

## Referências Bibliográficas

- [Bieman e Kang 1995] BIEMAN, J. M.; KANG, B.-K.. **Cohesion and reuse in an object-oriented system**. In: PROCEEDINGS OF THE 1995 SYMPOSIUM ON SOFTWARE REUSABILITY, SSR '95, p. 259–262, New York, NY, USA, 1995. ACM. 3.1.3, 3.1, A
- [Buckley et al. 2006] BUCKLEY, J.; MENS, T.; ZENGER, M.; RASHID, A. ; KNIESEL, G.. **Towards a taxonomy of software change: Research articles**. J. Softw. Maint. Evol., 17(5):309–332, 2005. 1
- [Carneiro et al. 2009] CARNEIRO, G.; SANT'ANNA, C.; GARCIA, A. F.; CHAVEZ, C. V. F. G. ; MENDONÇA, M. G.. **On the use of software visualization to support concern modularization analysis**. In: PROCEEDINGS OF THE 3RD WORKSHOP ON ASSESSMENT OF CONTEMPORARY MODULARIZATION TECHNIQUES (ACOM), p. 1–8, Orlando, Florida, 2009. 3.5, 7.1.1, 7.1.2
- [Carneiro et al. 2010a] CARNEIRO, G.; SILVA, M.; MARA, L.; SANT'ANNA, C.; GARCIA, A. F. ; MENDONÇA, M. G.. **Identifying code smells with multiple concern views**. In: IN: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE - SBES. ANAIS DO XXIV SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE - SBES, Salvador, BA, Brasil, 2010. 1, 1.2, 3.5, 3.5, 7, 7.1.1, 8.1
- [Carneiro et al. 2010b] CARNEIRO, G.; NUNES, A. S.; GARCIA, A. F. ; JUNIOR, P.. **An eclipse-based multi-perspective environment to visualize software coupling**. In: IN: SESSÃO DE FERRAMENTAS DO CONGRESSO BRASILEIRO DE SOFTWARE. ANAIS DA SESSÃO DE FERRAMENTAS DO CONGRESSO BRASILEIRO DE SOFTWARE, Salvador, Bahia, 2010. 3.5, 7.1.1
- [Chidamber e Kemerer 1994] CHIDAMBER, S. R.; KEMERER, C. F.. **A metrics suite for object oriented design**. IEEE Trans. Softw. Eng., 20(6):476–493, 1994. 1.2, 3.1, 3.1.1, 3.1.2, 3.1.3, 3.2, 3.1, 5.3.6, 8, A, A, A, A, A
- [Demeyer et al. 1999] DEMEYER, S.; DUCASSE, S. ; LANZA, M.. **A hybrid reverse engineering approach combining metrics and program**

- visualization**. In: WCRE '99: PROCEEDINGS OF THE SIXTH WORKING CONFERENCE ON REVERSE ENGINEERING, p. 175, Washington, DC, USA, 1999. IEEE Computer Society. 1.2, 3.5
- [Deursen et al., 2000] VAN DEURSEN, A.; KLINT, P. ; VISSER, J.. **Domain-specific languages: an annotated bibliography**. SIGPLAN Not., 35(6):26–36, 2000. 6.2.3
- [Dhambri et al. 2008] DHAMBRI, K.; SAHRAOUI, H. ; POULIN, P.. **Visual detection of design anomalies**. In: CSMR '08: PROCEEDINGS OF THE 2008 12TH EUROPEAN CONFERENCE ON SOFTWARE MAINTENANCE AND REENGINEERING, p. 279–283, Washington, DC, USA, 2008. IEEE Computer Society. 3.5
- [Dijkstra 1997] DIJKSTRA, E. W.. **A Discipline of Programming**. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1997. 3.1.4
- [Ducasse et al. 2004] RATIU, D.; DUCASSE, S.; GÎRBA, T. ; MARINESCU, R.. **Using history information to improve design flaws detection**. In: CSMR '04: PROCEEDINGS OF THE EIGHTH EUROMICRO WORKING CONFERENCE ON SOFTWARE MAINTENANCE AND REENGINEERING (CSMR'04), p. 223–232, Washington, DC, USA, 2004. IEEE Computer Society. 1.3, 3.2.2, 7.1.2
- [Eclipse 2010] **Eclipse**. Website. Acessado em 10/08/2010, URL: <http://www.eclipse.org/>. 3.3.1, 3.5, 8.2
- [Fenton e Pfleeger 1997] FENTON, N.; PFLEEGER, S. L.. **Software metrics: a rigorous and practical approach**. London ; Boston : PWS Pub., 1997. 3.1, 3.1.1, 3.1, A, A
- [Figueiredo 2006] FIGUEIREDO, E. M. L.. **Uma abordagem quantitativa para desenvolvimento de software orientado a aspectos**. Dissertação de mestrado em informática, Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), RJ - Brasil, 2006. 3.1.4, 5.1
- [Figueiredo et al., 2008] FIGUEIREDO, E.; CACHO, N.; SANT'ANNA, C.; MONTEIRO, M.; KULESZA, U.; GARCIA, A.; SOARES, S.; FERRARI, F.; KHAN, S.; CASTOR FILHO, F. ; DANTAS, F.. **Evolving software product lines with aspects: an empirical study on design stability**. In: PROCEEDINGS OF THE 30TH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING (ICSE' 2008), p. 261–270, New York, NY, USA, 2008. ACM. 3.2.2, 7.1.4

