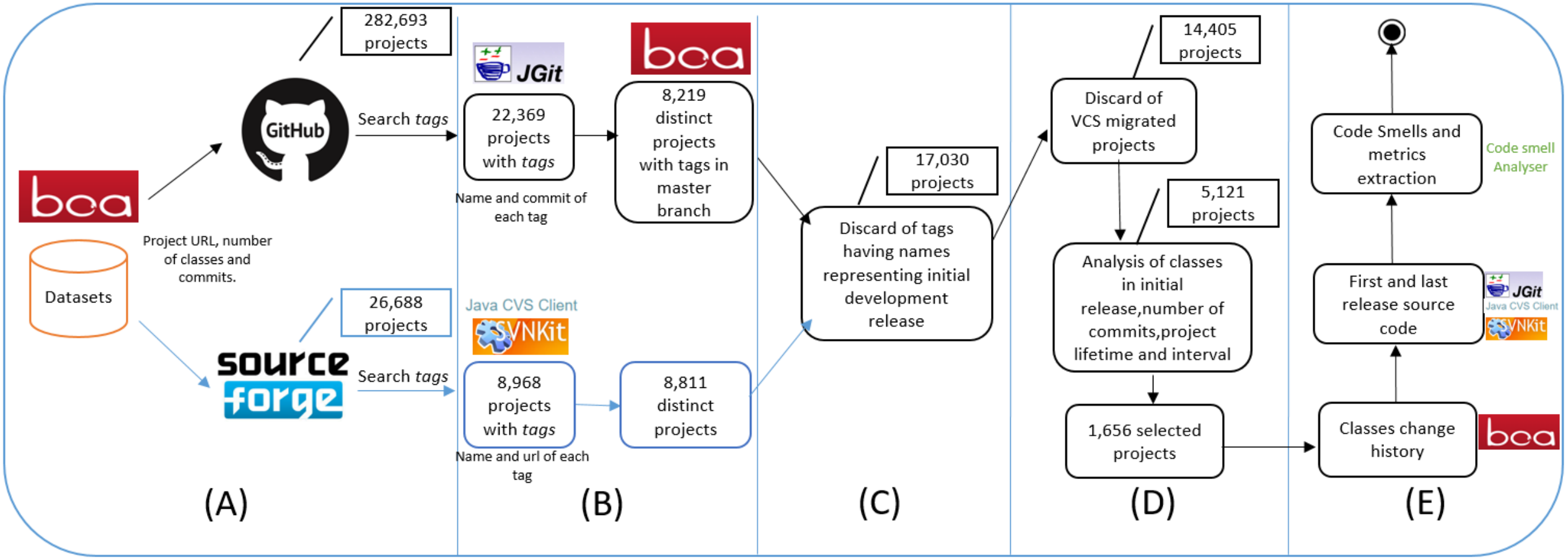
1. Contextualização do Problema.
2. Objetivos.
3. **Metodologia.**
4. Resultados.
4. Ameaças à validade.
5. Conclusão.

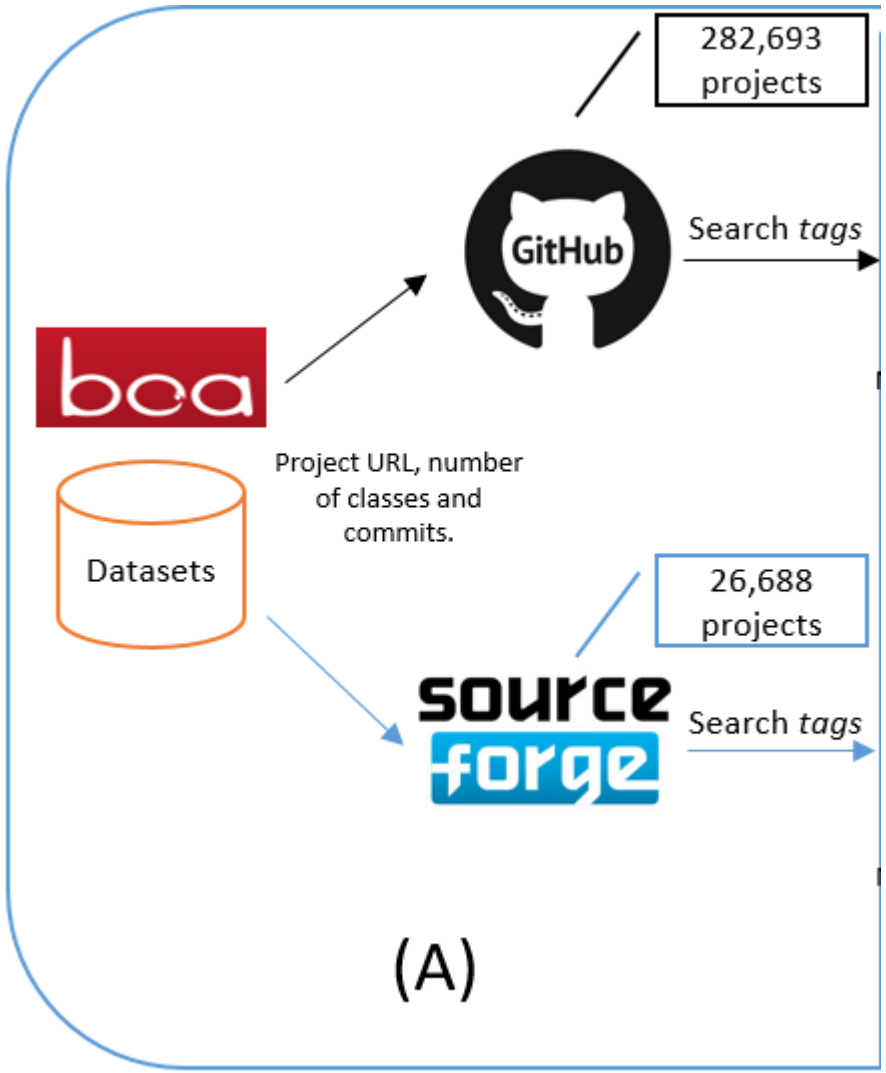
Considera-se apenas classes internas

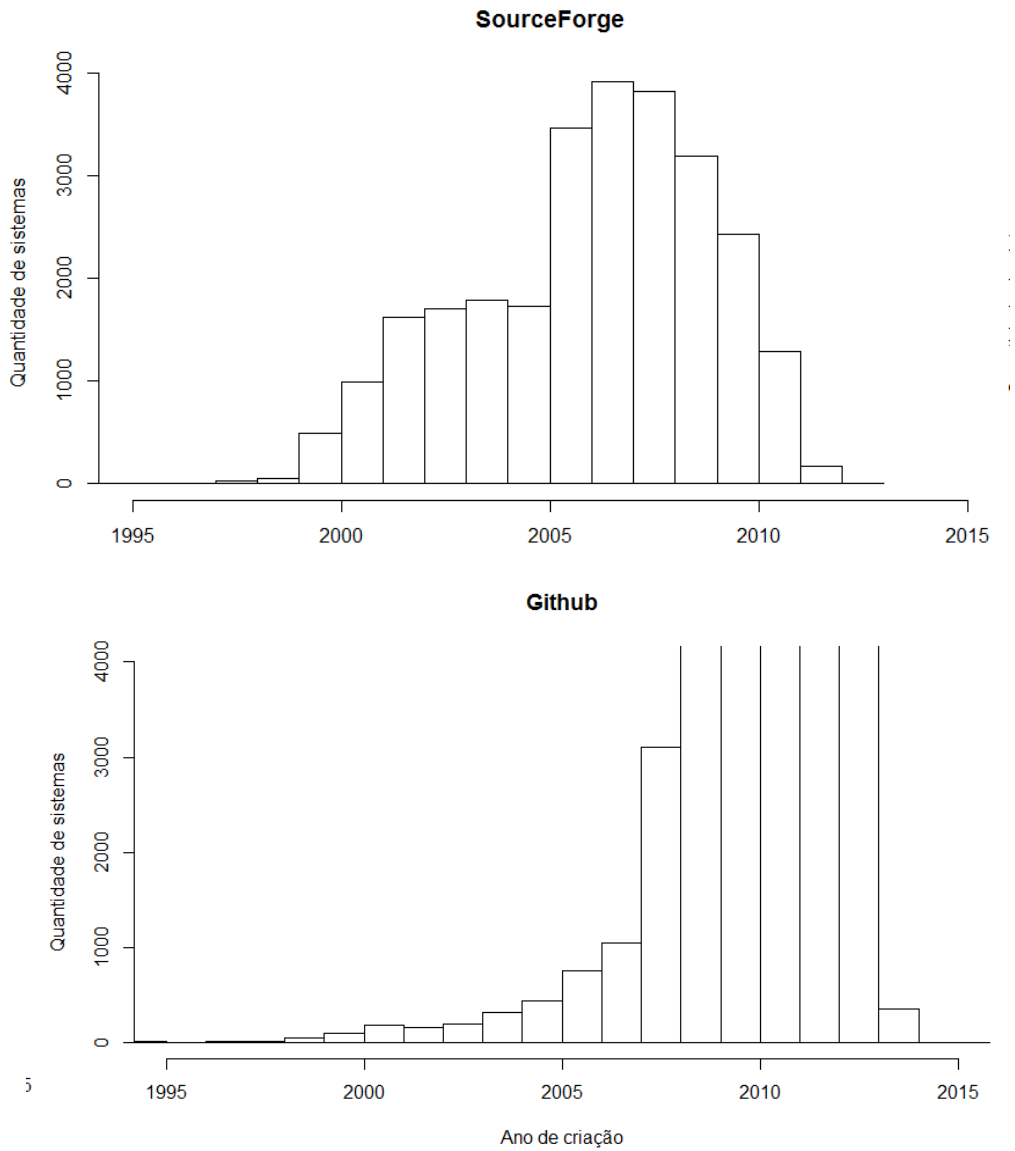


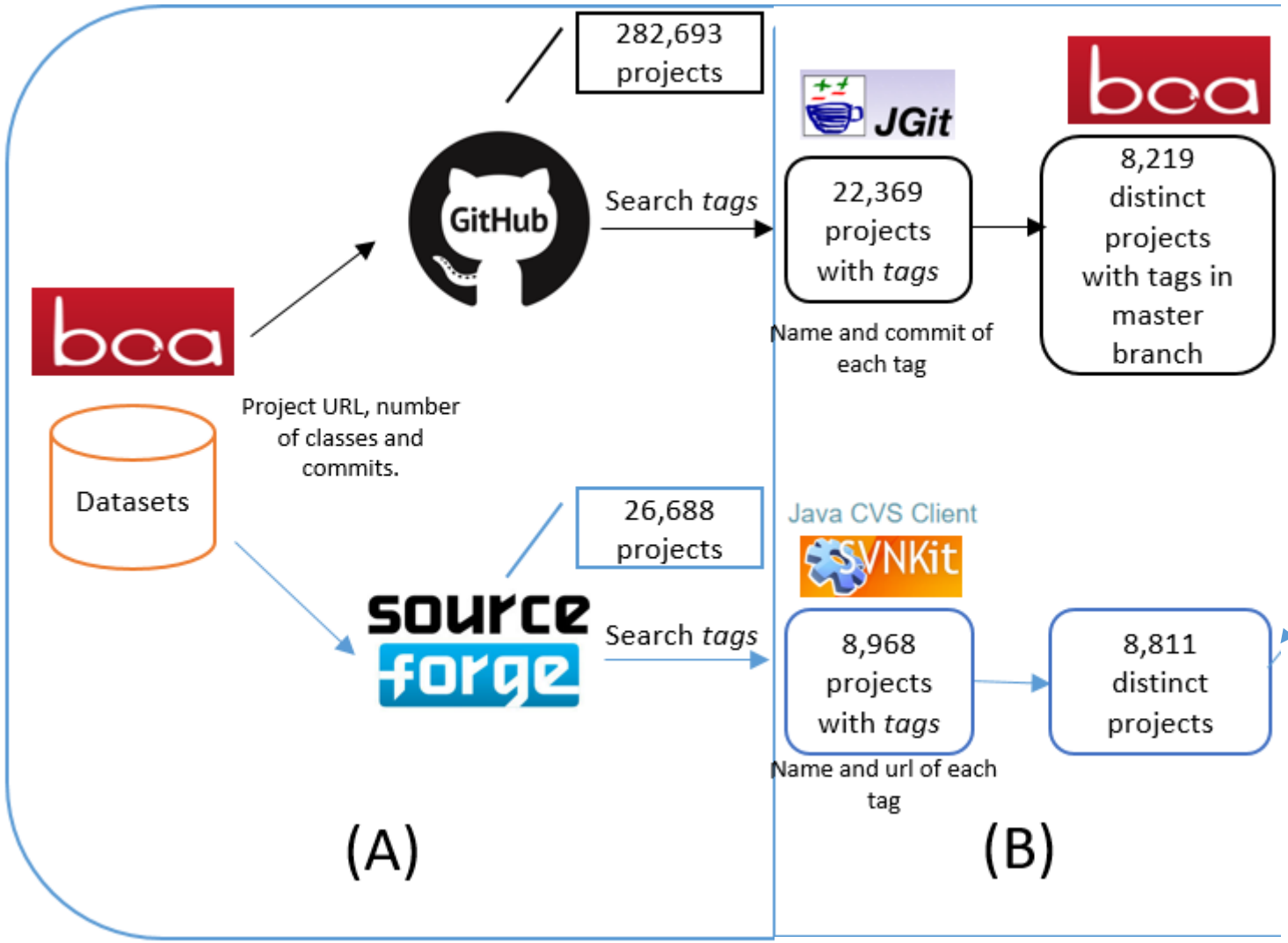


Seleção dos Sistemas

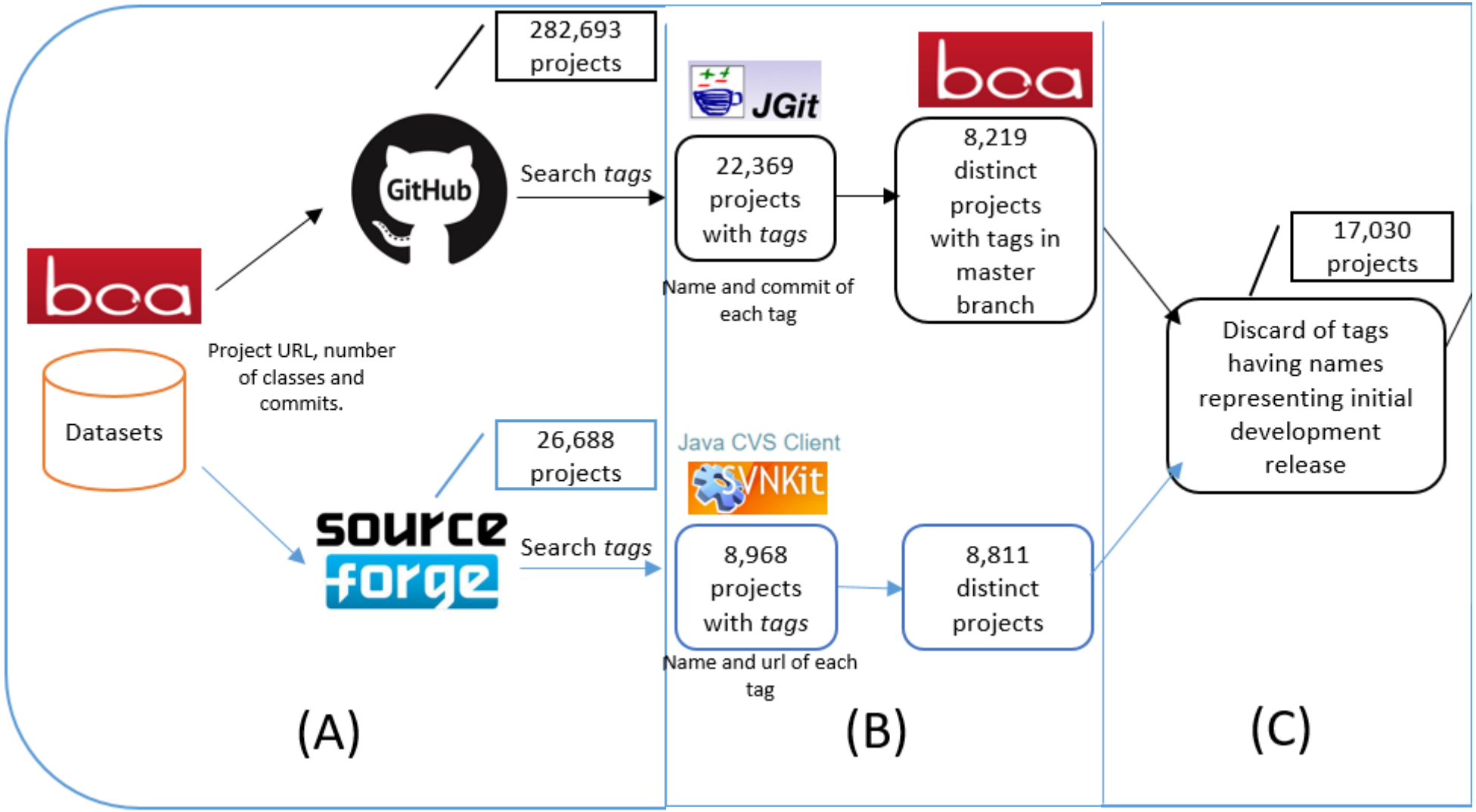



282,693
projects
GitHub
Search *tags*
bea
Project URL, number
of classes and
commits.
Datasets
26,688
projects
source
forge
Search *tags*
(A)

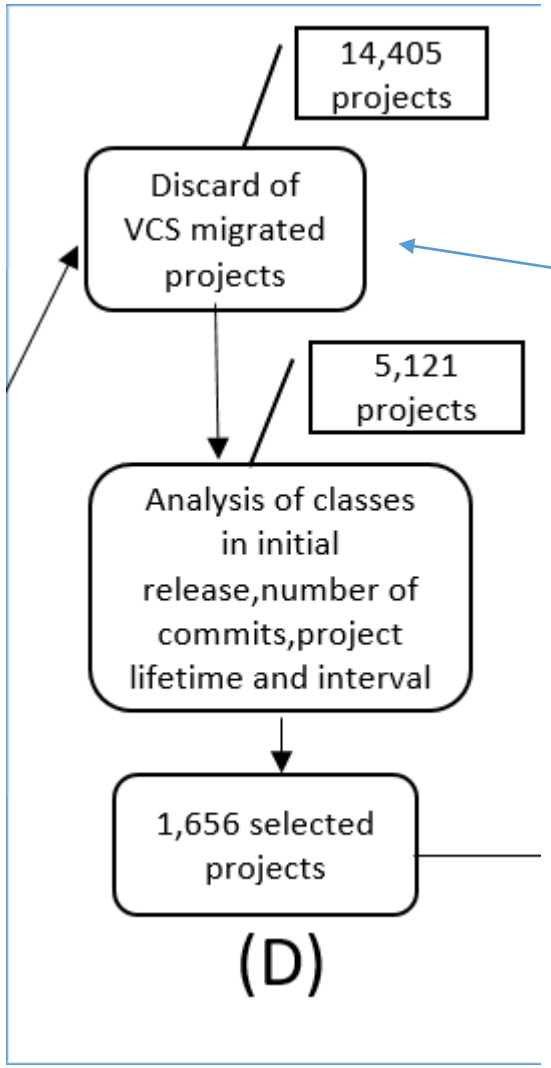

SourceForge
Quantidade de sistemas
0
1000
2000
3000
4000
1995
2000
2005
2010
2015
Github
Quantidade de sistemas
0
1000
2000
3000
4000
1995
2000
2005
2010
2015
Ano de criação



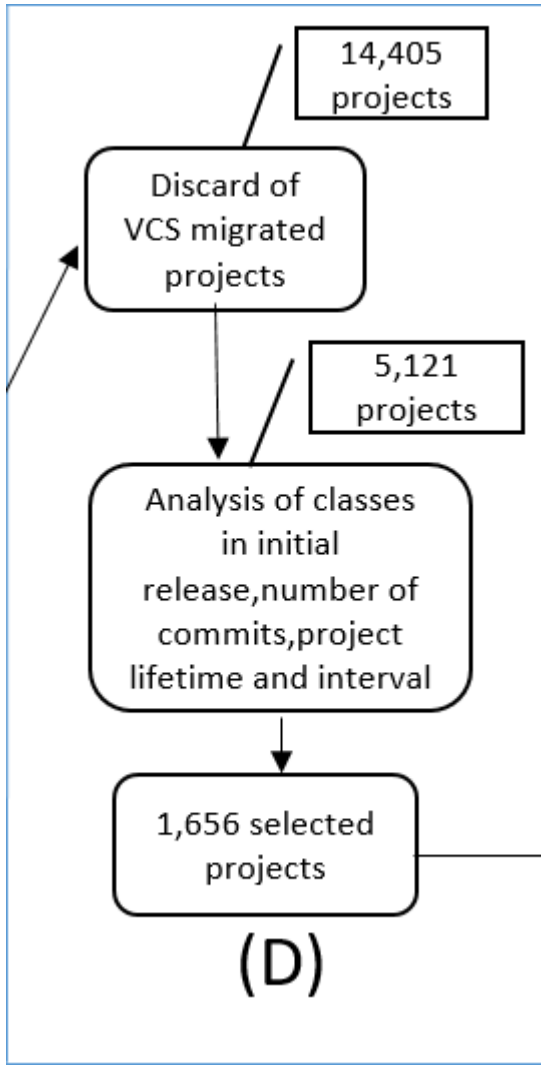
Tags
Autenticação
Duplicidade CVS,SVN,GIT



start, initial, test, before, beta, alpha, pre, demo, old, init, none, dev, example, first import, experimental, hello world, inicio, readme,first commit,RC[0..9] e CR[0..9].



