

Referências

INTRODUÇÃO

1. LARMAN, Craig. Protected variation: The importance of being closed. In: *IEEE Software*, v. 18, p. 89-91. 2001.
2. FOWLER, Martin. *Patterns of Enterprise Application Architecture*. Boston: Addison-Wesley, 2003.
3. FOWLER, Martin. Who needs an architect? In: *IEEE Software*, v. 20. 2003.
4. EDEN, Amnon H.; KAZMAN, Rick. Architecture, design, implementation. In: *25th International Conference on Software Engineering – ICSE*, 2003.
5. *Defining Software architecture*. Pittsburgh: Carnegie Mellon University, 2009.
6. SPOLSKY, Joel. *Language wars*. 2006. Disponível em: <http://www.joelonsoftware.com/items/2006/09/01.html?everything_old_is_new_again>.
7. FOWLER, Martin. *Is Design Dead?* 2004. Disponível em: <<http://www.martinfowler.com/articles/designDead.html>>.
8. SPOLSKY, Joel. *Don't let architecture astronauts scare you*. 2001. Disponível em: <<http://www.joelonsoftware.com/articles/fog0000000018.html>>.
9. LARMAN, Craig; VODDE, Bas Design & Architecture. In: _____. *Practices for Scaling Lean and Agile Development*. Addison-Wesley Professional, 2010.

CAPÍTULO 1

1. ORACLE CORPORATION. *The History of Java Technology*. Disponível em: <<http://www.oracle.com/technetwork/java/javase/overview/javahistory-index-198355.html>>.
2. MULLER, Hans. *Official: Swing is the dominant GUI toolkit*. 2005. Disponível em: <http://weblogs.java.net/blog/hansmuller/archive/2005/10/official_swing.html>.
3. LINDHOLM, Tim; YELLIN, Frank. *The Java™ Virtual Machine Specification*. 2. ed. Prentice Hall, 1999.
4. GOSLING, James; JOY, Bill; STEELE, Guy; BRACHA, Gilad. *The Java™ Language Specification*. 3. ed. Addison-Wesley, 2005.

5. MICROSOFT CORPORATION. *Extension methods (C# Programming Guide)*. Disponível em: <<http://msdn.microsoft.com/en-us/library/bb383977.aspx>>.
6. TOLKSDORF, Robert. *Programming languages for the Java Virtual Machine JVM and JavaScript*. Disponível em: <<http://www.is-research.de/info/vmlanguages/>>.
7. WILLISON, Simon . *A re-introduction to JavaScript*. 2006. Disponível em: <https://developer.mozilla.org/en/a_re-introduction_to_javascript>.
8. VINOSKI, Steve. *Welcome to “The Functional Web”*. 2009. Disponível em: <http://steve.vinoski.net/pdf/IC-Welcome_to_the_Functional_Web.pdf>.
9. SILVEIRA, Paulo. *Trabalhando com closures no Java 8*. 2011. Disponível em: <<http://blog.caelum.com.br/trabalhando-com-closures-no-java-8>>

CAPÍTULO 2

1. ZORN, Benjamin. The measured cost of conservative garbage collection. In: *Software Practice and Experience*, v. 23(7), p. 733-756. 1993.
2. PREISS, Bruno R. *Mark-and-sweep Garbage Collection*. Disponível em: <<http://www.brpreiss.com/books/opus5/html/page424.html>>.
3. LIEBERMAN, Henry; HEWITT, Carl. A Real-Time Garbage Collector Based on the Lifetimes of Objects. In: *Communications of the ACM*, v. 26. 1983.
4. SUN MICROSYSTEMS. *Memory Management in the Java HotSpot™ Virtual Machine*. 2006. Disponível em: <http://java.sun.com/j2se/reference/whitepapers/memorymanagement_whitepaper.pdf>.
5. PRINTEZIS, Tony. *Garbage Collection in the Java HotSpot™ Virtual Machine (Generational Algorithm)*. 2004. Disponível em: <<http://www.devx.com/Java/Article/21977>>.
6. SUN MICROSYSTEMS. “*Java SE 6 Java HotSpot™ Virtual Machine Garbage Collection Tuning*”. Disponível em: <<http://www.oracle.com/technetwork/java/javase/gc-tuning-6-140523.html>>.
7. MASAMITSU, Jon. *Our collectors (Sun)*. Disponível em: <http://blogs.sun.com/jonthecollector/entry/our_collectors>.
8. SUN MICROSYSTEMS. *The Garbage-First Garbage Collector* . Disponível em: <<http://www.oracle.com/technetwork/java/javase/tech/g1-intro-jsp-135488.html>>.