- [Figueiredo et al. 2009] FIGUEIREDO, E.; SANT'ANNA, C.; GARCIA, A. ; LUCENA, C.. **Applying and evaluating concern-sensitive design heuristics**. In: SBES '09: PROCEEDINGS OF THE 2009 XXIII BRAZILIAN SYMPOSIUM ON SOFTWARE ENGINEERING, p. 83–93, Washington, DC, USA, 2009. IEEE Computer Society. 1, 1.2, 3.1.4, 3.1, 3.2.2, 3.2.3, 7.1.2, A
- [Fowler et al. 1999] FOWLER, M.; BECK, K.; BRANT, J.; OPDYKE, W. ; ROBERTS, D.. **Refactoring: Improving the Design of Existing Code**. Addison-Wesley, Reading, MA, USA, 1999. 1, 1.1, 1.4, 2.1.2, 2.2, 3.1.1, 3.3.1, 3.3.2, 5, 5.3, 5.3.4, 8, 8.1, 8.2, B
- [Gall et al. 1997] GALL, H.; JAZAYERI, M.; KLOSCH, R. R. ; TRAUSMUTH, G.. **Software evolution observations based on product release history**. In: IN PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON SOFTWARE MAINTENANCE 1997 (ICSM 97, p. 160–166. IEEE Computer Society Press, 1997. 2.1.1
- [Gamma et al. 1995] GAMMA, E.; HELM, R.; JOHNSON, R. ; VLISSIDES, J.. **Design patterns: elements of reusable object-oriented software**. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995. 1, 2.1.2
- [Garcia 2004] GARCIA, A. F.. **From objects to agents: an aspect-oriented approach**. Doctoral dissertation, Departamento de Informática, PUC-Rio, RJ - Brasil, 2004. 3.1.4
- [Girba et al. 2004] GIRBA, T.; DUCASSE, S. ; RATIU, D.. **Identifying entities that change together**. In: IN NINTH IEEE WORKSHOP ON EMPIRICAL STUDIES OF SOFTWARE MAINTENANCE, 2004. 8.2
- [Greenwood et al., 2007] GREENWOOD, P.; BARTOLOMEI, T.; FIGUEIREDO, E.; GARCIA, A.; CACHO, N.; SANT'ANNA, C.; BORBA, P.; UIRAKULESZA ; RASHID, A.. **On the impact of aspectual decompositions on design stability: An empirical study**. In: ECOOP 2007, LNCS, p. 176–200. Springer-Verlag, 2007. 7.1.4
- [Henderson-Sellers 1996] HENDERSON-SELLERS, B.. **Object-oriented metrics: measures of complexity**. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996. 1.2, 3.1, 3.2, 8
- [Hist-Inspect 2010] **Hist-inspect**. Acessado em 10/08/2009, URL: <http://www.inf.puc-rio.br/~lsilva>. 6