50% do total de classes nos 20 primeiros commits

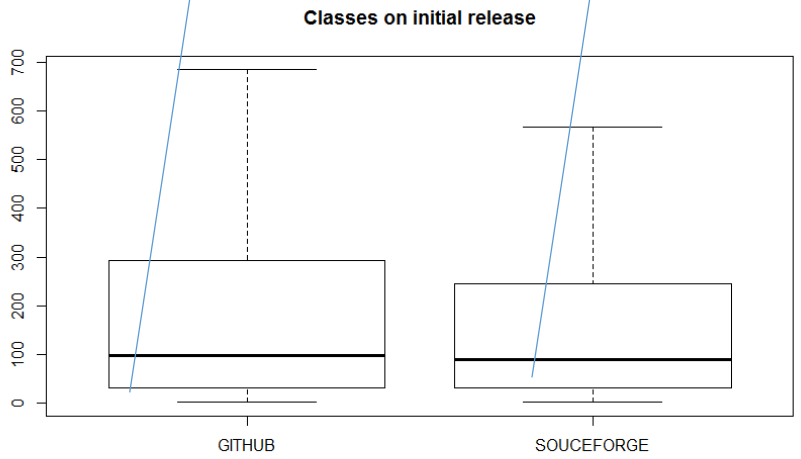


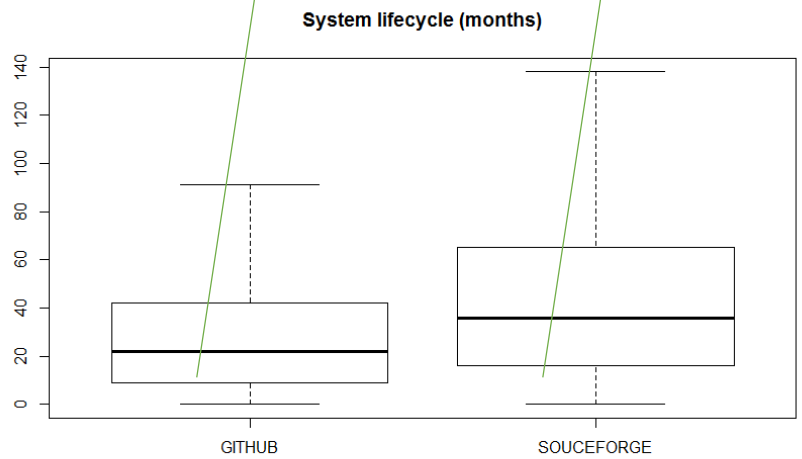
32 classes

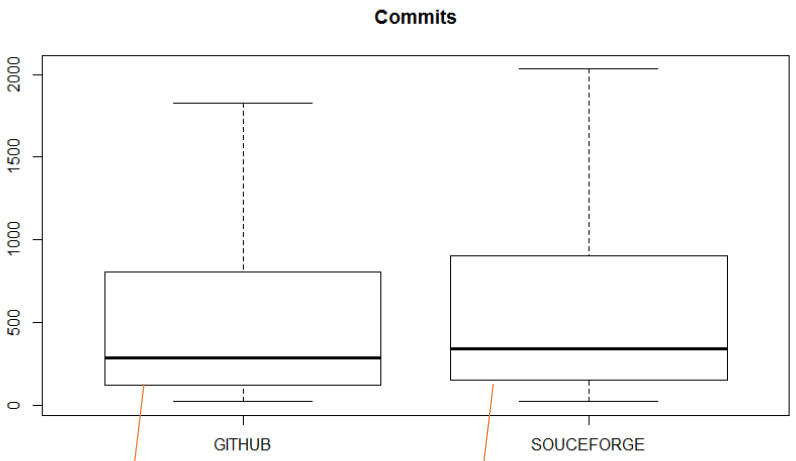
32 classes

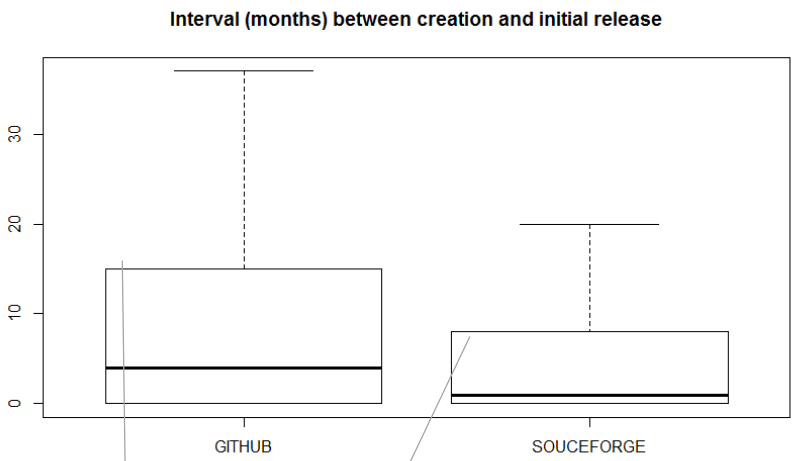
9 meses

16 meses







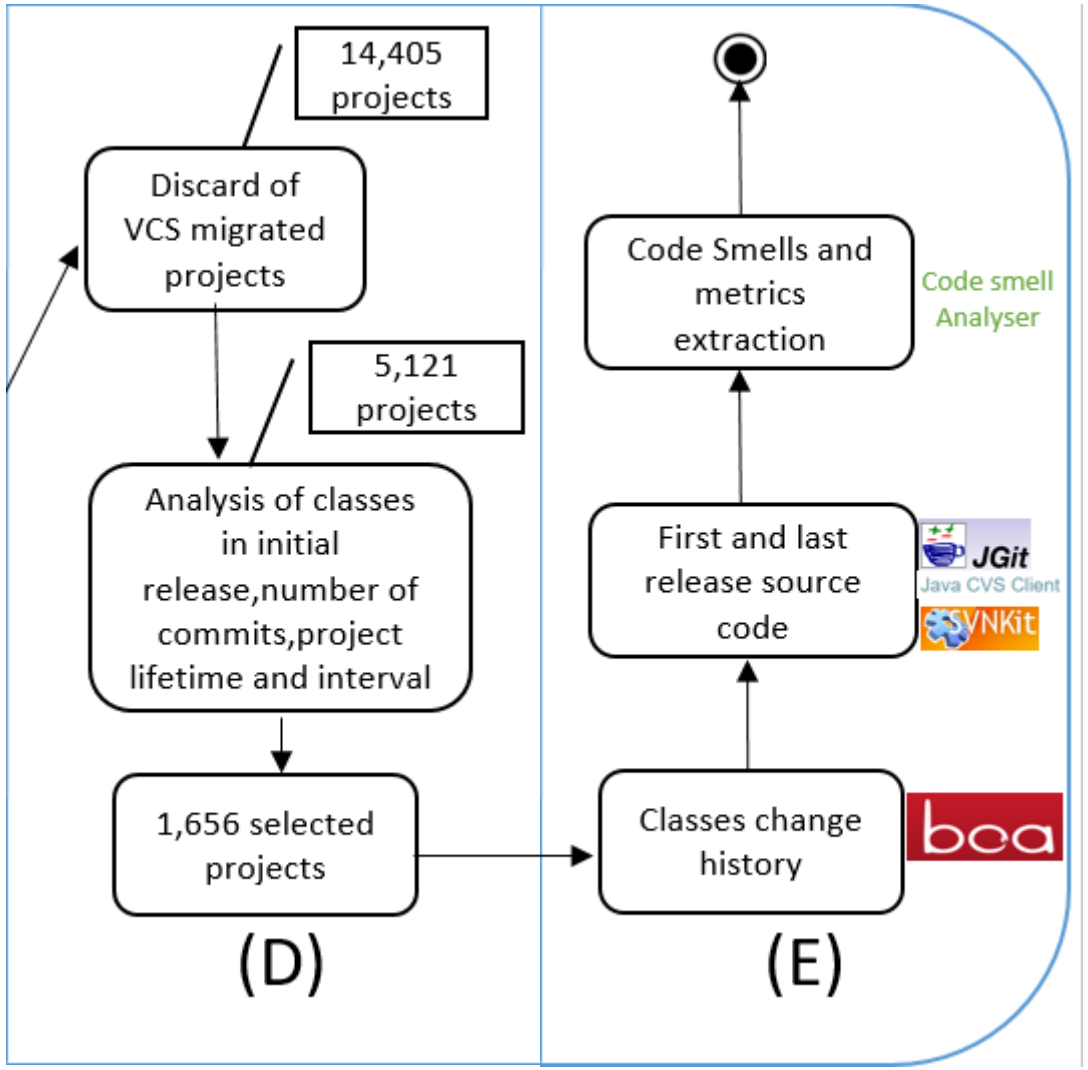


122 commits

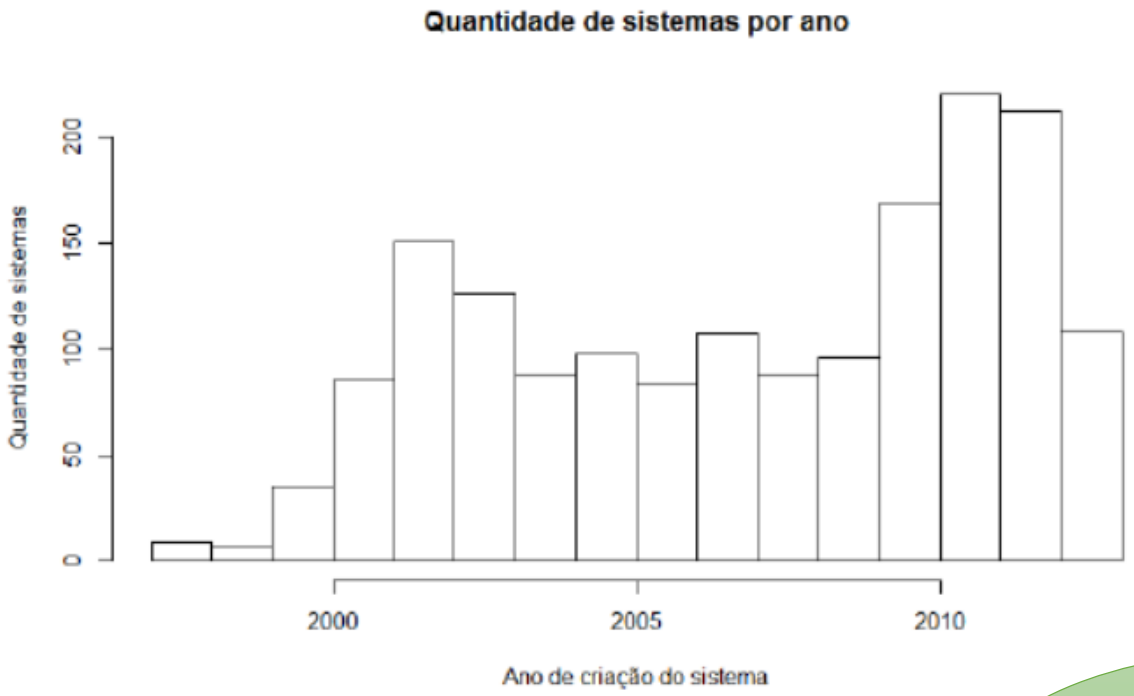
150 commits

15 meses

8 meses


14,405 projects
Discard of VCS migrated projects
5,121 projects
Analysis of classes in initial release,number of commits,project lifetime and interval
1,656 selected projects
(D)
Classes change history
First and last release source code
Code Smells and metrics extraction
Code smell Analyser
JGit
Java CVS Client
VNKit
boa
(E)

Informações sobre os Sistemas



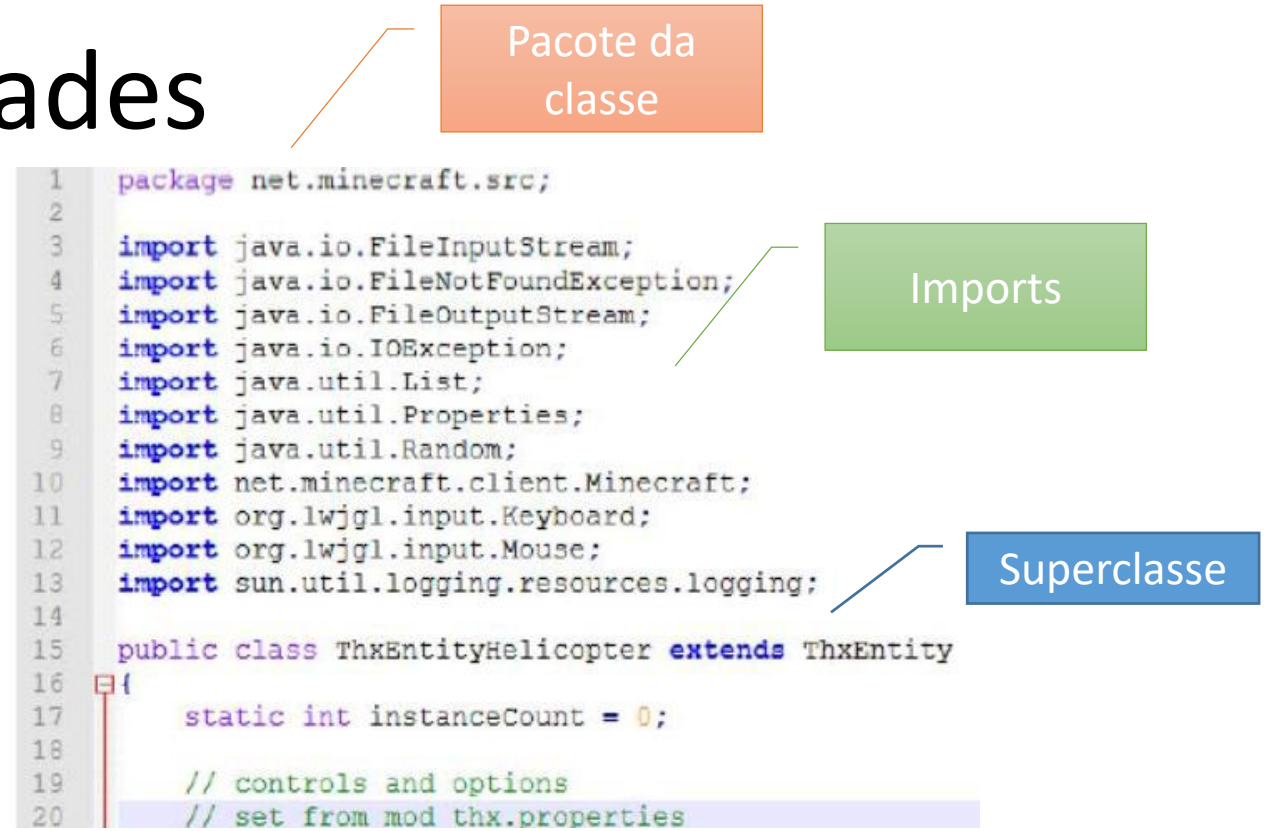
1.301.856 commits

1.297.199 classes & interfaces

1.656 sistemas

Modelagem dos dados

1) Consistência dos dados
2) Ligações entre entidades
3) Descartes
4) Geração de arquivos



Análise dos Dados – RQ #1

Arquitetura original do sistema

Influência da data da versão inicial do sistema sobre a quantidade de classes em herança ou implementando interfaces

$$\ln_{(INHER)} = INTERCEPT + \beta_1(CLASSES) + \beta_2(AGE)$$

$$\ln_{(INTER)} = INTERCEPT + \beta_1(CLASSES\_INTERFACE) + \beta_2(AGE)$$

Regressão Binomial Negativo (mesmo modelo da regressão de Poisson)

Análise dos Dados – RQ #1

Avaliar a distribuição da proporção de classes com herança e implementação de interface

$$FATOR\_INHERITANCE = \frac{CLASSES\_INHERITANCE}{CLASSES}$$

$$FATOR\_INTERFACE = \frac{SUBCLASSES\_INTERFACE}{CLASSES\_INTERFACE}$$

Análise dos Dados – RQ #2

Histórico de alterações sobre as classes

Influência da data de criação do sistema sobre a quantidade de subclasses referenciadas pelo operador *instanceof*

$$\ln_{(INSTANCEOF\_INHER)} = INTERCEPT + \beta_1(SUBCLASSES) + \beta_2(CREATION)$$

$$\ln_{(INSTANCEOF\_INTER)} = INTERCEPT + \beta_1(IMPLEMENTS) + \beta_2(CREATION)$$

Análise dos Dados – RQ #2

Avaliar a distribuição da proporção de classes em herança e implementação referenciados por instanceof

$$FATOR\_INHERITANCE = \frac{INSTANCEOF\_INHERITANCE}{SUBCLASSES}$$

$$FATOR\_INTERFACE = \frac{INSTANCEOF\_INTERFACE}{IMPLEMENTS}$$

Análise dos Dados – RQ #3

Histórico de alterações sobre as classes

Influência da data de criação do sistema sobre a quantidade de alterações corretivas sobre classes com e sem herança

$$\ln_{(FIXING\_INHER)} = INTERCEPT + \beta_1(FIXING) + \beta_2(CREATION)$$

$$\ln_{(FIXING\_NOINHER)} = INTERCEPT + \beta_1(FIXING) + \beta_2(CREATION)$$

$$\ln_{(FIXING\_INTER)} = INTERCEPT + \beta_1(FIXING) + \beta_2(CREATION)$$

$$\ln_{(FIXING\_NOINTER)} = INTERCEPT + \beta_1(FIXING) + \beta_2(CREATION)$$

Análise dos Dados – RQ #3

Avaliar a distribuição da proporção de classes em herança e implementação de interfaces que possuem mudanças corretivas

$$FATOR\_INHERITANCE = \frac{FIXING\_INHERITANCE}{FIXING}$$

$$FATOR\_INTERFACE = \frac{FIXING\_INTERFACE}{FIXING}$$

Análise dos Dados – RQ #4

Arquiteturas original e final do sistema

Comparação das classes para cada métrica (teste *Mann-Whitney U*)

- Com e sem herança
- Com e sem implementação de interfaces

Análise dos Dados – RQ #5

Arquiteturas original e final do sistema

1) Construção de duas tabelas de contingência (versões inicial e final)
   Com e sem herança/implementação de interfaces
   Com e sem a ocorrência de determinado Code Smell

2) Aplicações dos testes *Fisher* e *Qui Quadrado*

Análise dos Dados – RQ #6

Histórico de alterações sobre as classes

1) Análise do Histórico de Commits das classes
   a) Sempre
   b) Perdeu
   c) Depois
   d) Nunca

Análise dos Dados – RQ #6

2) Análise qualitativa sobre as classes que adicionaram e perderam herança e interface

2.1) Seleção de 40 alterações em 40 sistemas.

2.2) Diagramas de classe e avaliação das mensagens de commit.