9. ORACLE CORPORATION. *Tuning the Memory Management System*. Disponível em: <http://download.oracle.com/docs/cd/E13150_01/jrockit_jvm/jrockit/geninfo/diagnos/memman.html>.
10. TENE, Gil. *Azul's Pauseless Garbage Collector*. 2010. Disponível em: <http://www.artima.com/lejava/articles/azul_pauseless_gc.html>.
11. ORACLE CORPORATION. *Javadoc do system.gc()*. Disponível em: <[http://download.oracle.com/javase/7/docs/api/java/lang/System.html#gc\(\)](http://download.oracle.com/javase/7/docs/api/java/lang/System.html#gc())>.
12. DARCY, Joseph. *Project Coin: Updated ARM Spec*. 2010. Disponível em: <http://blogs.sun.com/darcy/entry/project_coin_updated_arm_spec>.
13. PRINTEZIS, Tony. *How to Handle Java Finalization's Memory-Retention Issues*. 2007. Disponível em: <<http://java.sun.com/developer/technicalArticles/javase/finalization>>.
14. ORACLE CORPORATION. *Javadoc do java.lang.ref*. Disponível em: <<http://download.oracle.com/javase/7/docs/api/java/lang/ref/package-summary.html>>.
15. ORACLE CORPORATION. *Frequently Asked Questions About the Java HotSpot VM*. Disponível em: <http://www.oracle.com/technetwork/java/hotspotfaq-138619.html#compiler_types>.
16. ORACLE CORPORATION. *Ergonomics in the 5.0 Java™ Virtual Machine*. Disponível em: <<http://java.sun.com/docs/hotspot/gc5.0/ergo5.html>>.
17. CHUNG, Mandy. *Monitoring and Managing Java SE 6 Platform Application*. 2006. Disponível em: <<http://java.sun.com/developer/technicalArticles/J2SE/monitoring/>>.
18. LINDHOLM, Tim; YELLIN, Frank. *The Java Virtual Machine Specification: Loading, Linking, and Initializing*. Disponível em: <http://java.sun.com/docs/books/jvms/second_edition/html/ConstantPool.doc.html>.
19. JBOSS DOCUMENTATION. *ClassCastExceptions - I'm Not Your Type*. Disponível em: <http://docs.jboss.org/jbossas/docs/Server_Configuration_Guide/4/html/Class>Loading_and_Types_in_Java-ClassCastExceptions___Im_Not_Your_Type.html>.
20. SUN MICROSYSTEMS. *JAXP Compatibility Guide for the J2SE 5 Platform*. Disponível em: <http://download.oracle.com/javase/1.5.0/docs/guide/xml/jaxp/JAXP-Compatibility_150.html>.
21. SILVEIRA, Paulo. *Java 6, as APIs de XML, Webservices e classloaders*. 2007. Disponível em: <<http://blog.caelum.com.br/java-6-as-apis-de-xml-webservices-e-classloaders>>.

22. BROCK, Adrian. *Jboss ClassLoader History*. Disponível em: <<http://www.jboss.org/community/wiki/JBossClassLoaderHistory>>.
23. LECOMTE, Julien. *Yahoo UI Compressor and Java Class Loader*. Disponível em: <<http://www.julienlecomte.net/blog/2008/10/80/>>.
24. ALVESTAD, Magnus. *Memory leaks where the classloader cannot be garbage collected*. Disponível em: <<http://victor-jan.blogspot.com/2010/05/memory-leaks-where-classloader-cannot.html>>.
25. SZEGEDI, Attila. *A day in the life of a memory leak hunter*. 2005. Disponível em: <<http://www.szegeedi.org/articles/memleak.html>>.