- [Hudak 1996] HUDAK, P.. **Building domain-specific embedded languages**. ACM Comput. Surv., p. 196. 6.2.3
- [Kan 1994] KAN, S. H.. **Metrics and Models in Software Quality Engineering**. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1994. Foreword By-Thomas, Brian. 3.1.1
- [Lanza e Marinescu 2006] LANZA, M.; MARINESCU, R. ; DUCASSE, S.. **Object-Oriented Metrics in Practice**. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. 1.2, 2.1.2, 3.2, 3.2.2, 3.2.2, 3.2.3, 3.2.3, 3.2.3, 5.3.2, 5.3.7, 7, 7.1.2, 8
- [Lehman e Belady 1985] Lehman, M. M.; Belady, L. A., editors. **Program evolution: processes of software change**. Academic Press Professional, Inc., San Diego, CA, USA, 1985. 2.1.2
- [Lehman et al. 1997] LEHMAN, M. M.; RAMIL, J. F.; WERNICK, P. D.; PERRY, D. E. ; TURSKI, W. M.. **Metrics and laws of software evolution - the nineties view**. In: METRICS '97: PROCEEDINGS OF THE 4TH INTERNATIONAL SYMPOSIUM ON SOFTWARE METRICS, p. 20, Washington, DC, USA, 1997. IEEE Computer Society. 2.1.2, 2.2, 8
- [Li e Henry 1993] LI, W.; HENRY, S.. **Maintenance metrics for the object-oriented paradigm**. In: PROC. IEEE SYMP. SOFTWARE METRICS, p. 52–60, May 1993. 3.1, 8, A
- [Lorenz e Kidd 1994] LORENZ, M.; KIDD, J.. **Object-oriented software metrics: a practical guide**. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1994. 3.1, 3.1.1, 3.1, 5.1, 8, A, A
- [Macia et al 2008] MACIA, I.; STAA, A. ; SANT'ANNA, C.. **Towards a catalog of aspect-oriented refactorings**. In: V EXPERIMENTAL SOFTWARE ENGINEERING LATIN AMERICAN WORKSHOP (ESELAW 2008), 2008. 3.2.3
- [Marinescu 2002] MARINESCU, R.. **Measurement and Quality in Object-Oriented Design**. Ph.d in computer science, Faculty of Automatics and Computer Science of the Politehnica University of Timisoara, RO, 2002. 1, 3.2.3, 3.2.3, 5.3.7, 8, A, A, A
- [Marinescu 2004] MARINESCU, R.. **Detection strategies: Metrics-based rules for detecting design flaws**. In: ICSM '04: PROCEEDINGS OF THE 20TH IEEE INTERNATIONAL CONFERENCE ON SOFTWARE