2.3) Codificação dos temas para cada alteração

2.4) Hierarquização dos temas

ROTEIRO

1. Contextualização do Problema.
2. Objetivos.
3. Metodologia.
4. **Resultados.**
4. Ameaças à validade.
5. Conclusão.

Resultados – RQ #1

$e^{0,0020224} = 1,002$

Pr(Chi) = $6.462469e^{-05}$

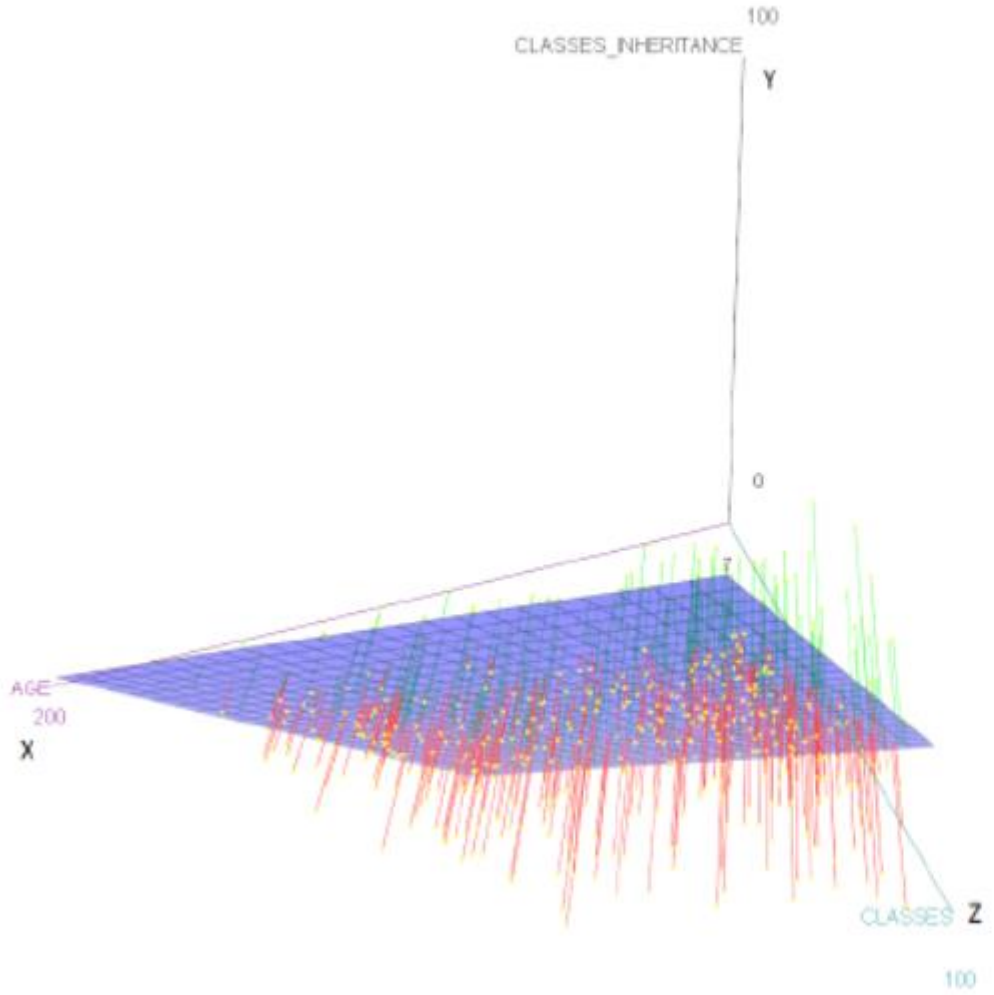
	Estimate	Std. Error	z value	Pr(> |z|)
(Intercept)	3,0155966	0,0512405	58.852	< 2e-16 ***
CLASSES	0,0032857	0,0000574	57.241	< 2e-16 ***
AGE	0,0020224	0,0005005	4.041	5.32e-05 ***

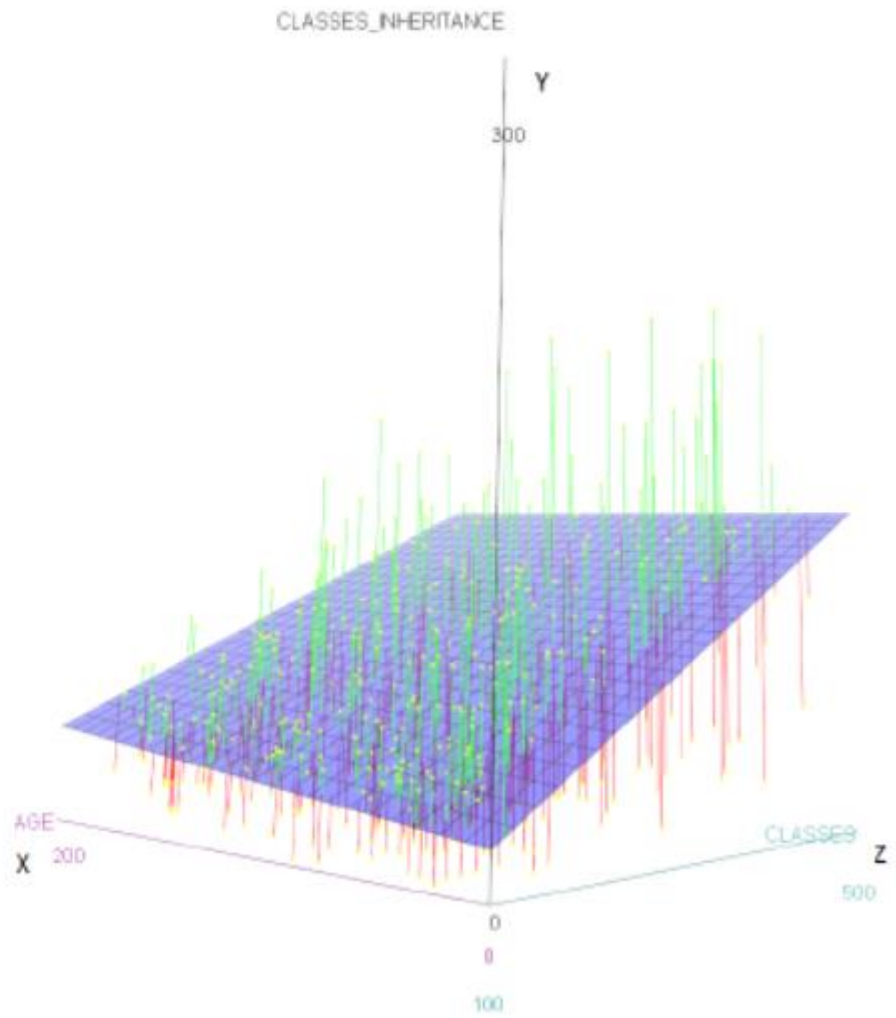
	Estimate	Std. Error	z value	Pr(> |z|)
(Intercept)	2,704e+00	5,602e-02	48.266	< 2e-16 ***
CLASSES_INTERFACE	2,880e-03	5,553e-05	51.857	< 2e-16 ***
AGE	2,240e-03	5,472e-04	4.094	4.25e-05 ***