CAPÍTULO 3

1. MARTIN, Robert C. *The Interface Segregation Principle*. 1996. Disponível em: <<http://www.objectmentor.com/resources/articles/isp.pdf>>.
2. MARTIN, Robert C. *Stability*. 1997. Disponível em: <<http://www.objectmentor.com/resources/articles/stability.pdf>>.
3. VENNERS, Bill. *Design Principles from Design Patterns – A Conversation with Erich Gamma, Part III*. 2005. Disponível em: <<http://www.artima.com/lejava/articles/designprinciples.html>>.
4. BLOCH, Joshua. *Effective Java*. 2 ed. Nova Jersey: Addison-Wesley, 2008.
5. FOWLER, Martin. *Refactoring: Improving the Design of Existing Code*. Massachusetts: Addison-Wesley, 1999.
6. GAMMA, Erich; HELM, Richard; JOHNSON, Ralph; VLISSIDES, John M. *Design Patterns: elements of reusable object-oriented software*. Massachusetts: Addison-Wesley, 1994.
7. MARTIN, Robert C. *The Dependency Inversion Principle*. 1996. Disponível em: <<http://www.objectmentor.com/resources/articles/dip.pdf>>.
8. VENNERS, Bill. *Inheritance versus composition: Which one should you choose?* 1998. Disponível em: <<http://www.javaworld.com/javaworld/jw-11-1998/jw-11-techniques.html>>.
9. HERMAN, Eric. *Composition vs. Inheritance in Java*. Disponível em: <<http://www.tiedyedfreaks.org/eric/CompositionVsInheritance.html>>.
10. SUMPTION, Bernie. *Inheritance is evil, and must be destroyed*. 2007. Disponível em: <<http://www.berniecode.com/writing/inheritance/>>.

11. MARTIN, Robert C. *The Liskov Substitution Principle*. 1996. Disponível em: <www.objectmentor.com/resources/articles/lsp.pdf>.
12. SILVEIRA, Guilherme. *Compondo seu comportamento: herança, chain of responsibility e interceptors*. 2010. Disponível em: <<http://blog.caelum.com.br/compondo-seu-comportamento-heranca-chain-of-responsibility-e-interceptors>>.
13. MARTIN, Robert C. *The open-closed principle*. 1996. Disponível em: <<http://www.objectmentor.com/resources/articles/ocp.pdf>>.
14. GOETZ, Brian. *Java theory and practice: To mutate or not to mutate?* 2003. Disponível em: <<http://www.ibm.com/developerworks/java/library/j-jtp02183.html>>.
15. BLANDY, Joe; BLOCH, Joshua. *Why doesn't Java have an immutable date class?* 2003. Disponível em: <<http://discuss.fogcreek.com/joelonsoftware3/default.asp?cmd=show&ixPost=73959>>
16. ORACLE CORPORATION. *Javadoc do método valueof(int) da classe integer*. Disponível em: <[http://download.oracle.com/javase/7/docs/api/java/lang/Integer.html#valueOf\(int\)](http://download.oracle.com/javase/7/docs/api/java/lang/Integer.html#valueOf(int))>.
17. SUN MICROSYSTEMS. *Código fonte da classe string*. Disponível em: <<http://hg.openjdk.java.net/jdk6/jdk6/jdk/file/536cbf2d9d0e/src/share/classes/java/lang/String.java>>.
18. FOWLER, Martin. *Fluent interface*. 2005. Disponível em: <<http://martinfowler.com/bliki/FluentInterface.html>>.
19. BOOCH, Grady. *Object-oriented analysis and design with applications*. Nova Jersey: Addison-Wesley Professional, 1993.
20. HOLUB, Allen. *Why getter and setter methods are evil*. 2003. Disponível em: <<http://www.javaworld.com/javaworld/jw-09-2003/jw-0905-toolbox.html>>.
21. SHARP, Alec. *Smalltalk by example: the developer's guide*. New York: McGraw Hill, 1997.
22. FOWLER, Martin. *Anemic domain model*. 2003. Disponível em: <<http://martinfowler.com/bliki/AnemicDomainModel.html>>.
23. CALÇADO, Phillip. *Objetos fantoches*. Disponível em: <<http://fragmental.com.br/wiki/index.php?title=Fantoches>>.
24. COMPUTERWORLD. *A to Z of Programming Languages: Smalltalk-80*. 2010. Disponível em: <<http://www.computerworld.com.au/article/352182/>>