- MAINTENANCE, p. 350–359, Washington, DC, USA, 2004. IEEE Computer Society. (document), 1.2, 3.2, 3.1, 3.2.1, 3.2.1, 3.2.2, 5.3.7, 8
- [Mara et al. 2010b] MARA, L.; HONORATO, G.; DANTAS, F.; GARCIA, A. ; LUCENA, C.. **Hist-inspect: Uma ferramenta de apoio à avaliação sensível à história de código.** In: SESSÃO DE FERRAMENTAS DO CONGRESSO BRASILEIRO DE SOFTWARE. ANAIS DA SESSÃO DE FERRAMENTAS DO CONGRESSO BRASILEIRO DE SOFTWARE, Salvador, BA, Brasil, 2010. 6, 8.1
- [Mara et al. 2010a] MARA, L.; DANTAS, F.; HONORATO, G.; GARCIA, A. ; LUCENA, C.. **Detectando anomalias de código em evolução: O que a história pode revelar?** In: QUARTO SIMPÓSIO BRASILEIRO DE COMPONENTES, ARQUITETURAS E REUTILIZAÇÃO DE SOFTWARE (SBCARS' 2010), Salvador, BA, Brasil, 2010. 8.1
- [McCabe 1976] MCCABE, T. J.. **A complexity measure.** In: ICSE '76: PROCEEDINGS OF THE 2ND INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, p. 407, Los Alamitos, CA, USA, 1976. IEEE Computer Society Press. A
- [Mens e Demeyer 2001] MENS, T.; DEMEYER, S.. **Future trends in software evolution metrics.** In: IWPSE '01: PROCEEDINGS OF THE 4TH INTERNATIONAL WORKSHOP ON PRINCIPLES OF SOFTWARE EVOLUTION, p. 83–86, New York, NY, USA, 2001. ACM. 8.2
- [Mens e Tourwé 2004] MENS, T.; TOURWÉ, T.. **A survey of software refactoring.** IEEE Trans. Softw. Eng., 30(2):126–139, 2004. 2.2
- [Moha et al. 2006] MOHA, N.; GUEHENEUC, Y.-G. ; LEDUC, P.. **Automatic generation of detection algorithms for design defects.** In: ASE '06: PROCEEDINGS OF THE 21ST IEEE/ACM INTERNATIONAL CONFERENCE ON AUTOMATED SOFTWARE ENGINEERING, p. 297–300, Washington, DC, USA, 2006. IEEE Computer Society. 3.4
- [Moha et al. 2010] MOHA, N.; GUEHENEUC, Y.-G.; DUCHIEN, L. ; LE MEUR, A.-F.. **Decor: A method for the specification and detection of code and design smells.** IEEE Trans. Softw. Eng., 36(1):20–36, 2010. 1.4, 3.4, 6.3
- [Olbrich et al., 2009] OLBRICH, S.; CRUZES, D. S.; BASILI, V. ; ZAZWORKA, N.. **The evolution and impact of code smells: A case study of**

- two open source systems. In: ESEM '09, p. 390–400, Washington, DC, USA, 2009. IEEE Computer Society. 7.1.4
- [Parnas 1994] PARNAS, D. L.. **Software aging**. In: ICSE '94: PROCEEDINGS OF THE 16TH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, p. 279–287, Los Alamitos, CA, USA, 1994. IEEE Computer Society Press. 2.1.2
- [Pree e Sikora 1997] PREE, W.; SIKORA, H.. **Design patterns for object-oriented software development (tutorial)**. In: ICSE '97: PROCEEDINGS OF THE 19TH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, p. 663–664, New York, NY, USA, 1997. ACM. 1.1
- [Pressman 2006] PRESSMAN, R. S.. **Software engineering: a practitioner's approach (6nd ed.)**. McGraw-Hill, Inc., New York, NY, USA, 2006. 2.1.1, 2.1.2
- [Ratiu 2003] RATIU, D.. **Time based detection strategies**. Masters thesis in computer science, Faculty of Automatics and Computer Science of the Politehnica University of Timisoara, RO, 2003. 3.1, A, A
- [Ratiu et al. 2004] RATIU, D.; DUCASSE, S.; GIRBA, T. ; MARINESCU, R.. **Evolution-enriched detection of god classes**. In: IN PROCEEDINGS OF THE 2ND WORKSHOP ON COMPUTER AIDED VERIFICATION OF INFORMATION SYSTEMS (CAVIS), p. 3–7, 2004. 1.2, 1.3, 3.1.5, 3.2.2, 8
- [Riel 1996] RIEL, A. J.. **Object-Oriented Design Heuristics**. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1996. 1.1, 1.3, 2.1.2, 3.3.2, 5.3, 5.3.1, 5.3.7, 8, 8.1, B
- [Rijsbergen 2001] VAN RIJSBERGEN, C. J.. **Getting into information retrieval**. p. 1–20, 2001. 1.4, 7.1.5
- [Saleh e Gomma 2005] SALEH, M.; GOMAA, H.. **Separation of concerns in software product line engineering**. In: MACS 05, p. 1–5, New York, NY, USA, 2005. ACM. 7.1.4
- [Sant'Anna et al. 2003] SANT'ANNA, C.; GARCIA, A. F.; DE LUCENA, C. J. P.; CHAVEZ, C. ; STAA, A.. **On the reuse and maintenance of aspect-oriented software: An evaluation framework**. In: IN: XVII BRAZILIAN SYMPOSIUM ON SOFTWARE ENGINEERING, p. 19–34, Manaus, Brasil, 2003. 3.1.4, 3.1, A