Pr(Chi) = $6.326497e^{-05}$

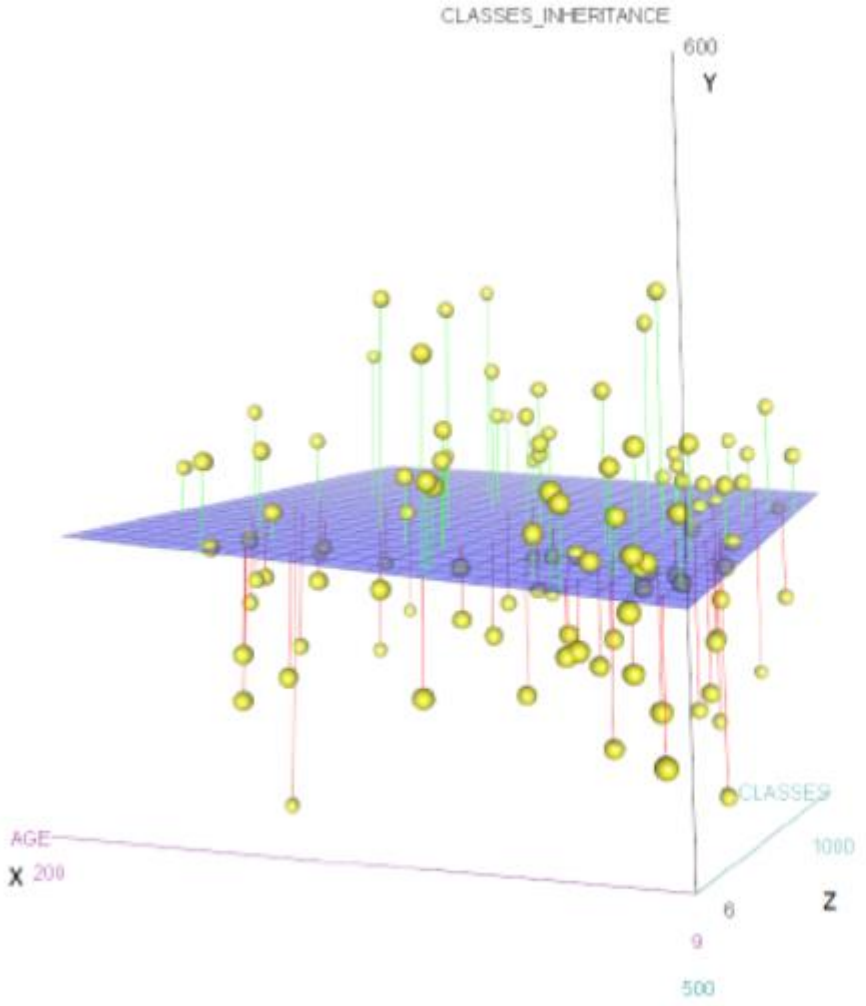
$e^{2,240e-03} = 1,002$

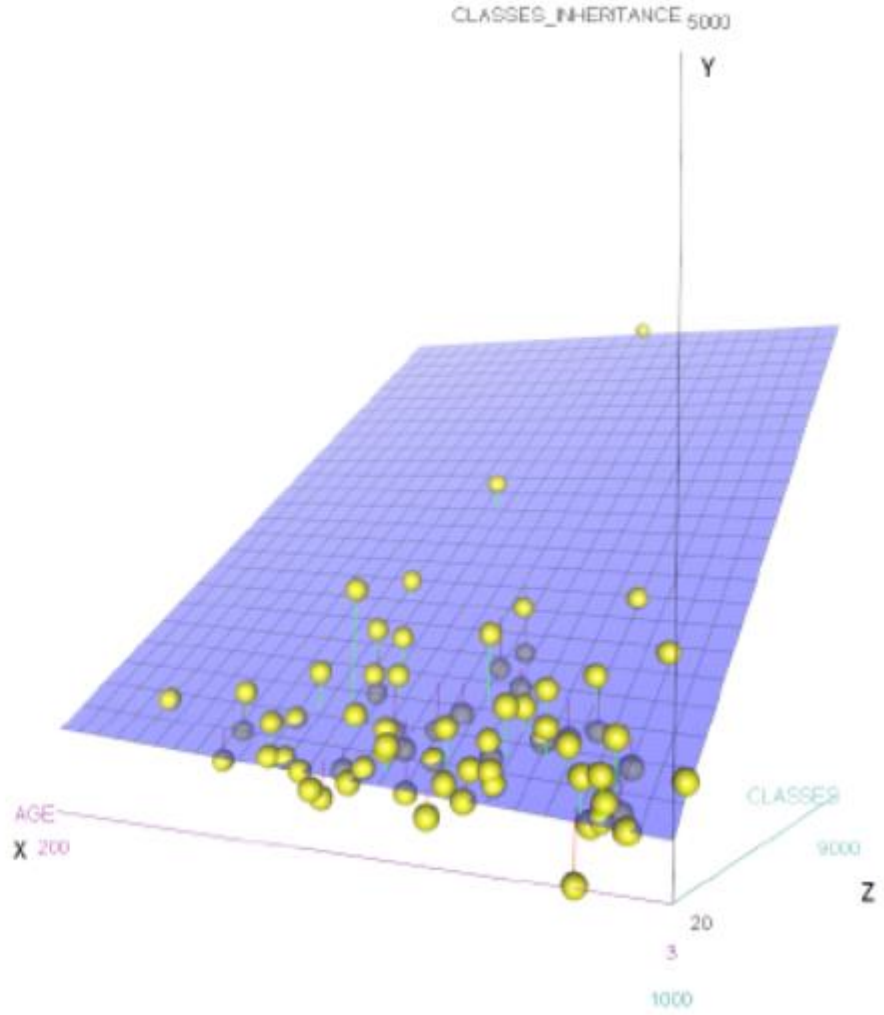
Resultados – RQ #1



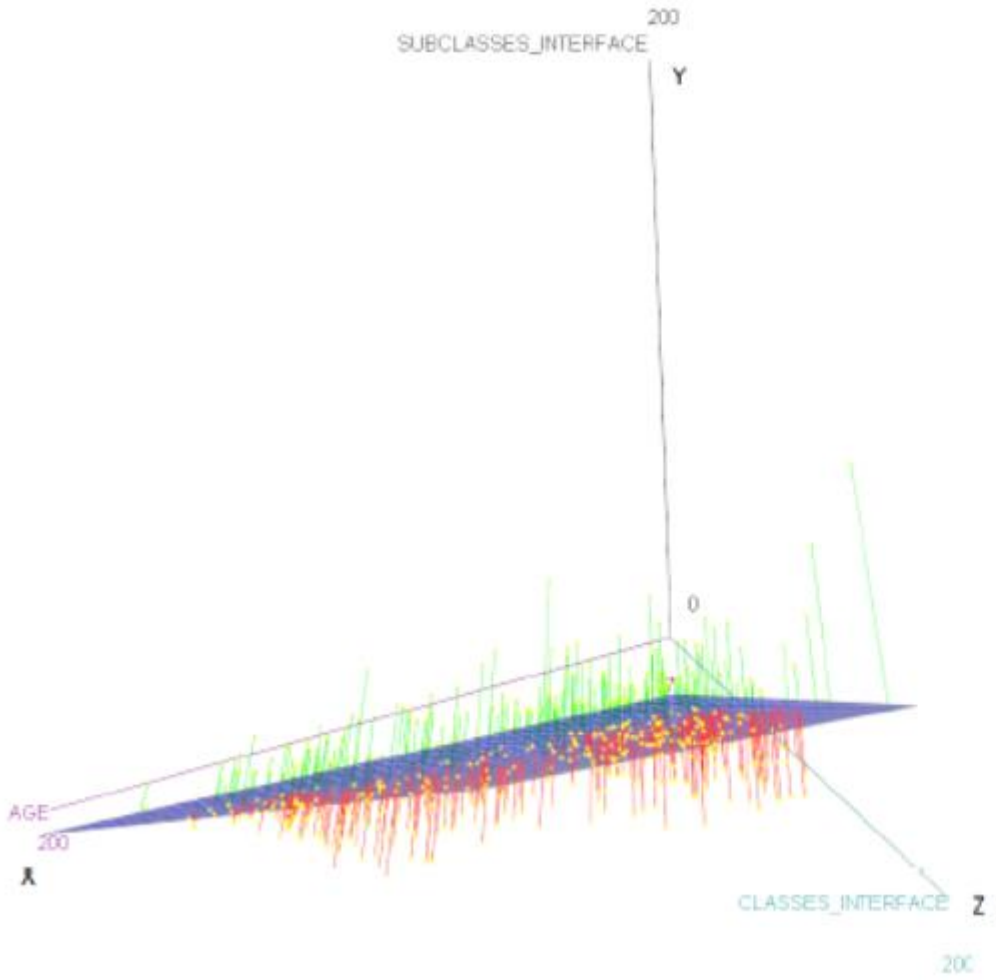


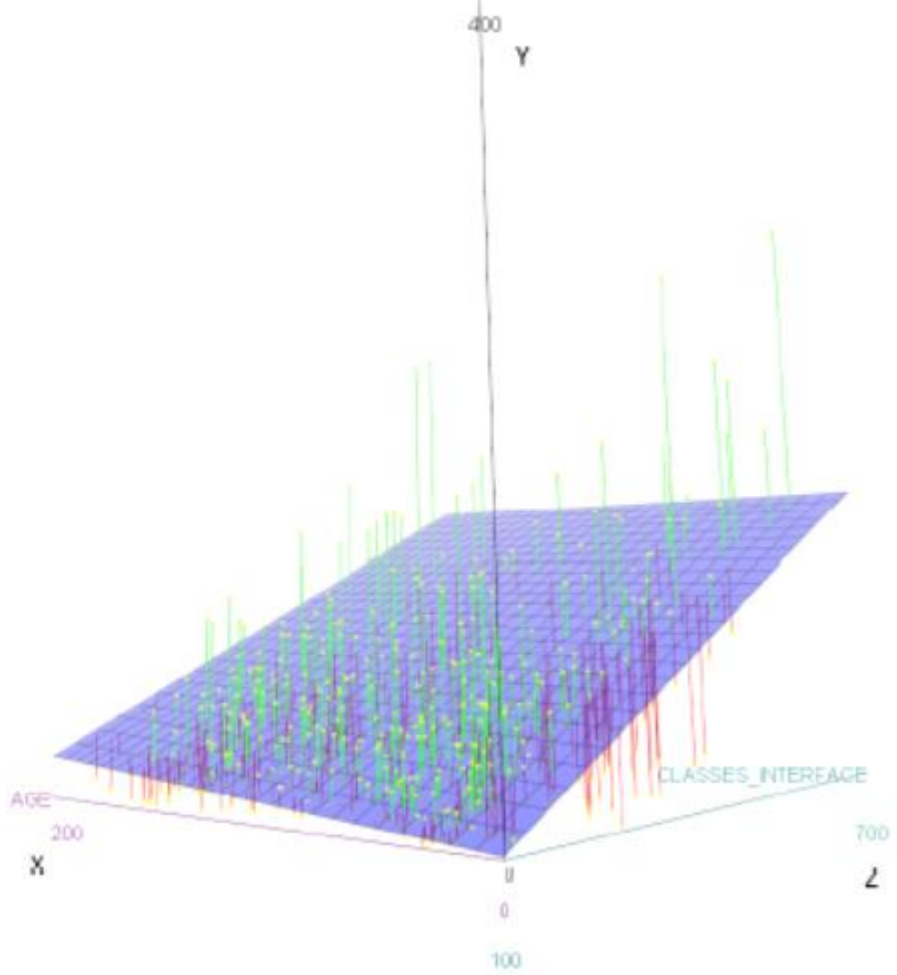
Resultados – RQ #1



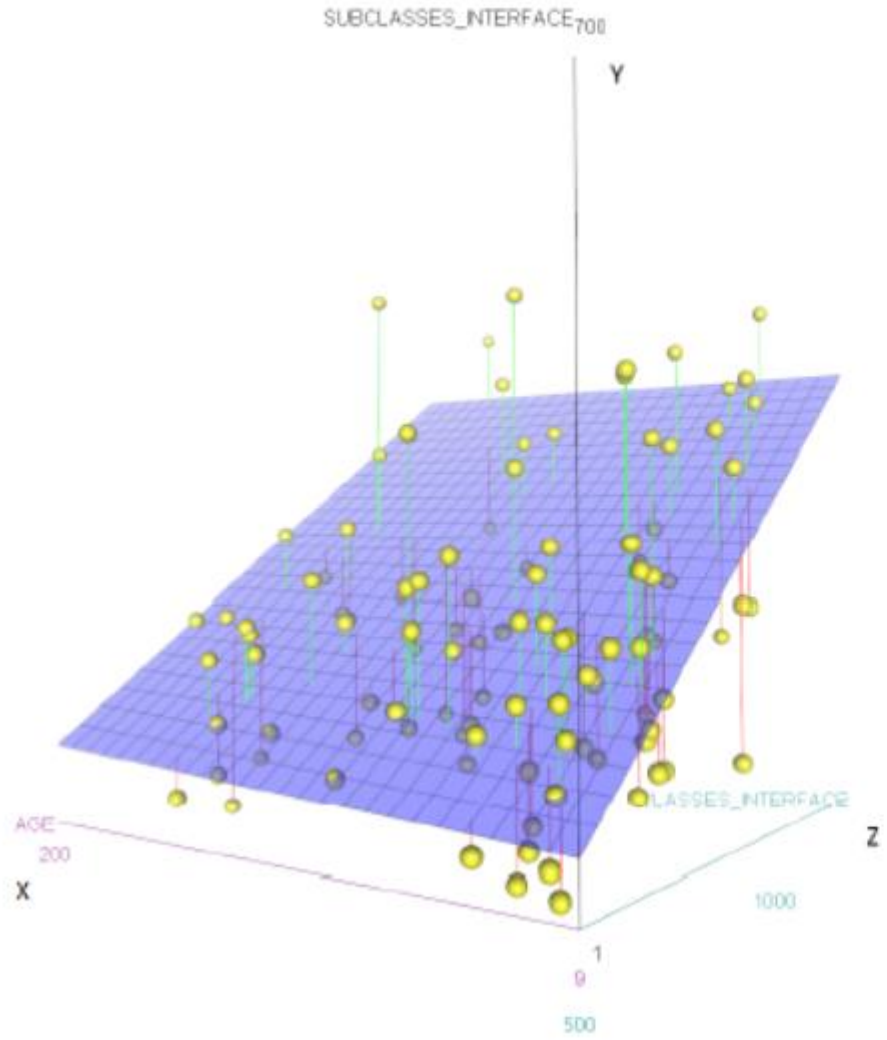


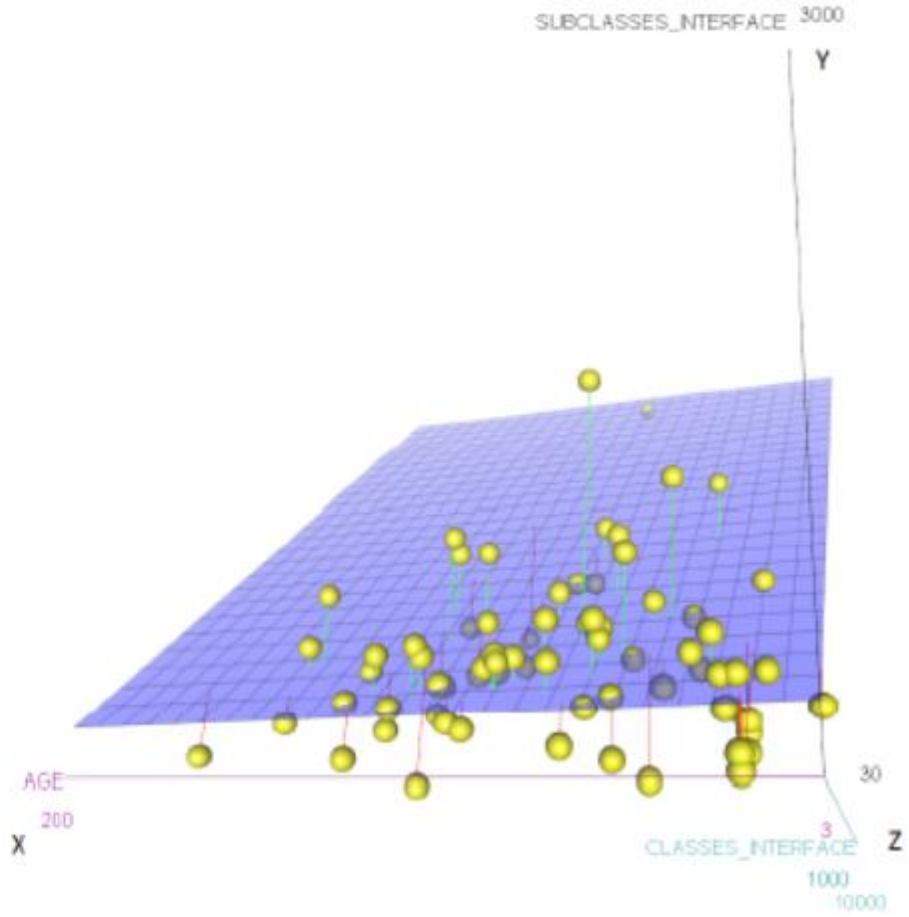
Resultados – RQ #1





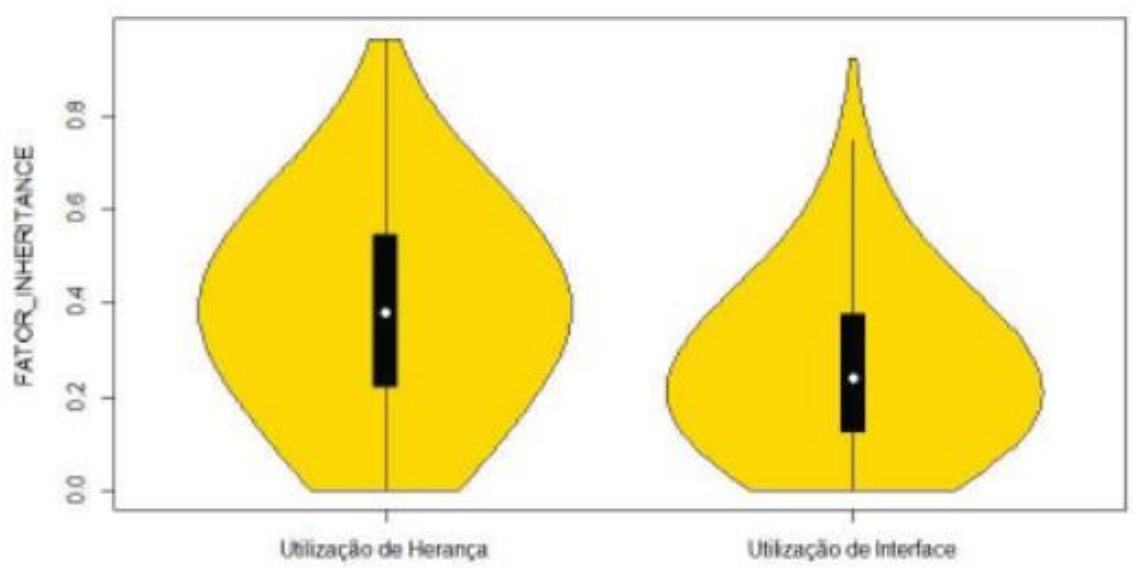
Resultados – RQ #1

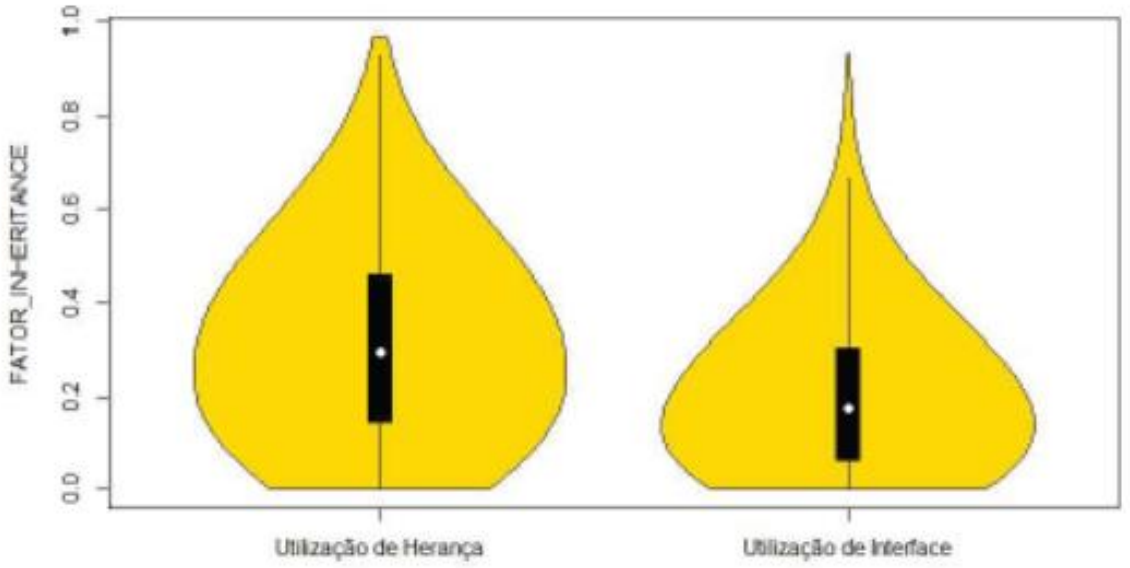




Resultados – RQ #1

Março-2009





Resultados – RQ #2

Pr(Chi) = $2.220446e^{-16}$

$e^{7,282e-03} = 1,007$

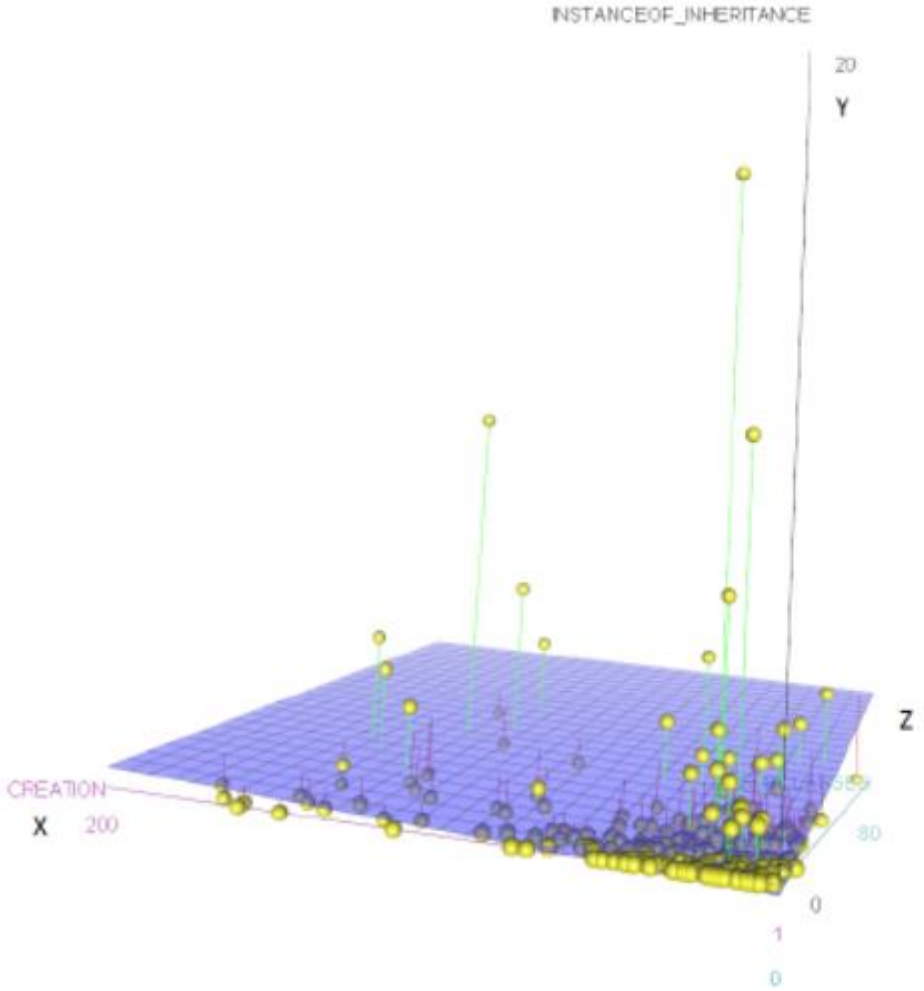
	Estimate	Std. Error	z value	Pr(> |z|)
(Intercept)	1,059e+00	7,283e-02	14.543	<2e-16 ***
SUBCLASSES	2,685e-03	6,663e-05	40.304	<2e-16 ***
CREATION	7,282e-03	8,518e-04	8.549	<2e-16 ***

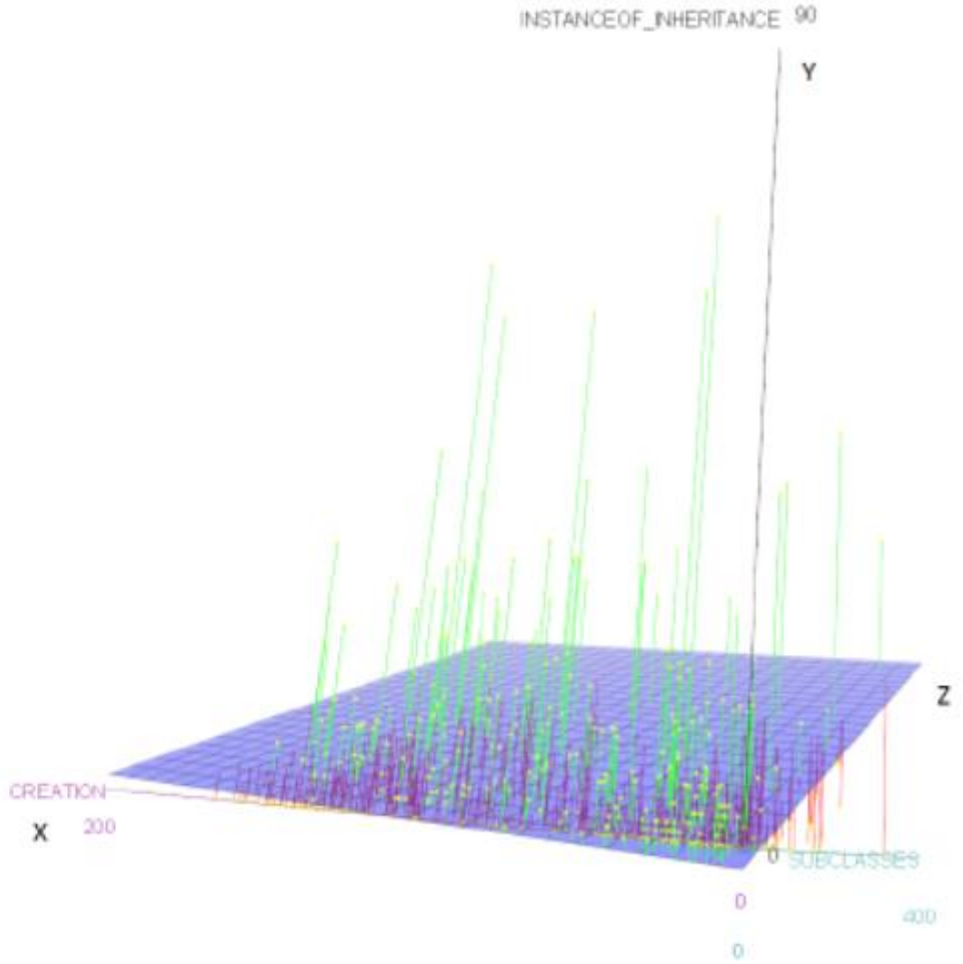
	Estimate	Std. Error	z value	Pr(> |z|)
(Intercept)	5,139e-01	7,177e-02	7.161	8,02e-13 ***
IMPLEMENTS	4,421e-03	9,317e-05	47.453	< 2e-16 ***
CREATION	8,888e-03	8,278e-04	10.737	< 2e-16 ***