25. FOWLER, Martin. *GetterEradication*. 2006. Disponível em: <<http://martinfowler.com/bliki/GetterEradicator.html>>.
26. EVANS, Eric. *Domain-driven design: Tackling complexity in the heart of software*. Boston: Addison-Wesley, 2003.
27. CALÇADO, Phillip. *How to write a repository*. 2010. Disponível em: <<http://fragmental.tw/2010/12/23/how-to-write-a-repository>>.
28. FOWLER, Martin. *Domain-Specific Language*. Disponível em: <<http://martinfowler.com/bliki/DomainSpecificLanguage.html>>.
29. FOWLER, Martin. *BusinessReadableDSL*. 2008. Disponível em: <<http://martinfowler.com/bliki/BusinessReadableDSL.html>>.

CAPÍTULO 4

1. MARTIN, Robert C. *Design principles and design patterns*. 2000. Disponível em: <http://www.objectmentor.com/resources/articles/Principles_and_Patterns.pdf>.
2. FOWLER, Martin. *UML Distilled: a brief guide to the standard object modeling language*. Boston: Addison-Wesley, 2003.
3. MARTIN, Robert C. *The Single Responsibility Principle*. 1995. Disponível em: <<http://www.objectmentor.com/resources/articles/srp.pdf>>.
4. DIJKSTRA, Edsger W. *On the role of scientific thought*. 1974. Disponível em: <<http://userweb.cs.utexas.edu/users/EWD/transcriptions/EWD04xx/EWD447.html>>.
5. LARMAN, Craig. *Applying UML and Patterns: an introduction to object-oriented analysis and design and iterative development*. 3. ed. Nova Jersey: Prentice Hall PTR, 2004.
6. NORTH, Dan; HELLESØY, Aslak. *PicoContainer Design Patterns*. Disponível em: <<http://www.picocontainer.org/patterns.html>>.
7. MARTIN, Robert C. *OO Design Quality Metrics - An Analysis Of Dependencies*. 1994. Disponível em: <<http://www.objectmentor.com/resources/articles/oodmetrc.pdf>>.
8. HAMMANT, Paul. *Inversion of control history*. Disponível em: <<http://www.picocontainer.org/inversion-of-control-history.html>>.
9. SWEET, Richard E. The Mesa Programming Environment. In: *Proceedings of the ACM SIGPLAN 85 symposium on Language issues in programming environments*. Nova York: ACM, 1985. p. 216-229.

10. O'BRIEN, Larry. *Design patterns 15 years later: An interview with Erich Gamma, Richard Helm, and Ralph Johnson*. 2009. Disponível em: <<http://www.informit.com/articles/article.aspx?p=1404056>>.
11. MARTIN, Robert C. Dependency Injection Inversion. Disponível em: <<http://blog.objectmentor.com/articles/2010/01/17/dependency-injection-inversion>>.
12. PRYCE, Nat. "Dependency injection" Considered Harmful. 2011. Disponível em: <<http://www.natpryce.com/articles/000783.html>>.
13. BONÉR, Jonas. *Real-world Scala: Dependency Injection (DI)*. Disponível em: <<http://jonasboner.com/2008/10/06/real-world-scala-dependency-injection-di.html>>.
14. KUNG, Fabio. *Ruby and Dependency Injection in a Dynamic World*. Disponível em: <<http://fabiokung.com/2010/05/06/ruby-and-dependency-injection-in-a-dynamic-world/>>.
15. PICOCONTAINER. *PicoContainer site*. Disponível em: <<http://www.picocontainer.org/>>.
16. FOWLER, Martin. *Inversion of Control Containers and the Dependency Injection Pattern*. Disponível em: <<http://martinfowler.com/articles/injection.html>>.
17. LADDAD, Ramnivas. *AspectJ in action: Enterprise AOP with Spring Applications*. 2. ed. Connecticut: Manning, 2009.