- [Sant'Anna et al. 2007] SANT'ANNA, C.; FIGUEIREDO, E.; GARCIA, A. F. ; DE LUCENA, C. J. P.. **On the modularity of software architectures: A concern-driven measurement framework**. In: ECSA, p. 207–224, 2007. 3.2.3
- [Selby e Basili 1991] SELBY, R. W.; BASILI, V. R.. **Analyzing error-prone system structure**. IEEE Trans. Softw. Eng., 17(2):141–152, 1991. 3.1.2
- [Shatnawi e Li 2006] SHATNAWI, R.; LI, W.. **An investigation of bad smells in object-oriented design**. In: ITNG '06, p. 161–165, Washington, DC, USA, 2006. IEEE Computer Society. 7.1.4
- [Sommerville 2006] SOMMERVILLE, I.. **Software Engineering: (Update) (8th Edition) (International Computer Science)**. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2006. 2.1.1
- [Tekinerdogan e Aksit 1998] TEKINERDOGAN, B.; AKSIT, M.. **Providing automatic support for heuristic rules of methods**. In: ECOOP '98: WORKSHOP ON OBJECT-ORIENTED TECHNOLOGY, p. 496–498, London, UK, 1998. Springer-Verlag. 3.2.3
- [Together 2009] **Together**. Website. Acessado em 10/08/2009, URL: <http://www.borland.com/br/products/together/>. (document), 1.3, 1.3, 1.4, 3.2, 3.3, 3.3.3, 6.1, 6.3, 7.1.1, 8
- [iPlasma 2009] **iplasma**. Website. Acessado em 12/03/2010, URL: <http://loose.upt.ro/iplasma/>. 1.3, 3.3, 3.3.3, 6.1, 8
- [inCode 2009] **incode**. Website. Acessado em 10/08/2009, URL: <http://loose.upt.ro/incode/pmwiki.php/Main/Incode?from=Main.HomePage>. 1.3, 1.4, 3.3, 3.3.2, 3.3.3, 6.1, 8
- [inFusion 2009] **infusion**. Website. Acessado em 12/03/2010, URL: <http://www.intooitus.com/inFusion.html>. 1.3, 3.3, 3.3.3, 6.1, 8

## A

### Catálogo de Métricas

O objetivo deste anexo é disponibilizar um catálogo de métricas que podem ser úteis em processos de avaliação de código. Tais métricas não estão agrupadas por categoria (métricas convencionais, sensíveis a interesses ou sensíveis à história), nem por finalidade (métricas de tamanho, coesão, acoplamento, dentre outras). Optamos por listar tais métricas em ordem alfabética de suas siglas visando contribuir com a fácil localização de cada um dos acrônimos citados nesta dissertação.

#### **ATFD - *Access To Foreign Data* (Acesso a Dados Estrangeiros)**

**Breve definição:** Dada uma classe, contabiliza o número de atributos que são acessados diretamente ou através de métodos de acesso e que pertencem a outras classes não relacionadas.

**Referências:** (Marinescu 2002).

#### **CBO - *Coupling Between Object Classes* (Acoplamento entre Classes de Objetos)**

**Breve definição:** Dada uma classe, mede o acoplamento entre essa classe e as demais classes do sistema.

**Referências:** (Chidamber e Kemerer 1994).

#### **CC - *Changing Classes* (Classes Modificadas)**

**Breve definição:** É definida pelo número de classes que potencialmente serão alteradas como resultado de alterações em uma dada classe.

**Referências:** (Marinescu 2002).



**CM - *Changing Methods* (Métodos Modificados)**

**Breve definição:** Define-se pelo número de métodos distintos que dependem da classe ou método avaliado.

**Referências:** (Marinescu 2002).

**CDC - *Concern Diffusion over Components* (Difusão de Interesses em Componentes)**

**Breve definição:** Conta o número de componentes ou módulos de um sistema cujo propósito principal é contribuir com a implementação de um interesse que está sendo avaliado.