Pr(Chi) = 0

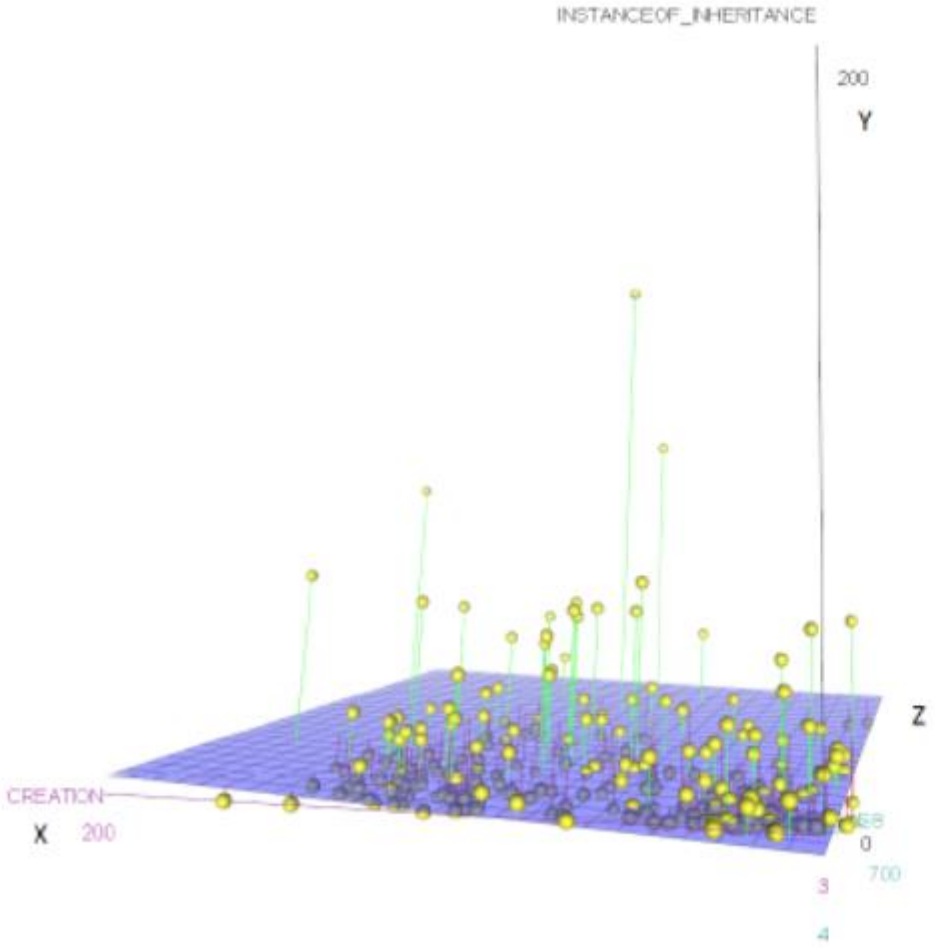
$e^{8,888e-03} = 1,008$

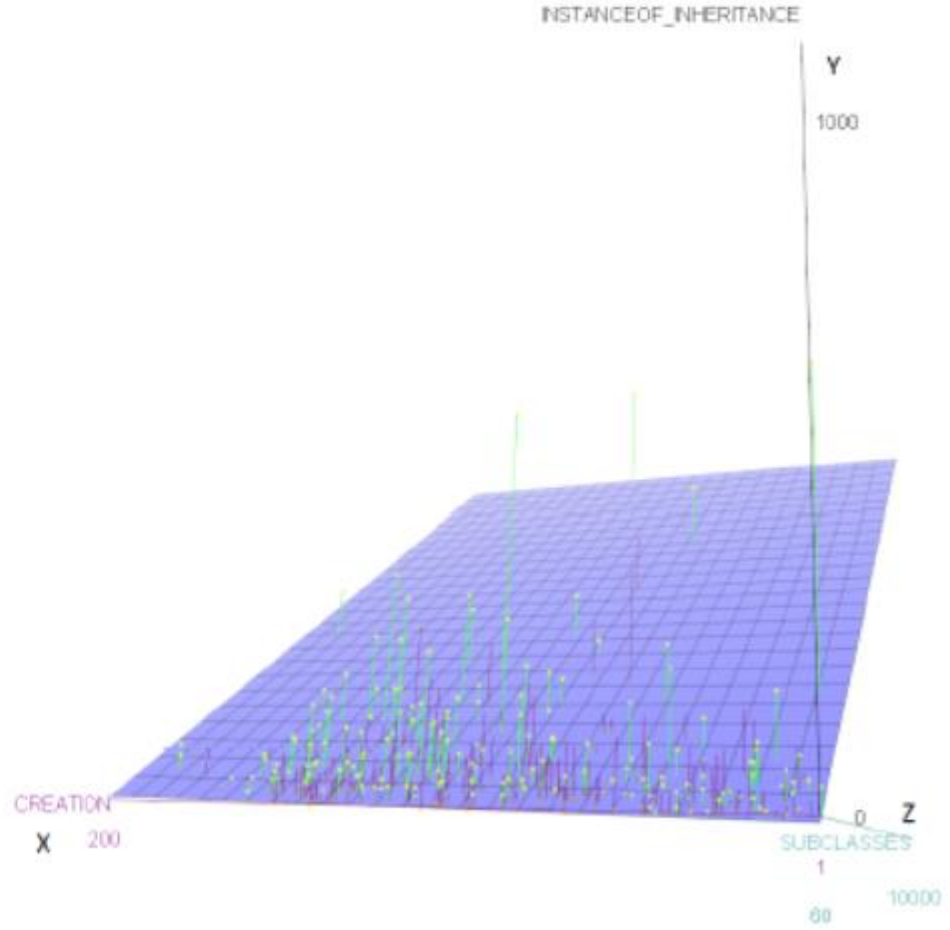
Resultados – RQ #2



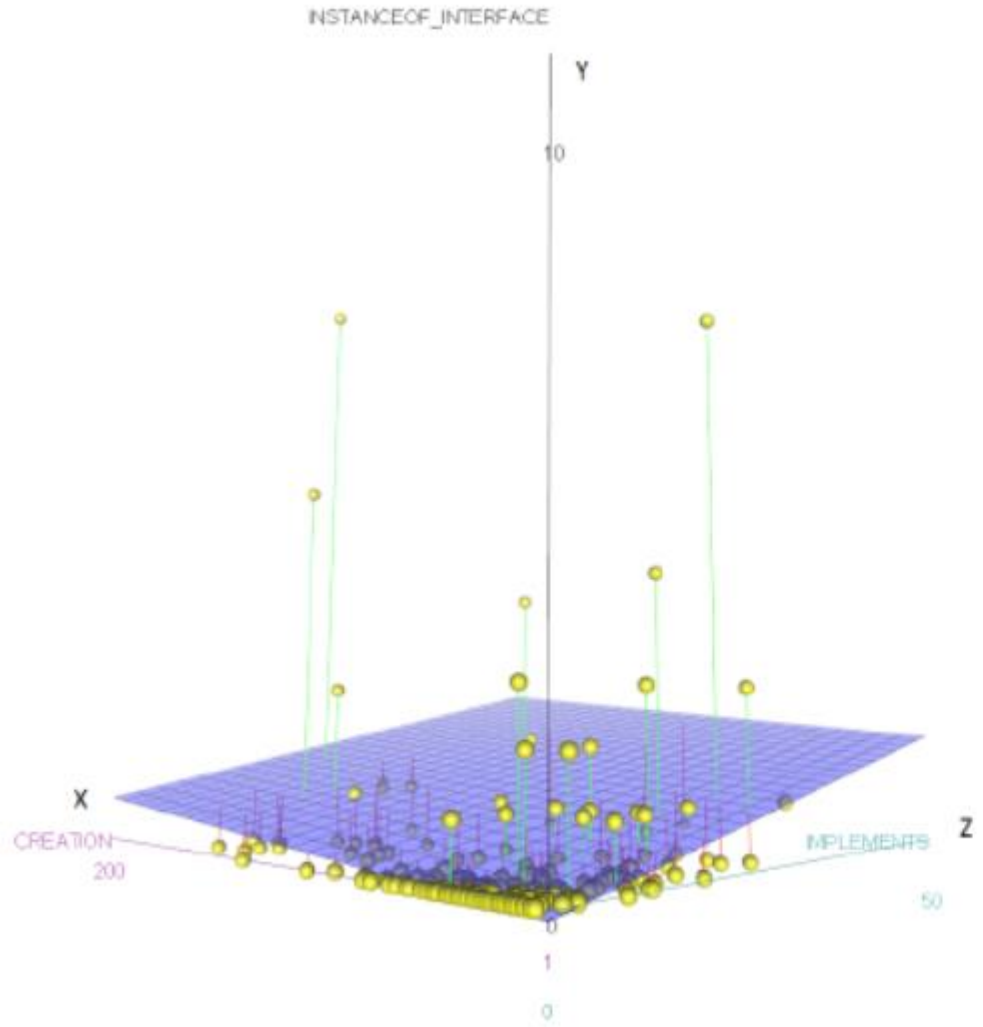


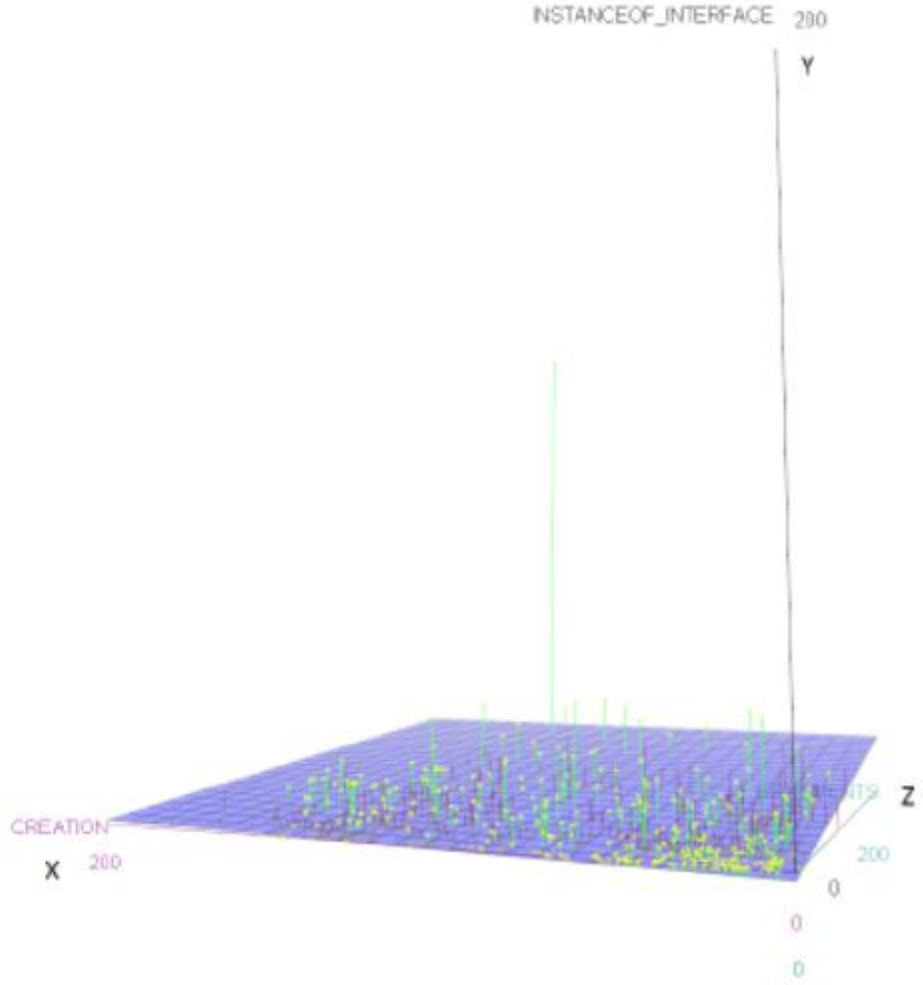
Resultados – RQ #2



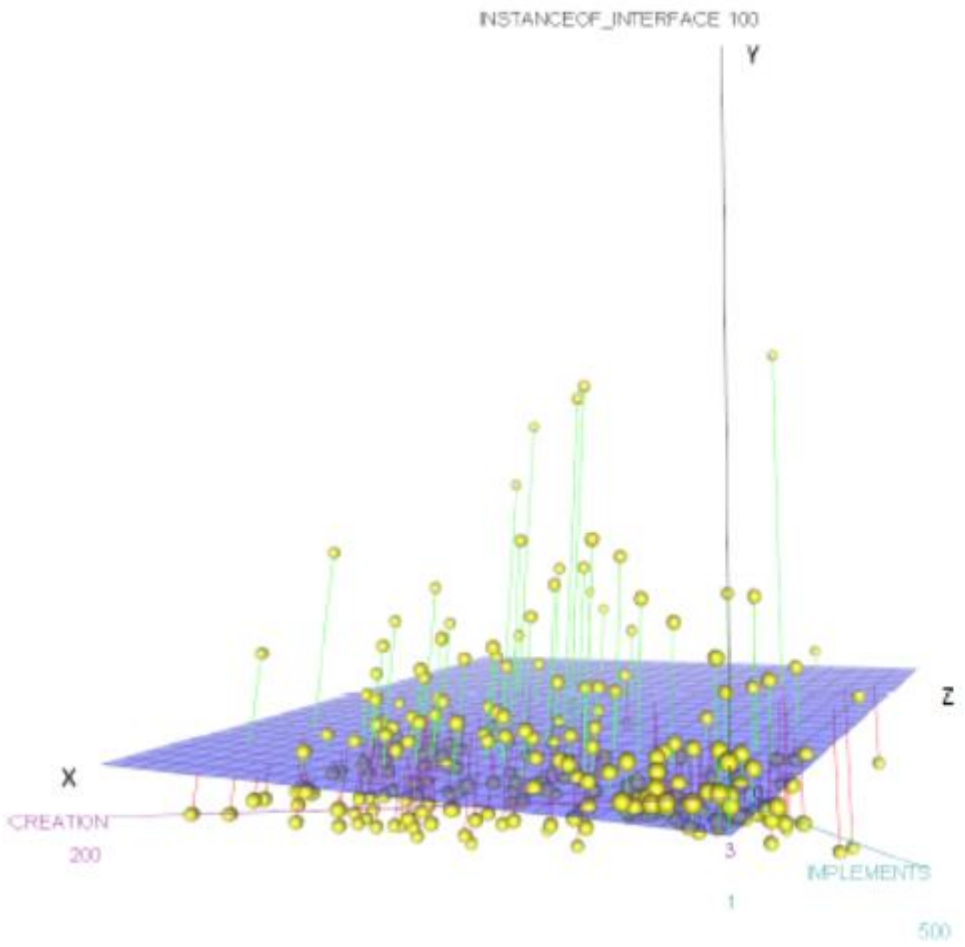


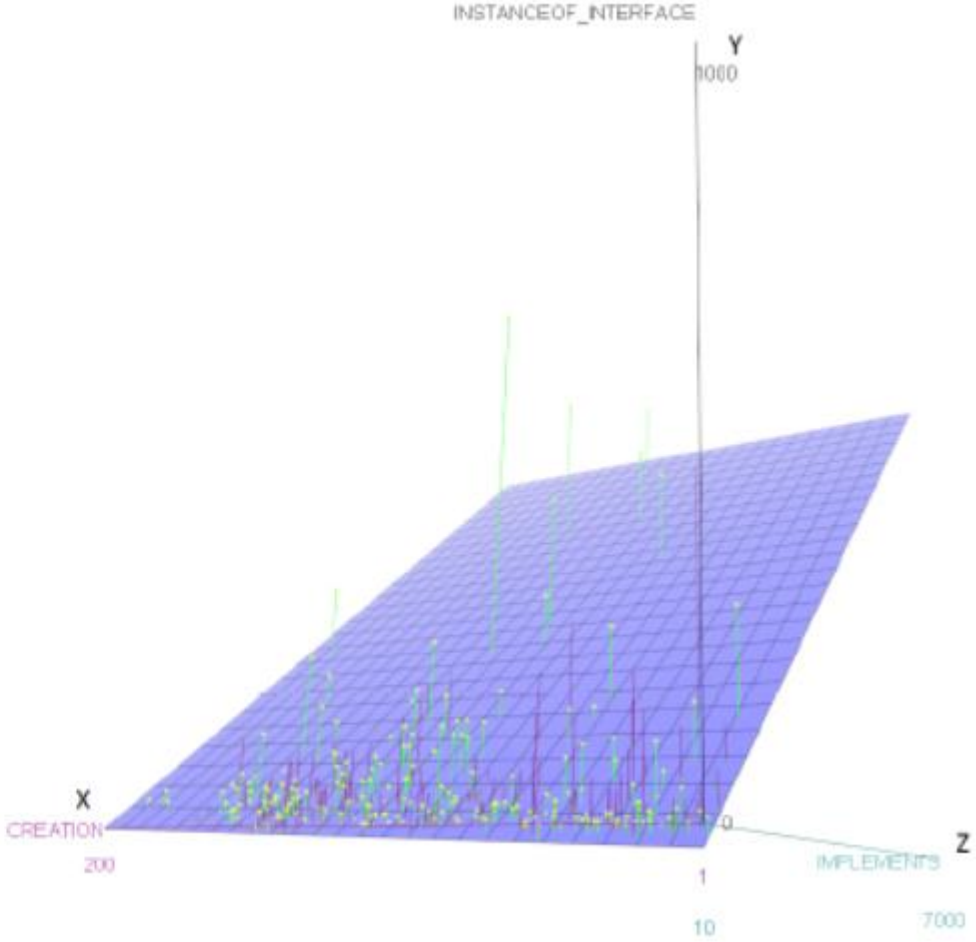
Resultados – RQ #2





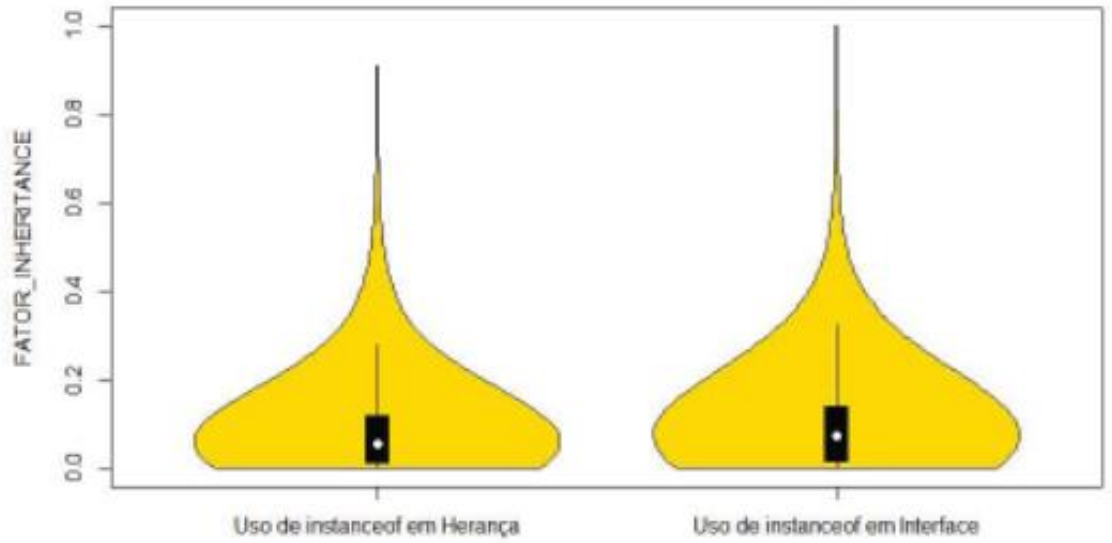
Resultados – RQ #2

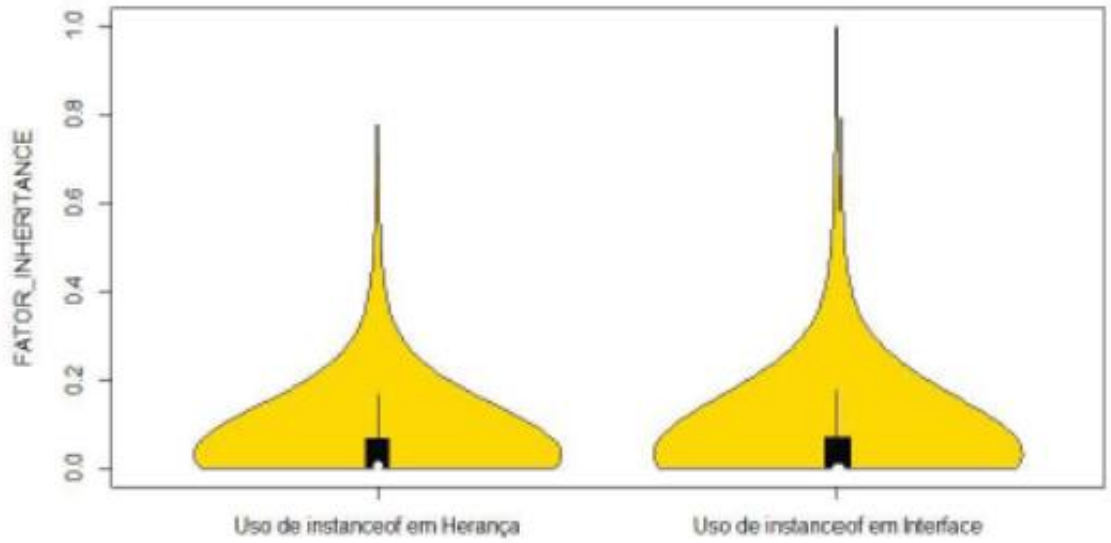




Resultados – RQ #2

Julho-2008





Resultados – RQ #3

$e^{1,763e-03} = 1,001$

Pr(Chi) = 0.0016

Pr(Chi) = 0.0422

Pr(Chi) = 5.009351e-06

Pr(Chi) = 0.036

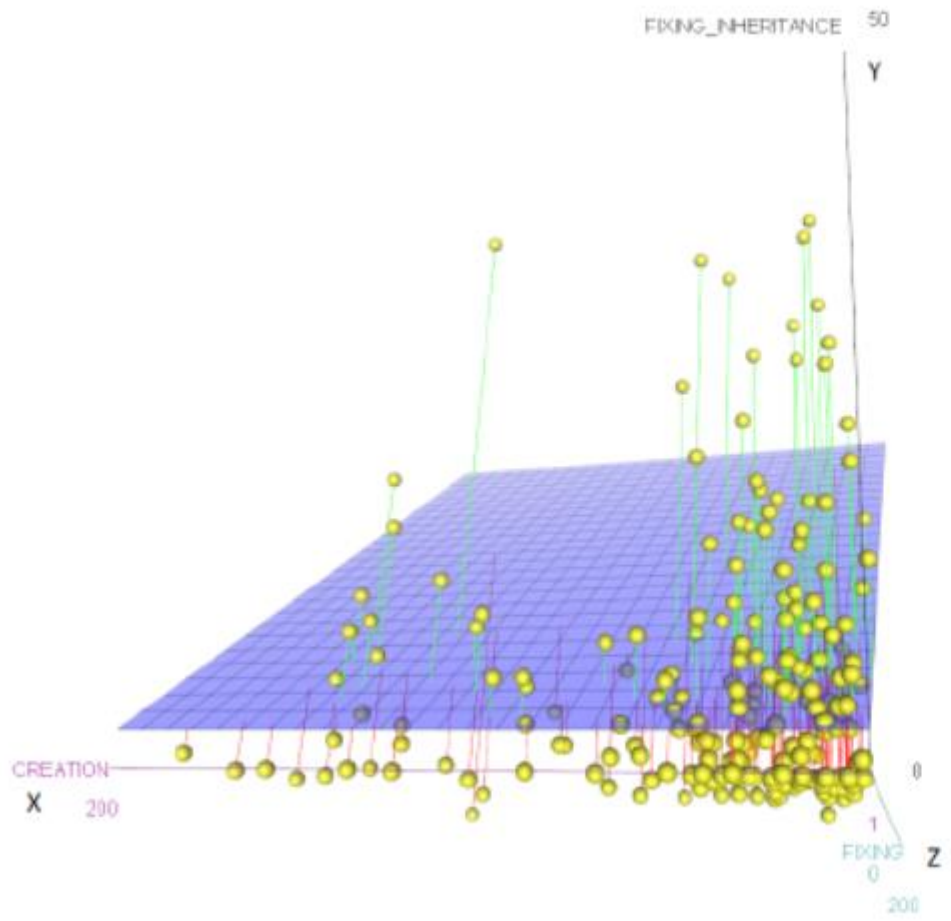
	Estimate	Std. Error	z value	Pr(> |z|)
Inheritance				
(Intercept)	2.737e+00	4.883e-02	56,048	< 2e-16 ***
FIXING	4.844e-03	8.149e-05	59,450	< 2e-16 ***
CREATION	1.763e-03	5.750e-04	3,067	0.00216 **
Without Inheritance				
(Intercept)	3.168e+00	4.086e-02	77.528	<2e-16 ***
FIXING	4.018e-03	6.817e-05	58.933	<2e-16 ***
CREATION	-9.758e-04	4.823e-04	-2.023	0.0431 *
Interface				
(Intercept)	2.357e+00	5.360e-02	43,970	<2e-16 ***
FIXING	4.606e-03	8.917e-05	51,660	<2e-16 ***
CREATION	2.832e-03	6.303e-04	4,494	7e-06 ***
Without Interface				
(Intercept)	3.352e+00	3.971e-02	84.416	<2e-16 ***
FIXING	4.317e-03	6.639e-05	65.032	<2e-16 ***
CREATION	-9.671e-04	4.687e-04	-2.064	0.0391 *