CAPÍTULO 5

1. MYERS, Glenford. *The Art of Software Testing*, 2. ed. Nova Jersey: John Wiley & Sons, 2004.
2. MARTIN, Robert. *Railsconf 2010 Keynote – Cucumber Scripts and Testing Business Rules Through the UI*. 2010 (38 min). Disponível em: <<http://www.youtube.com/watch?v=mslMLp5bQD0>, 2010>.
3. NORTH, Dan; HELLESØY, Aslak. *PicoContainer Design Patterns*. Disponível em: <<http://www.picocontainer.org/patterns.html>>.
4. FREEMAN, Steve; PRYCE, Nat. *Growing object-oriented software, guided by tests*. Nova Jersey: Addison-Wesley Professional, 2009.
5. GAMMA, Erich; HELM, Richard; JOHNSON, Ralph; VLISSIDES, John M. *Design Patterns: elements of reusable object-oriented software*. Massachusetts: Addison-Wesley, 1994.

6. BECK, Kent. *Test-driven development: by example*. Boston: Addison-Wesley, 2002.
7. AMBLER, Scott. *Introduction to Test Driven Design (TDD)*. 2003. Disponível em: <<http://www.agiledata.org/essays/tdd.html>>.
8. FOWLER, Martin. *Anemic Domain Model*. Disponível em: <<http://martinfowler.com/bliki/AnemicDomainModel.html>>.
9. CALÇADO, Phillip. *How to write a repository*. 2010. Disponível em: <<http://fragmental.tw/2010/12/23/how-to-write-a-repository>>.
10. MARTIN, Robert C. *Agile Software Development: Principles, Patterns, and Practices*. Nova Jersey: Prentice Hall, 2002.
11. FEATHERS, Michael. *The Deep Synergy Between Testability and Good Design*. 2007 (50:49 min). Disponível em: <<http://www.vimeo.com/15007792>>.
12. SHORE, James. *The Art of Agile Development*. Califórnia: O'Reilly Media, 2007.
13. SILVEIRA, Guilherme. *Branches e integração contínua: o problema de feature branches*. 2011. Disponível em: <<http://blog.caelum.com.br/branches-e-integracao-continua-o-problema-de-feature-branches>>.
14. SILVEIRA, Guilherme. *Martin Fowler and Jez Humble on Continuous Delivery*. 2011. Disponível em: <<http://www.infoq.com/interviews/jez-humble-martin-fowler-cd>>.
15. BOTTCH, Evan. *Continuous Integration – Single Code Line*. 2010. Disponível em: <<http://evan.bottch.com/2010/05/27/continuous-integration-single-code-line>>.
16. HUMBLE, Jez; FARLEY, David. *Continuous delivery: reliable software releases through build, test, and deployment automation*. Nova Jersey: Addison-Wesley, 2010.
17. SILVEIRA, Guilherme. *Deploy contínuo: pois integração contínua não basta*. 2010. Disponível em: <<http://www.infoq.com/br/presentations/deploy-continuo-guilherme>>.
18. TOMCAT WEBSITE. *Tomcat manager how to*. Disponível em: <<http://tomcat.apache.org/tomcat-7.0-doc/manager-howto.html>>.
19. PANT, Rajiv. *Organizing a Web Technology Department*. 2009. Disponível em: <<http://www.rajiv.com/blog/2009/03/17/technology-department>>.
20. FOWLER, Martin. *Blue Green Deployment*. 2010. Disponível em: <<http://martinfowler.com/bliki/BlueGreenDeployment.html>>.