**Referências:** (Sant'Anna et al. 2003).

**DIT - *Depth Inheritance Tree* (Profundidade da Árvore de Herança)**

**Breve definição:** Mede a profundidade de uma classe em sua árvore de herança, ou seja, o número de ancestrais que essa classe possui.

**Breve definição:** (Chidamber e Kemerer 1994).

**LCOM - *Lack of Cohesion of Methods* (Baixa Coesão de Métodos)**

**Breve definição:** Contabiliza quão baixa é a coesão de uma classe. É calculada baseando-se na utilização dos atributos da classe pelos métodos. São agrupados todos os pares de métodos de uma classe, verificando-se se eles compartilham algum atributo. Após tal verificação é subtraído o número de pares que tem algum tipo de compartilhamento de atributos dos que não tem.

**Breve definição:** (Chidamber e Kemerer 1994).

**LOC - *Lines of Code* (Linhas de Código):**

**Breve definição:** Contabiliza o número de linhas de código de um módulo. Linhas em branco e comentários não são contabilizados.

**Referências:** (Fenton e Pfleeger 1997, Lorenz e Kidd 1994).

### **LOCstab - *Lines Of Code Stability* (Estabilidade de Linhas de Código)**

**Breve definição:** Representa a fração do tempo de vida de um módulo em que o seu número de linhas de código não se altera.

**Referências:** (Ratiu 2003).

### **NCC - *Number of Concerns per Component* (Número de Interesses por Componente)**

**Breve definição:** Contabiliza o número de interesses implementados por um dado módulo ou componente.

**Referências:** (Figueiredo et al. 2009).

### **NOA - *Number of Attributes* (Número de Atributos)**

**Breve definição:** Contabiliza o número de atributos de uma dada classe.

**Referências:** (Lorenz e Kidd 1994).

### **NOC - *Number of Children* (Número de Filhos)**

**Breve definição:** Contabiliza o número de filhos ou descendentes diretos de uma determinada classe.

**Referências:** (Chidamber e Kemerer 1994).

### **NOM - *Number of Methods* (Número de Métodos)**

**Breve definição:** Contabiliza o número de métodos de classe ou outro módulo (um pacote ou todo o sistema).

**Referências:** (Li e Henry 1993).

**NOMstab - *Number of Methods Stability* (Estabilidade do Número de Métodos)**

**Breve definição:** Representa a fração do tempo de vida de uma classe que o seu número de métodos não se altera.

**Referências:** (Ratiu 2003).

**NOS - *Number of Statements* (Número de Comandos)**

**Breve definição:** Número de comandos em um dado módulo ou componente.

**Referências:** (Fenton e Pfleeger 1997).

**TCC - *Tight Class Cohesion* (Alta Coesão de Classes)**

**Breve definição:** Avalia o quão alta é a coesão de uma dada classe. Representa o número relativo de pares de métodos que acessam pelo menos um atributo pertencente a essa classe.

**Referências:** (Bieman e Kang 1995).

**WMC - *Weighted Method Count* (Contagem de Métodos Ponderada)**

**Breve definição:** Contabiliza a complexidade de uma classe. Representa a soma da complexidade ciclomática (McCabe 1976) de cada um dos métodos pertencente à classe avaliada.

**Referências:** (Chidamber e Kemerer 1994).

## B

### Oráculos de Anomalias dos Sistemas Avaliados

Neste anexo são apresentadas anomalias reais identificadas nos sistemas alvos pertencentes a esta pesquisa. Na detecção foram consideradas os seguintes tipos de anomalias de granularidade de classes: *God Class*, *Divergent Change* e *Shotgun Surgery* (Riel 1996, Fowler et al. 1999). Algumas análises manuais também foram iniciadas em relação à detecção de uma anomalia de granularidade de método conhecida como *Feature Envy* (Fowler et al. 1999).

Destacamos que, apesar de nem todas as anomalias citadas anteriormente terem sido consideradas nas análises de detecções sensíveis à história, disponibilizamos neste anexo os resultados de todas as detecções manuais realizadas para ambos os sistemas. Nosso objetivo é facilitar a realização de outras pesquisas que tomem como base estudos previamente realizados por nós.