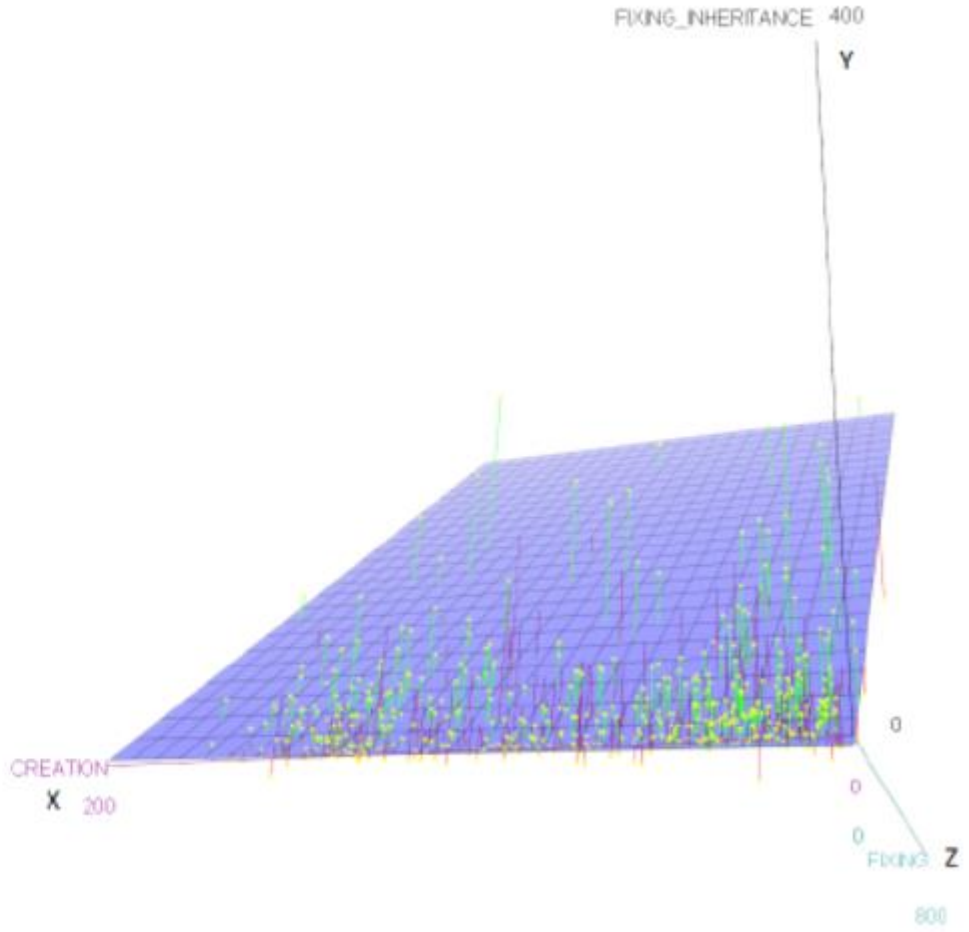
$e^{2832e-03} = 1,002$

$e^{-9671e-04} = -1,0009$

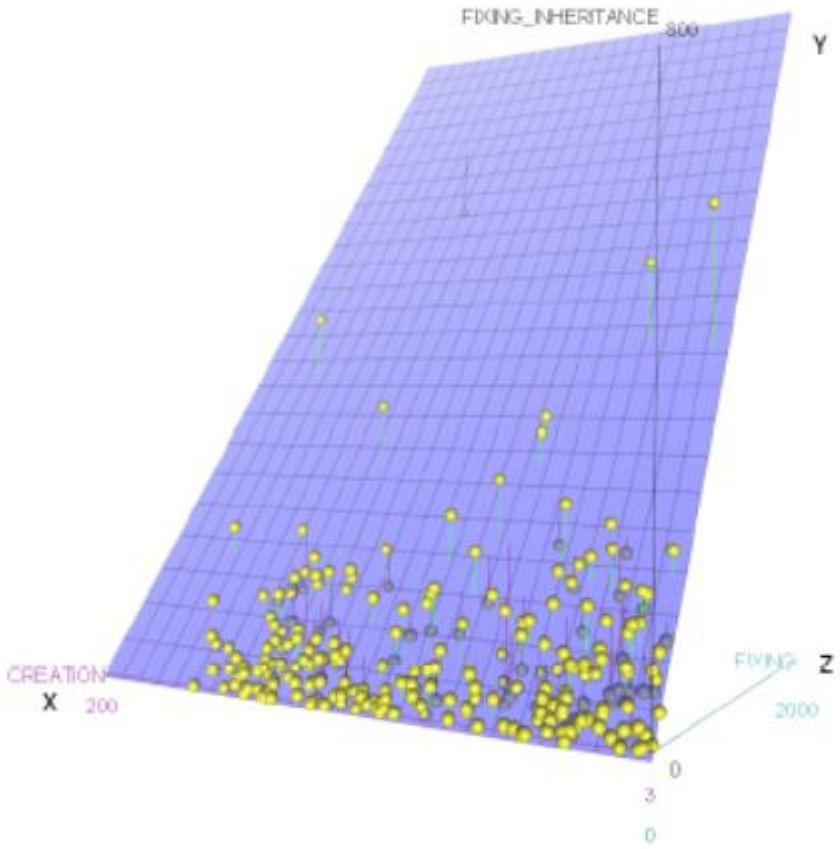
$e^{-9758e-04} = -1,0009$

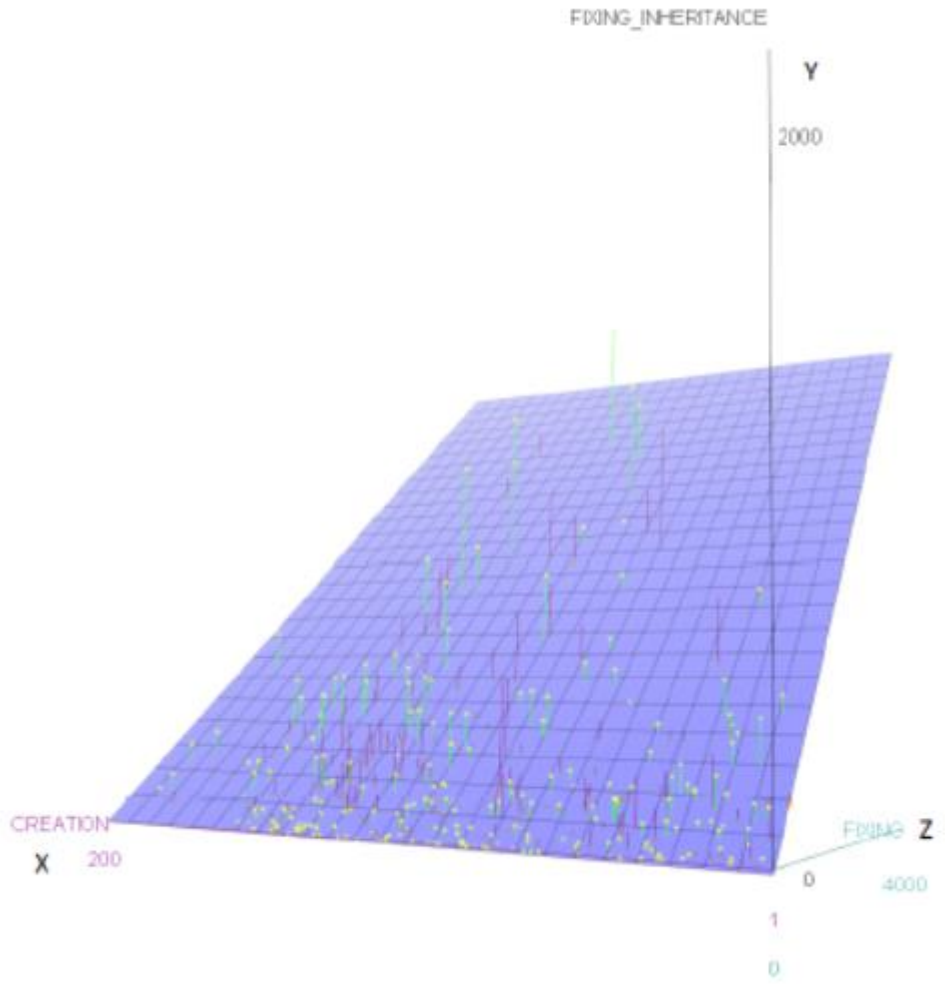
Resultados – RQ #3



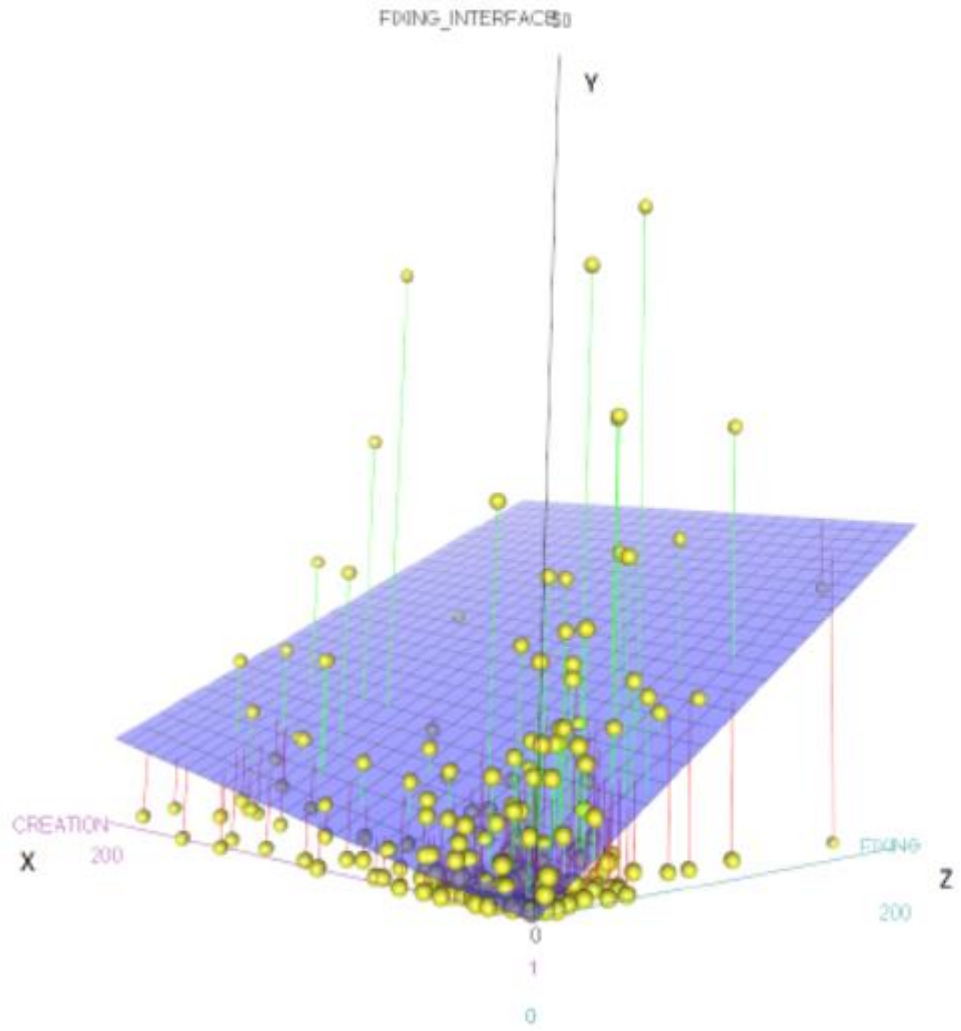


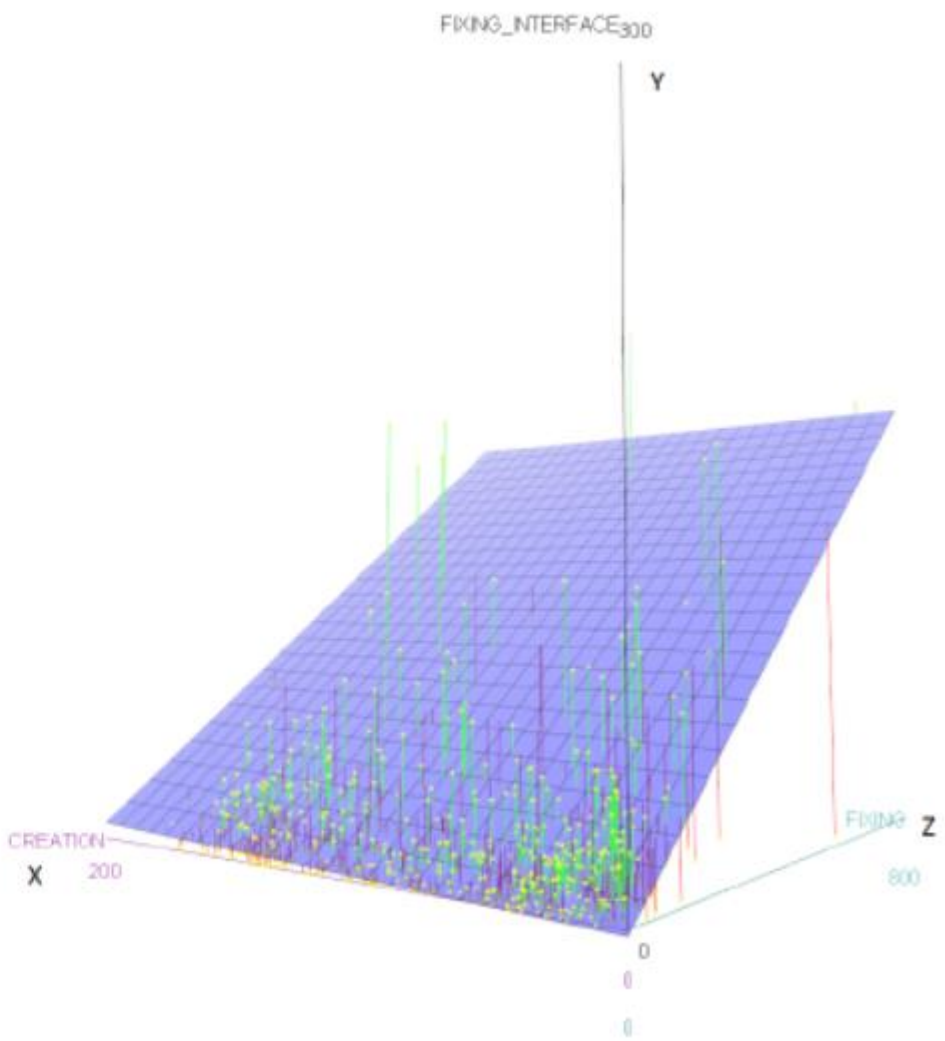
Resultados – RQ #3



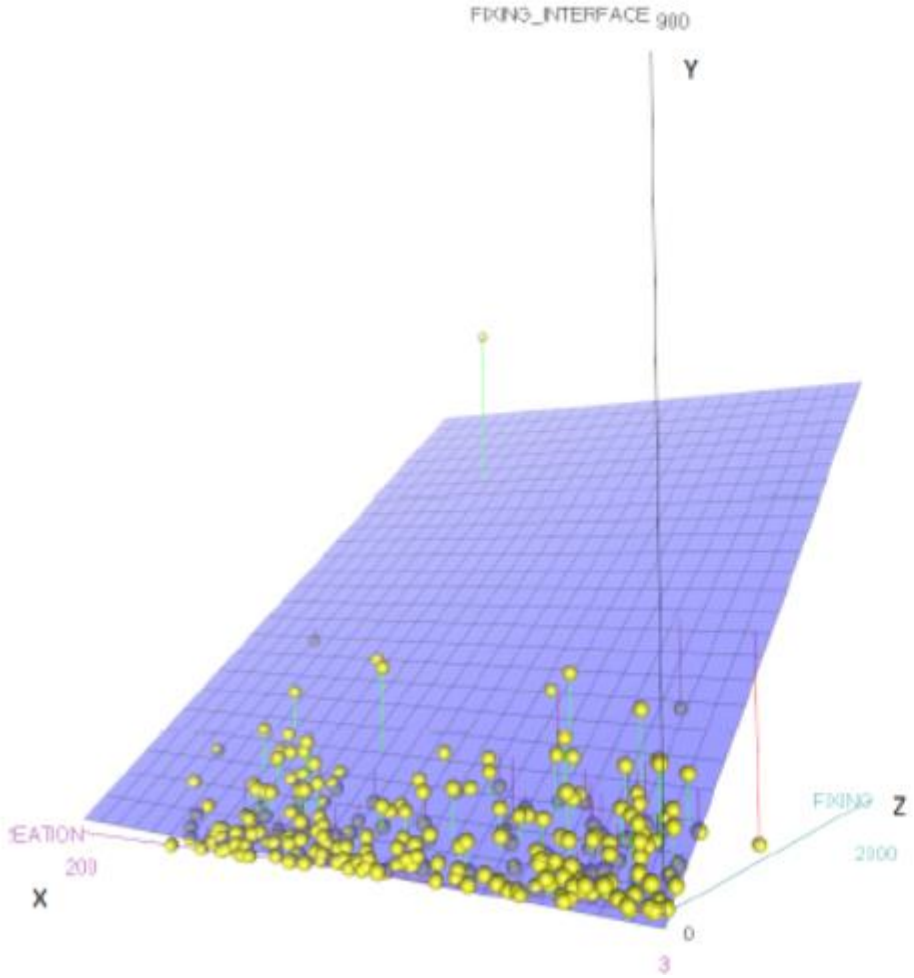


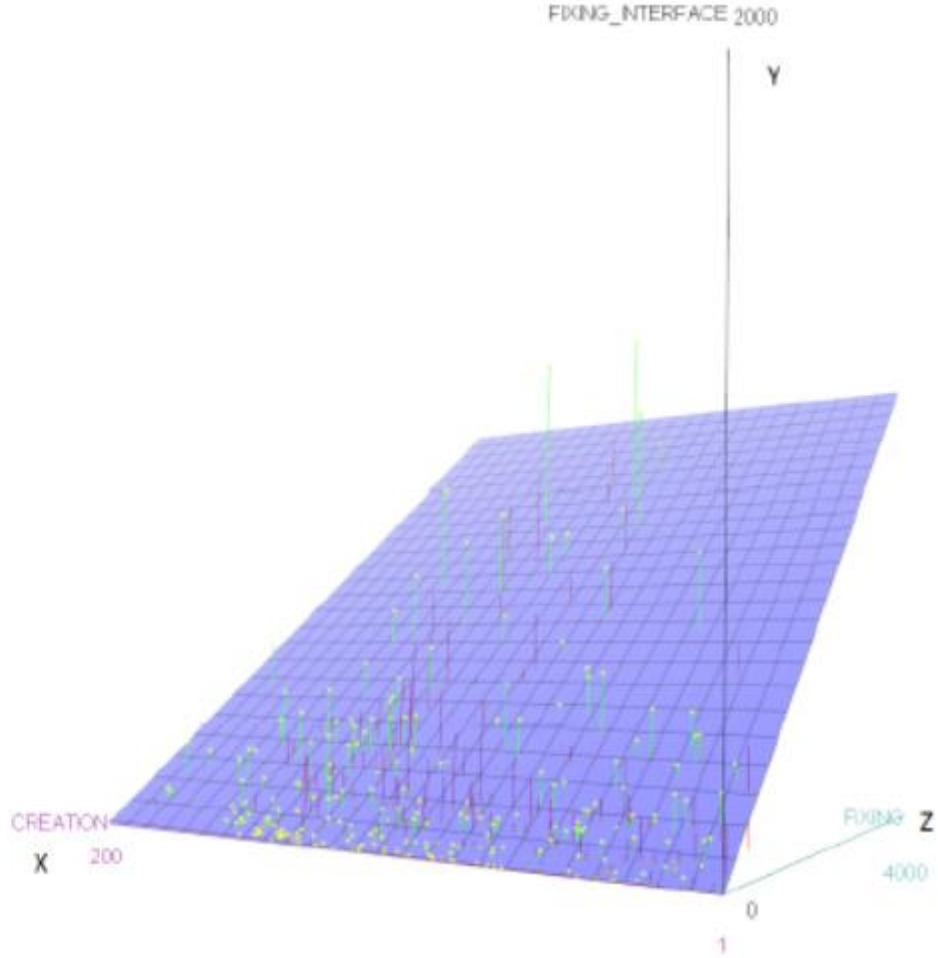
Resultados – RQ #3





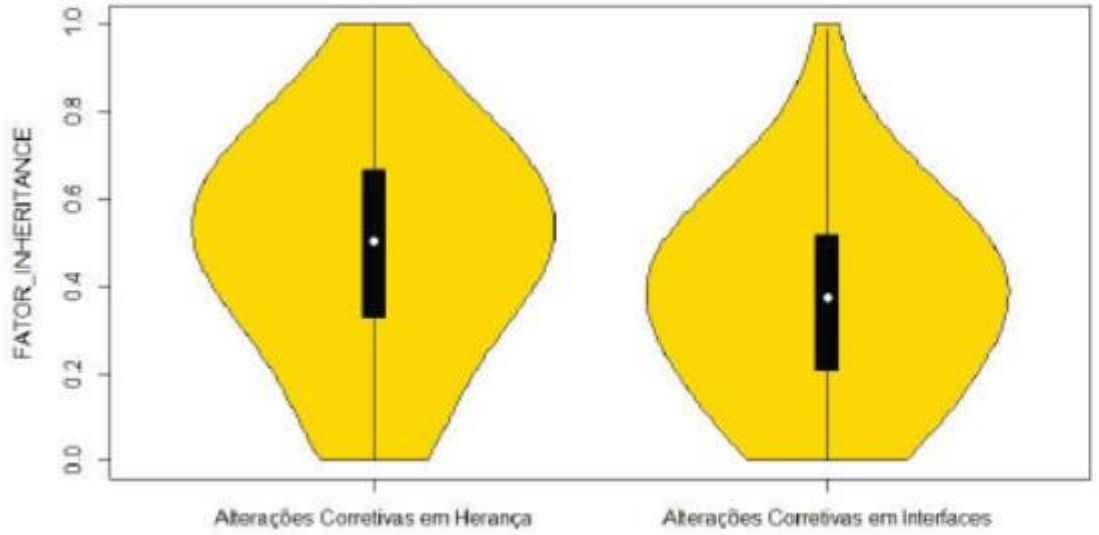
Resultados – RQ #3

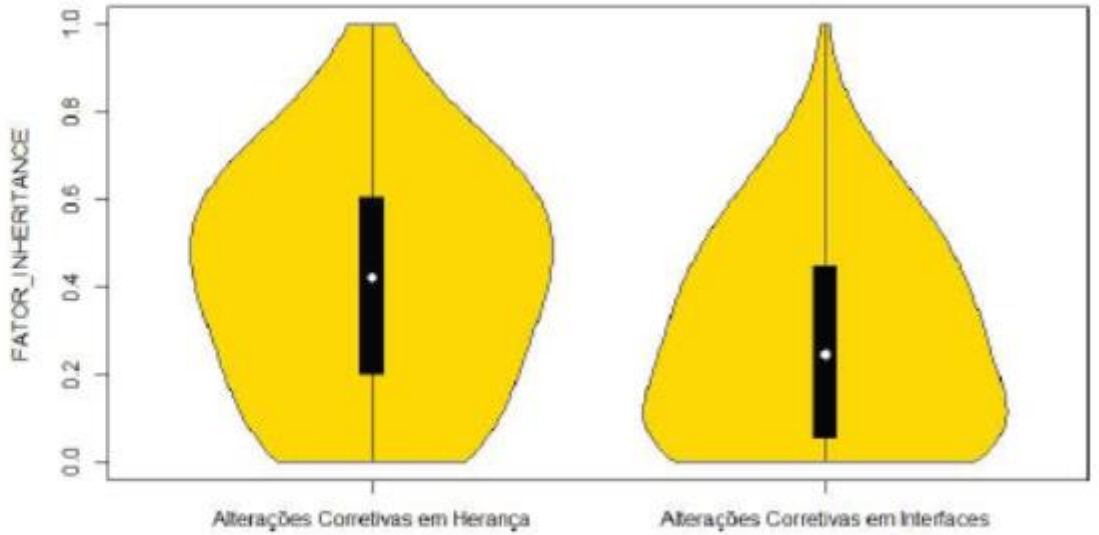




Resultados – RQ #3

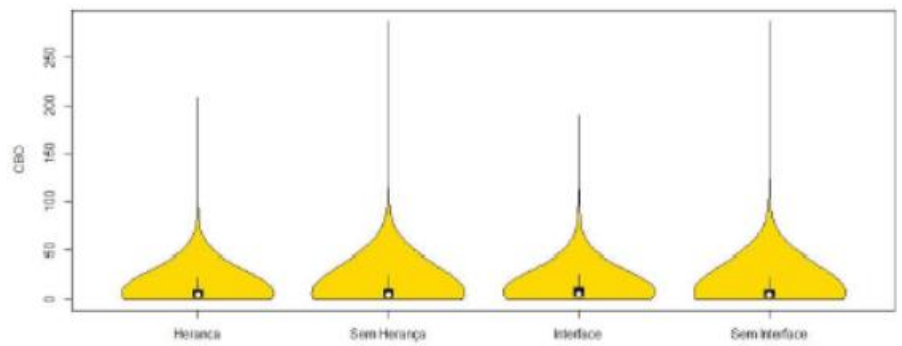
Julho-2008

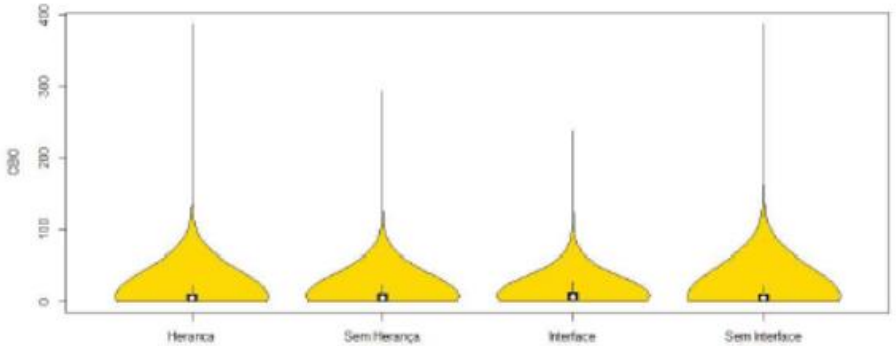




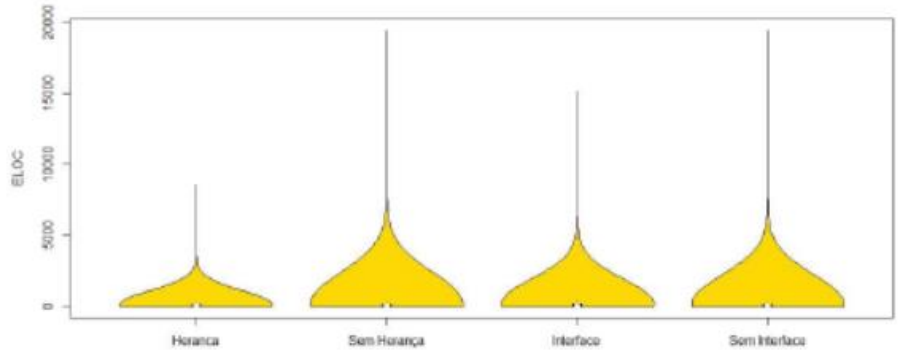
Resultados – RQ #4

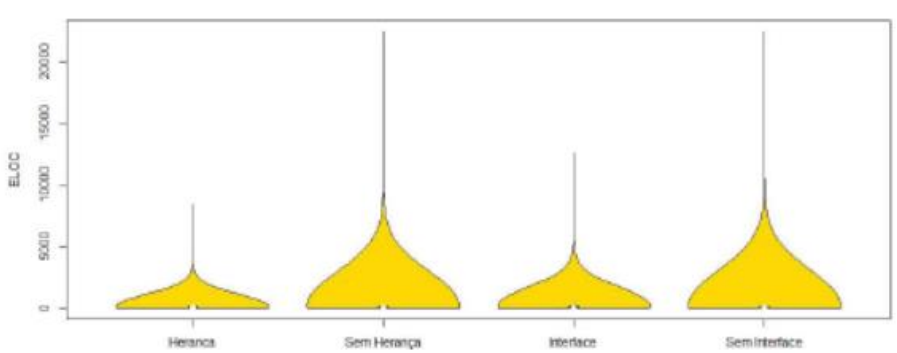
CBO



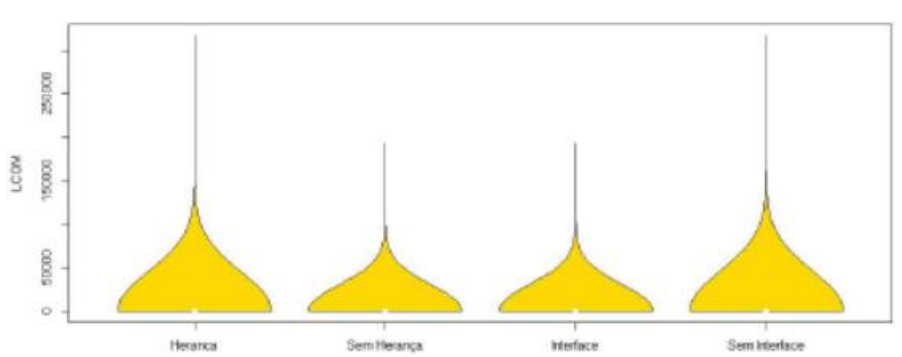


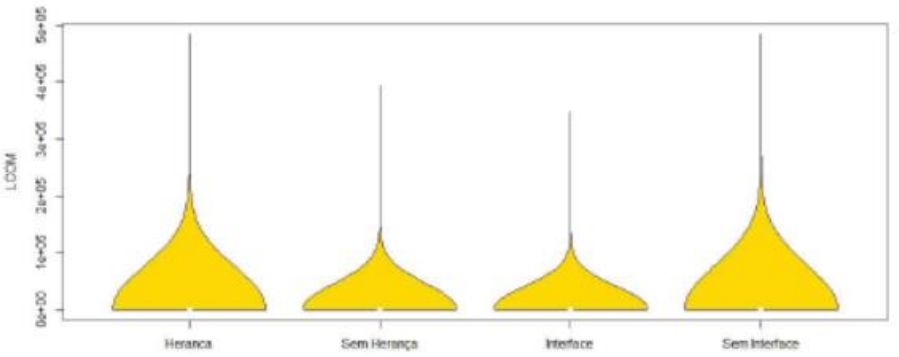
ELOC





LCOM





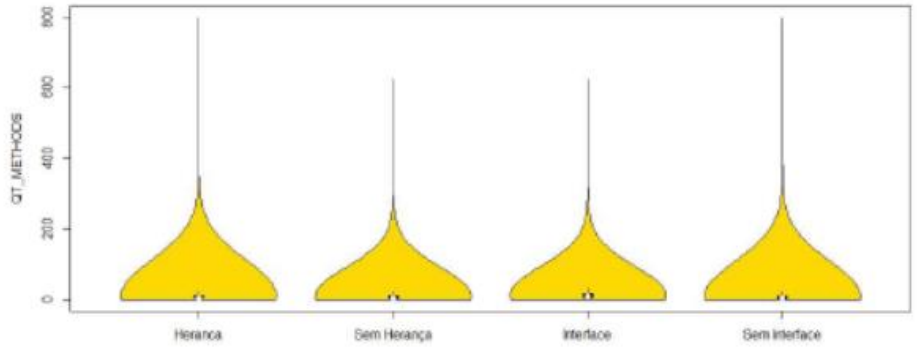
Resultados – RQ #4

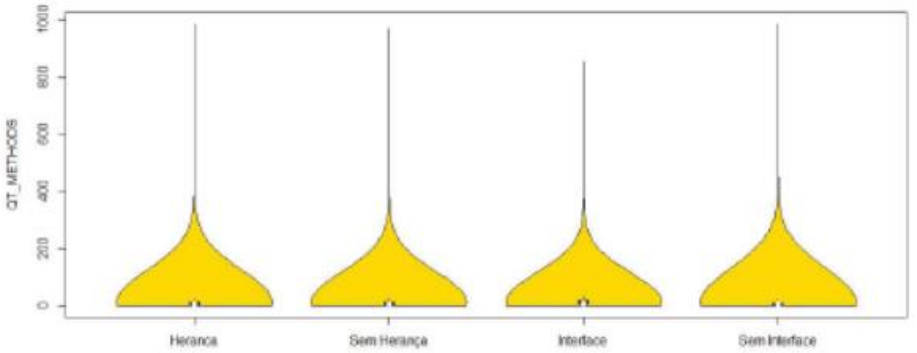
NOM

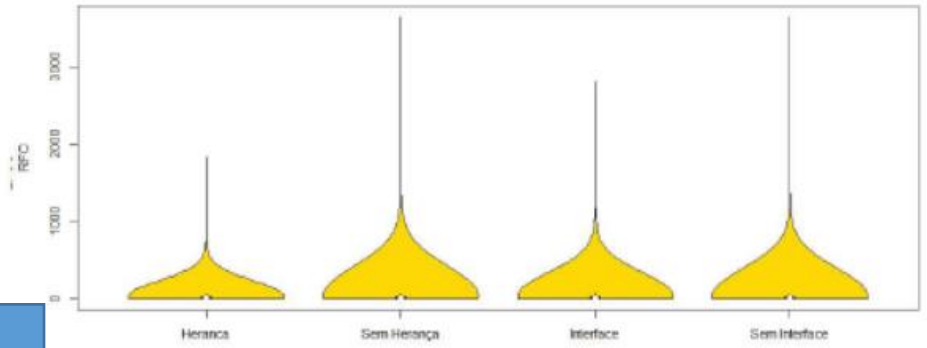
RFC

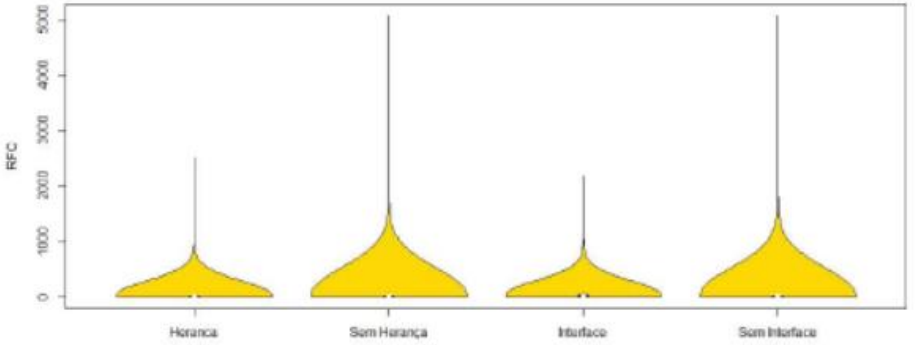
α = 0,1741

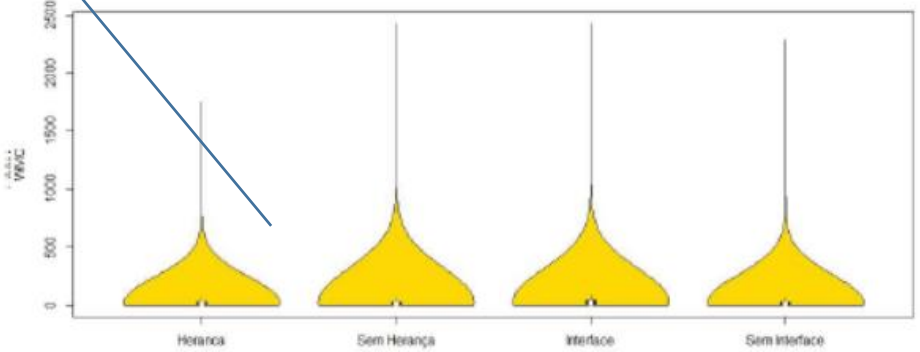
WMC

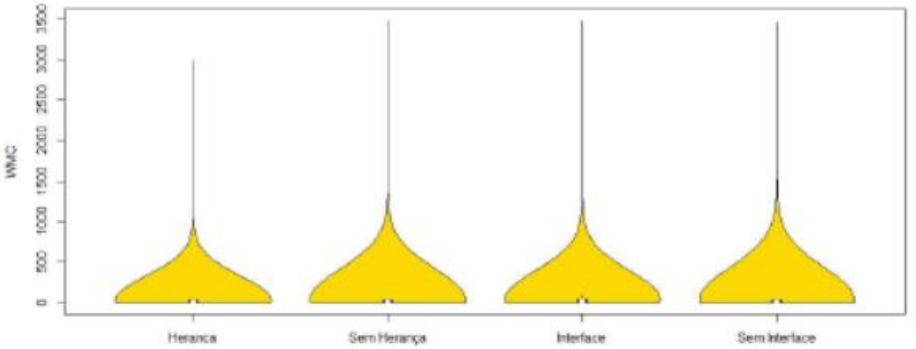












Resultados – RQ #4

	Versão Inicial				Versão Final			
	< Março-2009		>= Março-2009		< Junho-2011		>= Junho-2011	
	Herança	Interface	Herança	Interface	Herança	Interface	Herança	Interface
CBO	< 0,05	< 0,05	0,0367	< 0,05	< 0,05	< 0,05	< 0,05	< 0,05
ELOC	< 0,05	< 0,05	< 0,05	< 0,05	< 0,05	< 0,05	< 0,05	< 0,05
LCOM	< 0,05	< 0,05	< 0,05	< 0,05	< 0,05	< 0,05	< 0,05	< 0,05
NOM	< 0,05	< 0,05	< 0,05	< 0,05	< 0,05	< 0,05	< 0,05	< 0,05
RFC	< 0,05	< 0,05	< 0,05	< 0,05	< 0,05	< 0,05	0,5017	< 0,05
WMC	< 0,05	< 0,05	0,5079	< 0,05	< 0,05	< 0,05	< 0,05	< 0,05

Resultados – RQ #5

	Versão Inicial			Versão Final		
	Com C.D.S.Private	Sem C.D.S.Private	X^2 p-value	Com C.D.S.Private	Sem C.D.S.Private	X^2 p-value
Com Herança	1,051	105,749	< 2.2e-16	1,567	164,130	< 2.2e-16
Sem Herança	2,581	141,707		3,720	209,206	
Com Interface	408	45,592	< 2.2e-16	696	70,114	< 2.2e-16
Sem Interface	3,222	201,740		4,649	303,391	
	Com Complex Class	Sem Complex Class	X^2 p-value	Com Complex Class	Sem Complex Class	X^2 p-value
Com Herança	813	106,447	0.1346	1,297	165,150	5.592e-08
Sem Herança	1,022	143,754		1,346	211,960	
Com Interface	803	45,781	< 2.2e-16	1,067	70,449	< 2.2e-16
Sem Interface	1,033	204,298		1,599	307,064	
	Com Functional D.	Sem Functional D.	X^2 p-value	Com Functional D.	Sem Functional D.	X^2 p-value
Com Herança	300	106,398	0.5331	460	165,091	0.5603
Sem Herança	385	143,695		568	211,864	
Com Interface	85	45,777	8.09e-05	137	70,442	1.364e-05
Sem Interface	600	200,194		891	306,917	
	Com God Class	Sem God Class	X^2 p-value	Com God Class	Sem God Class	X^2 p-value
Com Herança	3,067	105,834	3.557e-12	4,975	164,106	0.0009353
Sem Herança	4,867	142,703		6,792	210,388	
Com Interface	2,672	42,259	< 2.2e-16	3,754	69,663	< 2.2e-16
Sem Interface	5,252	203,164		8,062	305,228	

Resultados – RQ #5

	Versão Inicial			Versão Final		
	Com Lazy Class	Sem Lazy Class	X^2 p-value	Com Lazy Class	Sem Lazy Class	X^2 p-value
Com Herança	17,809	89,071	0.8616	27,664	138,152	0.04209
Sem Herança	24,001	120,274		34,957	177,734	
Com Interface	4,788	41,094	< 2.2e-16	6,402	64,208	< 2.2e-16
Sem Interface	36,999	168,168		56,189	252,121	
	Com Long Method	Sem Long Method	X^2 p-value	Com Long Method	Sem Long Method	X^2 p-value
Com Herança	978	105,968	1.826e-12	1,720	164,293	3.757e-13
Sem Herança	1,750	142,927		2,760	210,633	
Com Interface	626	45,564	6.058e-10	1,116	70,039	< 2.2e-16
Sem Interface	2,102	203,209		3,367	305,292	
	Com Spaghetti Code	Sem Spaghetti Code	X^2 p-value	Com Spaghetti Code	Sem Spaghetti Code	X^2 p-value
Com Herança	2,294	106,444	1.54e-09	3,692	165,125	0.005518
Sem Herança	3,645	143,692		5,035	211,855	
Com Interface	2,053	45,776	< 2.2e-16	1,116	70,039	< 2.2e-16
Sem Interface	3,879	204,238		3,367	305,292	

Resultados – RQ #6

97,35% Herança

Tipo	Sempre		Perdeu		Depois		Nunca		Total	
	Ult	Del	Ult	Del	Ult	Del	Ult	Del	Ult	Del
Herança	384.933 (37,11%)	46.950 (35,39%)	5.875 (0,5%)	480 (0,3%)	21.510 (2,07%)	1827 (1,37%)	624.860 (60,24%)	83.371 (62,86%)	1.037.178 (88,66%)	132.628 (11,33%)
Interface	185.763 (17,91%)	23.226 (17,51%)	11.372 (1,09%)	814 (0,61%)	13.610 (1,31%)	1.043 (0,78%)	826.433 (79,68%)	107.545 (81,08%)	1.037.178 (88,66%)	132.628 (11,33%)

97,59% Imp. Interface

Resultados – RQ #6

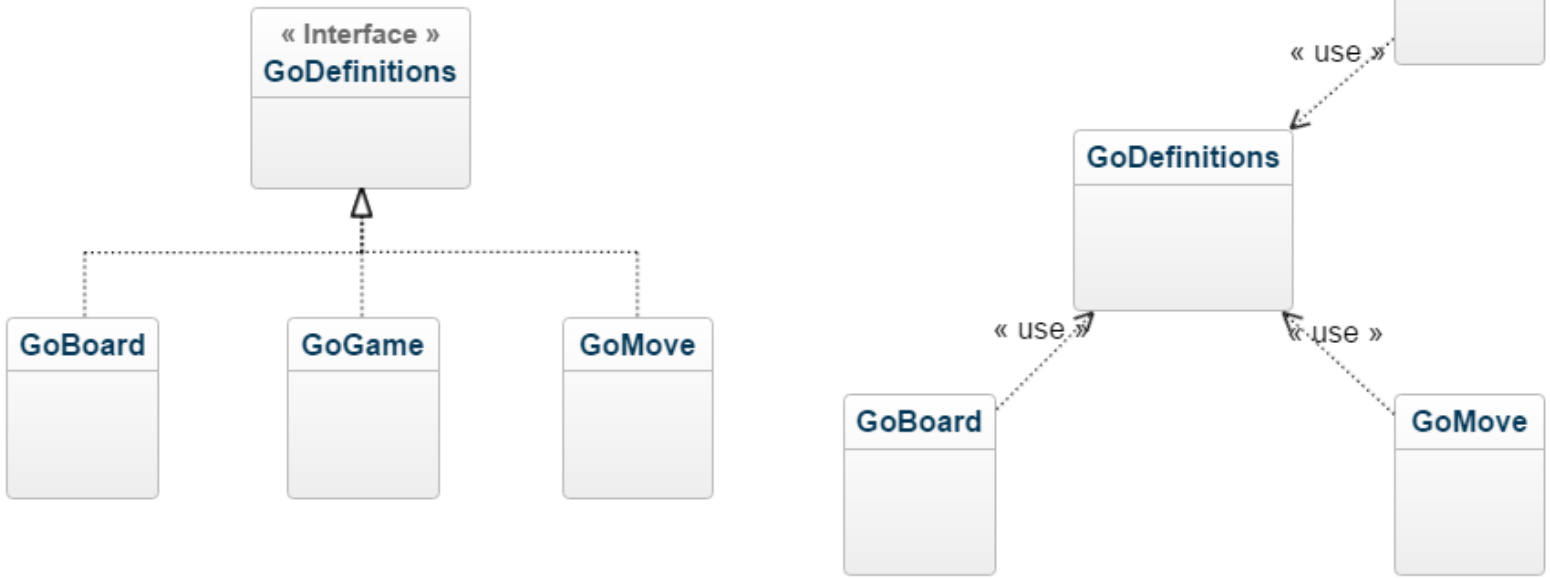
Tipo	Sempre		Perdeu		Depois		Nunca		Total	
	Ult	Del	Ult	Del	Ult	Del	Ult	Del	Ult	Del
Herança	384.933	46.950	5.875	480	21.510	1827	624.860	83.371	1.037.178	132.628
	(37,11%)	(35,39%)	(0,5%)	(0,3%)	(2,07%)	(1,37%)	(60,24%)	(62,86%)	(88,66%)	(11,33%)
Interface	185.763	23.226	11.372	814	13.610	1.043	826.433	107.545	1.037.178	132.628
	(17,91%)	(17,51%)	(1,09%)	(0,61%)	(1,31%)	(0,78%)	(79,68%)	(81,08%)	(88,66%)	(11,33%)