Com a padronização apresentada nas tabelas, por exemplo, é possível iniciar investigações relacionadas a reincidência de anomalias ao longo de versões ou coexistências frequentemente comuns entre duas ou mais anomalias. Em todas as tabelas, células em branco significam que não foram identificadas anomalias naquela versão do módulo apresentado. Células destacadas em tom de cinza significam que aquele módulo não existia nas versões referenciadas ou existia com outro nome e foi renomeado. Células com o padrão “<SIGLA<sub>1</sub>>, <SIGLA<sub>2</sub>>”, representam que o módulo representado pela linha da tabela possui as anomalias representadas por cada uma dessas siglas.

As siglas comumente apresentadas nas tabelas são GC, DC, SS e FE. Elas representam respectivamente:

- GC: *God Class*;
- DC: *Divergent Change*;
- SS: *Shotgun Surgery*;
- FE: *Feature Envy*.

Sistema Mobile Media (MM) - Anomalias de Classes						
	v2	v3	v4	v5	v6	v7
BaseController	GC	GC, DC	GC, DC			
ControllerInterface				SS	SS	SS
ImageAccessor	GC	GC, DC	GC, DC	GC, DC	GC, DC	
ImageMediaAccessor						DC
MediaAccessor						GC, DC, SS
MediaController						GC, DC
MediaListController						DC
PhotoController				DC	GC, DC	
PhotoListController				DC	DC	
PhotoViewScreen					DC	
ScreenSingleton				SS	SS	SS

Tabela B.1: Oráculos para o Mobile Media. Anomalias de classes: *God Class* (GC), *Divergent Change* (DC) e *Shotgun Surgery* (SS). Células em tons de cinza indicam que o módulo não existe nas versões especificadas ou que existe com um outro nome (refatoração “rename”).

Sistema Mobile Media (MM) - Anomalia Feature Envy						
	v2	v3	v4	v5	v6	v7
AlbumController.resetImageData				FE	FE	
AlbumController.resetMediaData						FE
BaseController.handleCommand		FE	FE			
BaseController.init	FE	FE	FE	FE	FE	FE
BaseController.resetImageData	FE	FE	FE			
BaseController.showImageList	FE	FE	FE			
MediaListController.bubbleSort						FE
MediaListController.exchange						FE
MediaListController.handleCommand						FE
MediaListController.showImageList						FE
PhotoListController.bubbleSort				FE	FE	
PhotoListController.exchange				FE	FE	
PhotoListController.handleCommand				FE	FE	
PhotoListController.showImageList				FE	FE	

Tabela B.2: Oráculo para o Mobile Media. Anomalia de método: *Feature Envy* (FE). Células em tons de cinza indicam que o módulo não existe nas versões especificadas ou que existe com um outro nome (refatoração “rename”).

Sistema Health Watcher (HW) - Anomalias de Classes										
	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
ComplaintRepositoryRDB	GC, DC	GC, DC	GC, DC	GC, DC	GC, DC	GC, DC	GC, DC	GC, DC	GC, DC	GC, DC
ComplaintRecord	DC	DC	DC	DC	DC	DC	DC	DC	DC	DC
HealthWatcherFacade	GC, DC	GC, DC	GC, DC	GC, DC	GC, DC	GC, DC	GC, DC	GC, DC	GC, DC	GC, DC
HealthWatcherFacadeInit	GC, DC	GC, DC	GC, DC	GC, DC						
IFacade	DC	DC	DC	DC	DC	DC	DC	DC	DC	DC
SearchComplaintData		GC, DC	GC, DC	GC, DC	GC, DC	GC, DC	GC, DC	GC, DC	GC, DC	GC, DC
ServletSearchComplaintData	GC, DC									

Tabela B.3: Oráculos para o Health Watcher. Anomalias de classes: *God Class* (GC) e *Divergent Change* (DC). Células em tons de cinza indicam que o módulo não existe nas versões especificadas ou que existe com um outro nome (refatoração “rename”).