98,25% Herança

98,59% Imp. Interface

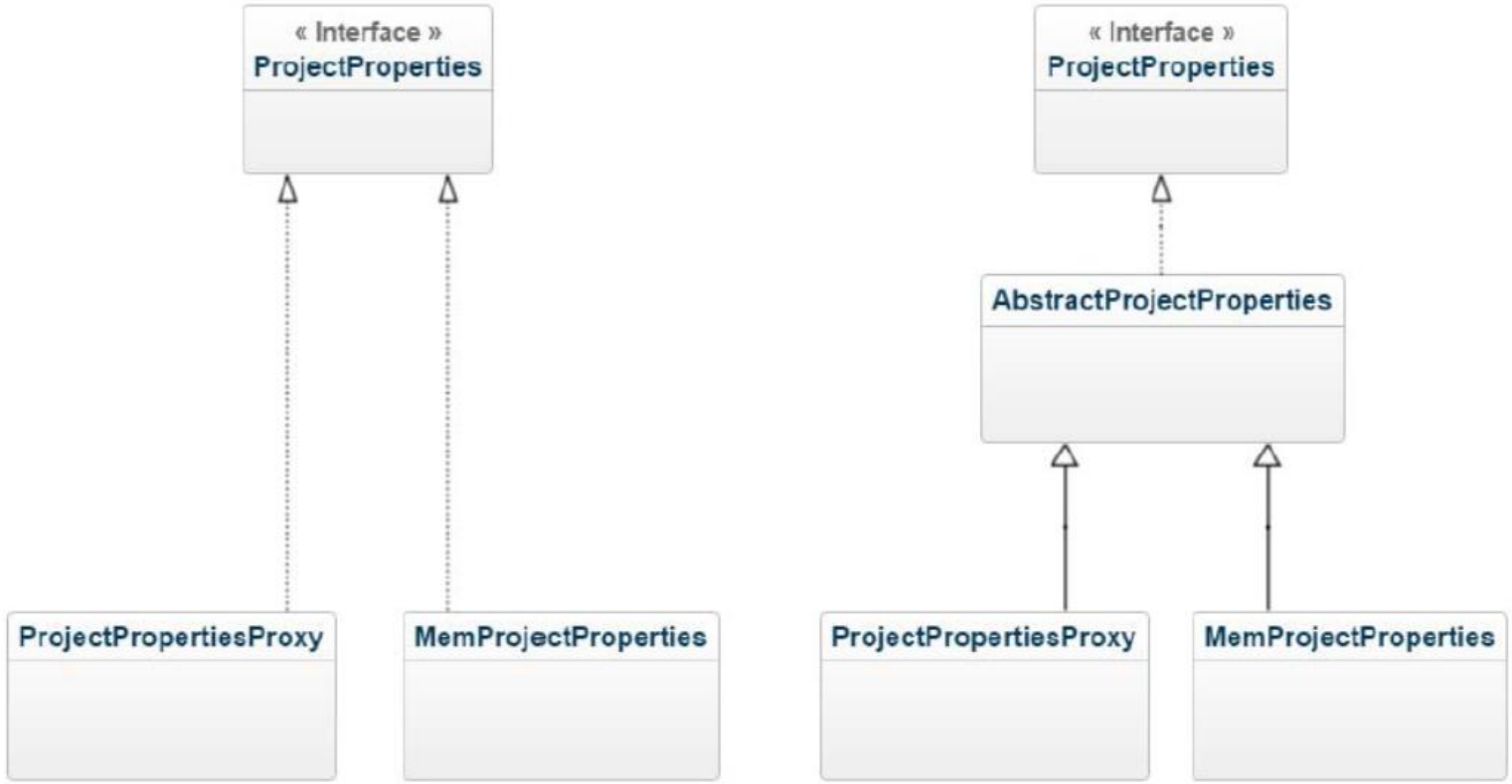
Resultados – RQ #6

1) Abstrações Incertas (15 alterações – 37,5%)



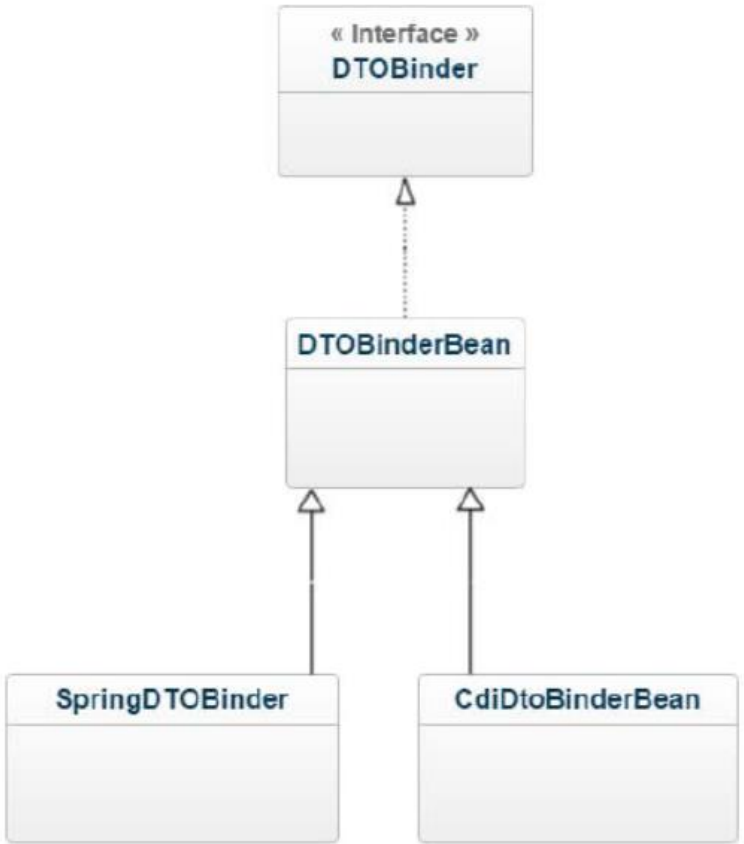
Resultados – RQ #6

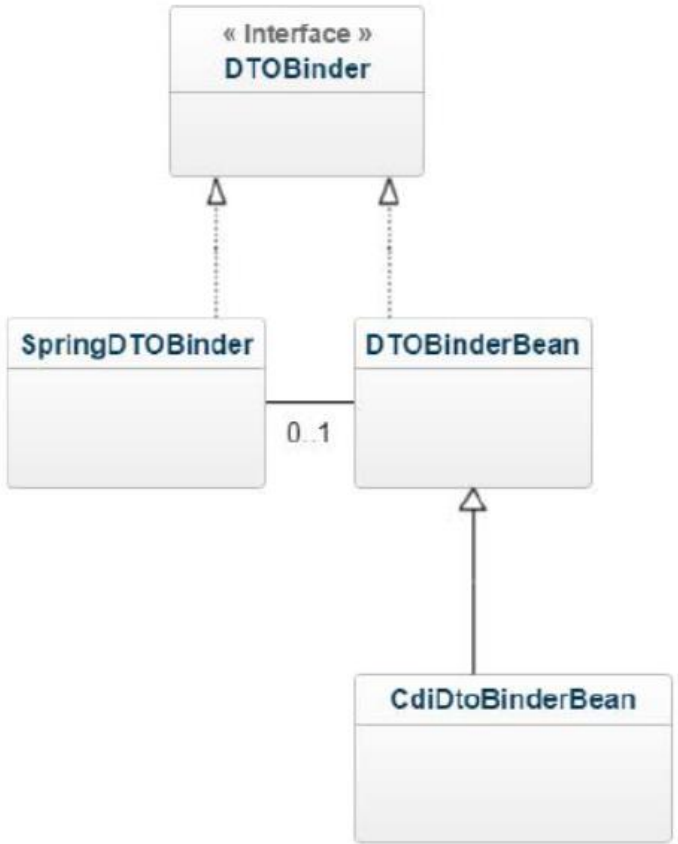
2) Comportamento padrão para as interfaces (9 alterações – 22,5%)



Resultados – RQ #6

3) Novas funcionalidades com adoção de boas práticas (7 alterações – 17,5%)





Resultados – RQ #6

4) Outros (9 alterações – 22,5%)

Sem identificação (4 alterações)

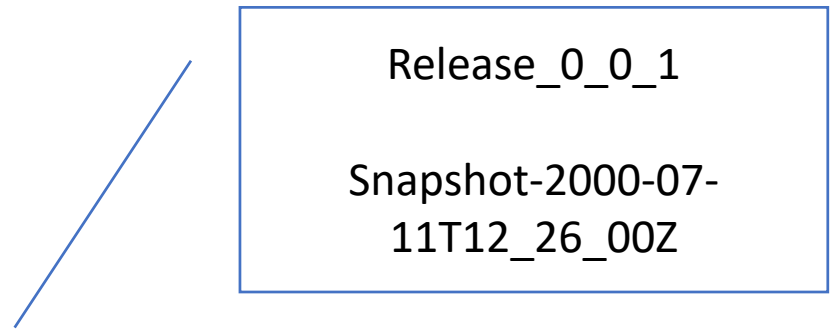
Estender classes externas (3 alterações)

Refactoring move package (2 alterações)

ROTEIRO

1. Contextualização do Problema.
2. Objetivos.
3. Metodologia.
4. Resultados.
4. **Ameaças à validade.**
5. Conclusão.

1) Tags.



1) Tags;

2) Refactoring move package ou rename class.

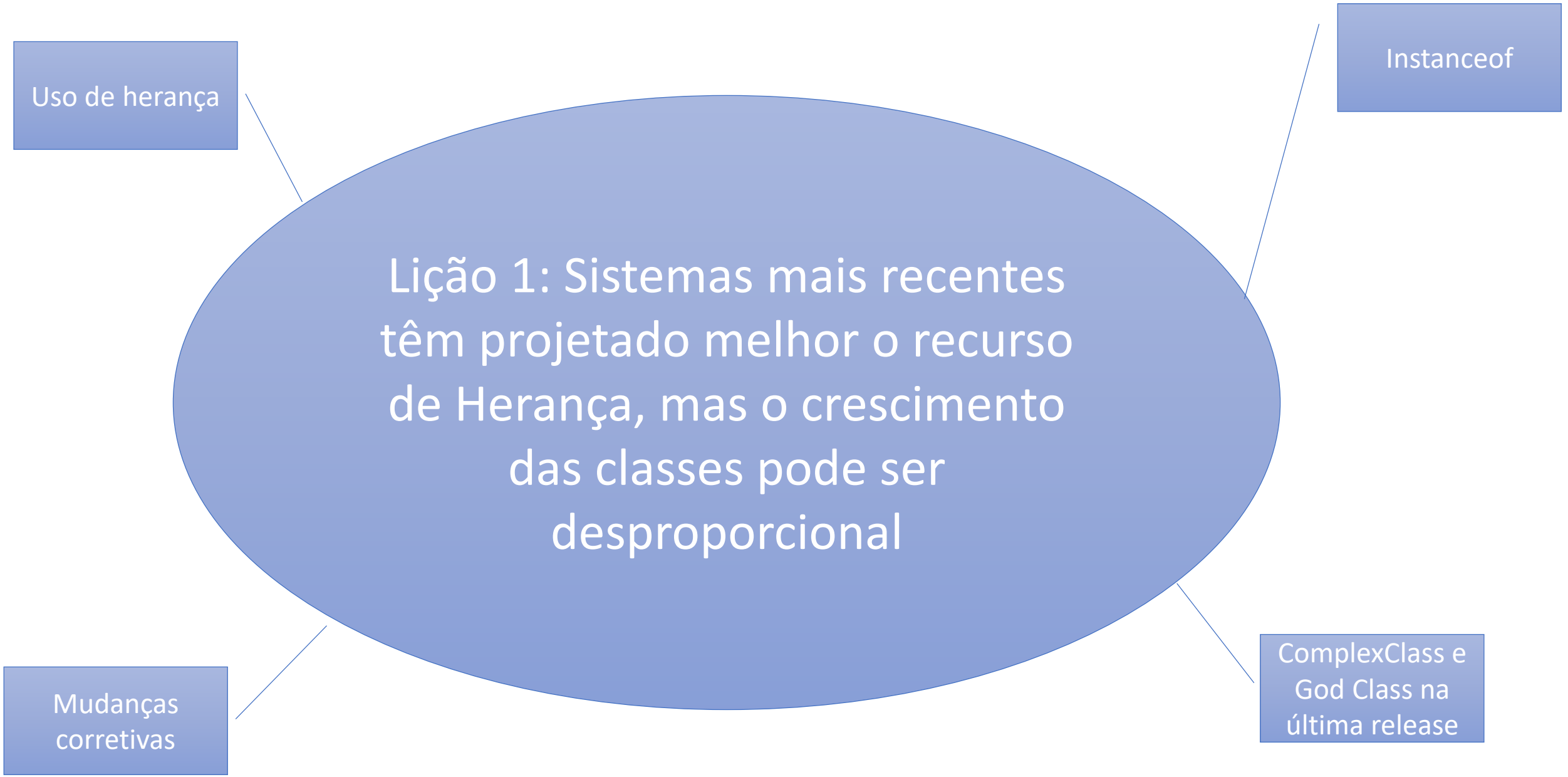
1) Tags;

2) Refactoring move package ou rename class;

3) Escopo restrito à plataforma Java.

ROTEIRO

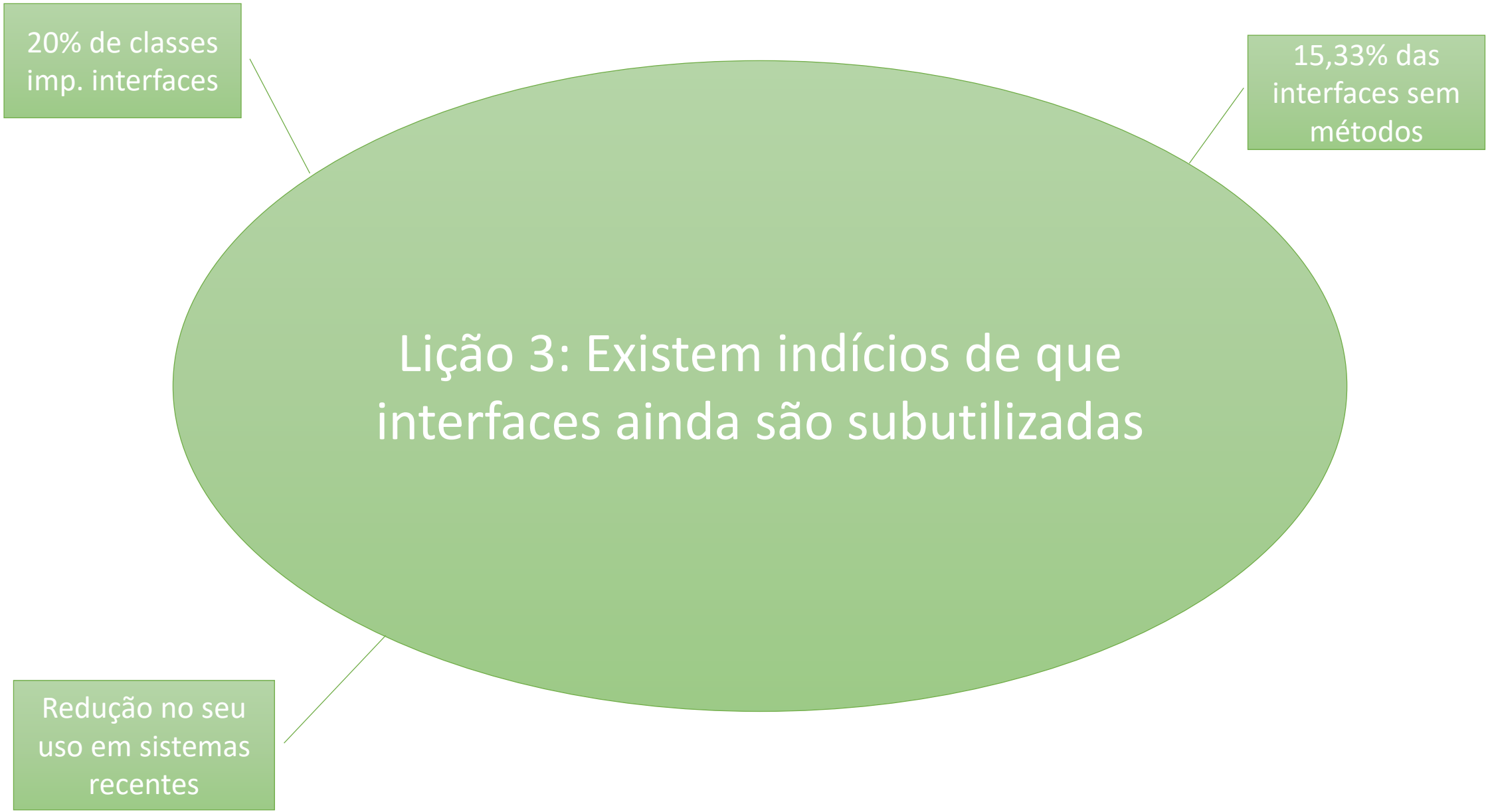
1. Contextualização do Problema.
2. Objetivos.
3. Metodologia.
4. Resultados.
4. Ameaças à validade.
5. **Conclusão.**

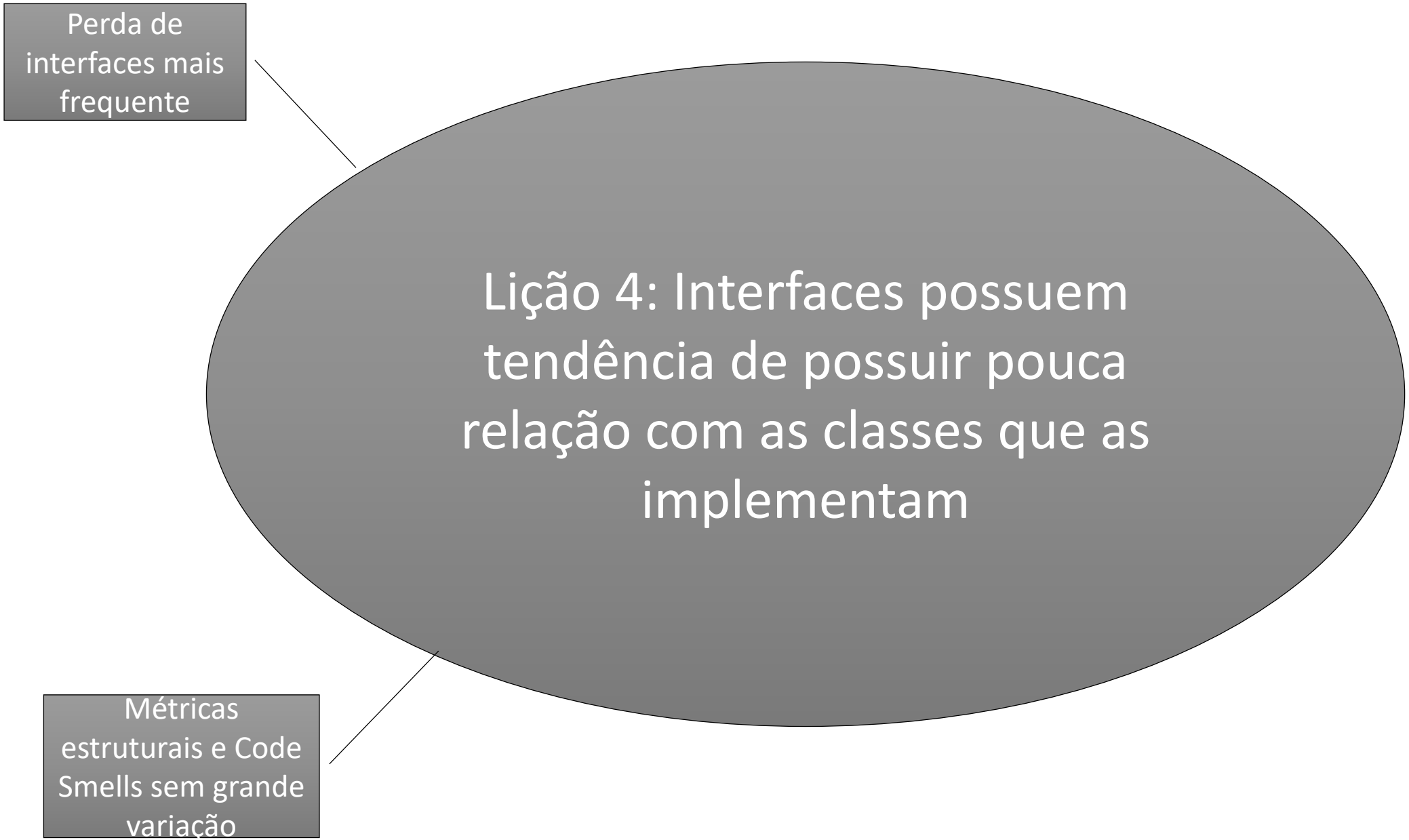

Uso de herança
Instanceof
Lição 1: Sistemas mais recentes têm projetado melhor o recurso de Herança, mas o crescimento das classes pode ser desproporcional
Mudanças corretivas
ComplexClass e God Class na última release

Inclusões de herança

Lazy Class

Lição 2: Desenvolvedores ainda tendem a projetar herança visando primariamente o reaproveitamento de código


20% de classes imp. interfaces
15,33% das interfaces sem métodos
Lição 3: Existem indícios de que interfaces ainda são subutilizadas
Redução no seu uso em sistemas recentes


Perda de interfaces mais frequente
Lição 4: Interfaces possuem tendência de possuir pouca relação com as classes que as implementam
Métricas estruturais e Code Smells sem grande variação