CAPÍTULO 6

1. FOWLER, Martin. *Patterns of Enterprise Application Architecture*. Boston: Addison-Wesley, 2002. p. 103.
2. HAYDEN, David. *Web applications: N-Tier vs. N-Layer*. 2005. Disponível em: <<http://davidhayden.com/blog/dave/archive/2005/07/22/2401.aspx>>.
3. JAIN, Pranshu. *Layers and Tiers*. 2006. Disponível em: <<http://pranshujain.wordpress.com/2006/09/15/layers-and-tiers>>.
4. FOWLER, Martin. *Errant architectures*. 2003. Disponível em: <<http://www.drdoobs.com/184414966>>.
5. AVRAM, Abel. *Practicing Agility in Application Architecture*. 2008. Disponível em: <<http://www.infoq.com/news/2008/11/Agile-Architecture>>.
6. FOWLER, Martin. *First Law of Distributed Object Design*. Disponível em: <<http://martinfowler.com/bliki/FirstLaw.html>>.
7. GOOGLE PAGE SPEED. *Web Performance Best Practices*. Disponível em: <http://code.google.com/speed/page-speed/docs/rules_intro.html>.
8. DRISCOLL, Jim. *Servlet history*. 2005. Disponível em: <http://weblogs.java.net/blog/driscoll/archive/2005/12/servlet_history_1.html>.
9. BURBECK, Steve. *Applications Programming in Smalltalk-80™: How to use Model-View-Controller (MVC)*. 1987. Disponível em: <<http://st-www.cs.illinois.edu/users/smarch/st-docs/mvc.html>>.
10. ALUR, Deepak; MALKS, Dan; CRUPI, John. *Core J2EE Patterns: Best Practices and Design Strategies*. 2. ed. Nova Jersey: Prentice Hall PTR, 2003.
11. SUN MICROSYSTEMS. *Core J2EE Patterns Front Controller*. Disponível em: <<http://java.sun.com/blueprints/corej2eepatterns/Patterns/FrontController.html>>.
12. FORD, Neal. *Evolutionary architecture and emergent design: Evolutionary Architecture*. 2010. Disponível em: <<http://www.ibm.com/developerworks/java/library/j-eaed10/index.html>>.
13. BROWN, Don. *My History of Struts 2*. 2006. Disponível em: <http://www.oreillynet.com/onjava/blog/2006/10/my_history_of_struts_2.html>.
14. JAVA BLUEPRINTS. *Model-View-Controller*. 2000. Disponível em: <<http://www.oracle.com/technetwork/java/mvc-detailed-136062.html>>.
15. HILLENIUS, Eelco. *Wicket is not suited for websites*. 2011. Disponível em:

- <<http://www.jtict.com/blog/wicket-isnt-suited-for-websites>>.
16. PENCHIKALA, Srini. *JDBC 4.0 Enhancements in Java SE 6*. 2006. Disponível em: <<http://onjava.com/pub/a/onjava/2006/08/02/jjdbc-4-enhancements-in-java-se-6.html>>.
 17. SILVEIRA, Paulo. *A java.net.SocketException Broken Pipe*. 2009. Disponível em: <<http://blog.caelum.com.br/a-java-net-socketexception-broken-pipe>>.
 18. BARCIA, Roland. *Tired of hand coding JDBC? Use iBatis as a data mapping framework instead*. 2005. Disponível em: <http://www.ibm.com/developerworks/websphere/techjournal/0510_col_barcia/0510_col_barcia.html>.
 19. AMBLER, Scott. *The Object-Relational Impedance Mismatch*. Disponível em: <<http://www.agiledata.org/essays/impedanceMismatch.html>>.
 20. SILVEIRA, Paulo. *JPA com Hibernate: herança e mapeamentos*. 2007. Disponível em: <<http://blog.caelum.com.br/jpa-com-hibernate-heranca-e-mapeamentos>>.
 21. GNANAM, Simson. *How inheritance works in Hibernate*. 2008. Disponível em: <<http://simsonlive.wordpress.com/2008/03/09/how-inheritance-works-in-hibernate>>.
 22. PATRICIO, Anthony. *A Short Primer on Fetching Strategies*. 2010. Disponível em: <<http://community.jboss.org/wiki/AShortPrimerOnFetchingStrategies>>.
 23. SILVEIRA, Paulo. *Enfrentando a LazyInitializationException no Hibernate*. 2009. Disponível em: <<http://blog.caelum.com.br/enfrentando-a-lazyinitializationexception-no-hibernate>>.
 24. PARTINGTON, Vincent. *JPA Implementation Patterns: Data Access Objects*. 2009. Disponível em: <<http://blog.xebia.com/2009/03/09/jpa-implementation-patterns-data-access-objects>>.
 25. SILVEIRA, Paulo. *GUJ: Acesso ao EntityManager: através de um DAO ou diretamente?* 2009. Disponível em: <<http://www.guj.com.br/posts/list/133946.java>>.
 26. PARTINGTON, Vincent. *JPA Implementation Patterns: Service Facades and Data Transfers Objects*. 2009. Disponível em: <<http://blog.xebia.com/2009/05/11/jpa-implementation-patterns-service-facades-and-data-transfers-objects>>.
 27. SUN MICROSYSTEMS. *Java Persistence Blueprints*. 2007. Disponível em:

- <<https://blueprints.dev.java.net/bpcatalog/ee5/persistence>>.
28. DEMICHEL, Linda. *Java Persistence 2.0 Proposed Final Draft*. 2009. Disponível em: <http://blogs.sun.com/ldemichiel/entry/java_persistence_2_0_proposed>.
 29. Silveira, Paulo. *Os 7 hábitos dos desenvolvedores Hibernate e JPA altamente eficazes*. 2008. Disponível em: <<http://blog.caelum.com.br/os-7-habitos-dos-desenvolvedores-hibernate-e-jpa-altamente-eficazes>>.
 30. JBOSS HIBERNATE. *Configuração do C3P0 no Hibernate*. Disponível em: <<https://www.hibernate.org/214.html>>.
 31. EHCACHE WEBSITE. *Hibernate Caching*. Disponível em: <<http://ehcache.org/documentation/hibernate.html>>.
 32. JBOSS HIBERNATE. *Query Cache*. 2009. Disponível em: <<http://community.jboss.org/wiki/QueryCache>>.
 33. HOFF, Todd. *Strategy: Break Up the Memcache Dog Pile*. 2009. <<http://highscalability.com/strategy-break-memcache-dog-pile>>.
 34. SCHMIDT, Douglas C.; HARRISON, Tim. *Double-Checked Locking: An Optimization Pattern for Efficiently Initializing and Accessing Thread-safe Objects*. 1996. Disponível em: <<http://www.cs.wustl.edu/~schmidt/PDF/DC-Locking.pdf>>.
 35. BLOCH, Joshua. Concurrency. In: _____. *Effective Java*. 2. ed. Nova Jersey: Addison-Wesley, 2008. p. 283.
 36. GOETZ, Brian. *Java Concurrency in Practice*. Nova Jersey: Addison-Wesley, 2006. p. 348.
 37. VUONG, Lan. *Extreme Transaction Processing Patterns: Write-Behind Caching*. 2009. Disponível em: <<http://www.infoq.com/articles/write-behind-caching>>.
 38. ZAMARREÑO, Galder. *Using Infinispan as JPA/Hibernate Second Level Cache Provider*. 2010. Disponível em: <<https://docs.jboss.org/author/display/ISPN/Using+Infinispan+as+JPA-Hibernate+Second+Level+Cache+Provider>>.
 39. JBOSS HIBERNATE. *Batch Processing*. Disponível em: <<http://docs.jboss.org/hibernate/stable/core/reference/en/html/batch.html>>.
 40. JBOSS HIBERNATE. *Improving Performance*. Disponível em: <<http://docs.jboss.org/hibernate/stable/core/reference/en/html/performance.html>>.
 41. CALÇADO, Phillip. *Internal Data Transfer Objects*. Disponível em: <<http://fragmental.tw/2008/12/22/internal-data-transfer-objects>>.

42. SUN MICROSYSTEMS. *Core J2EE Patterns Session Façade*. Disponível em: <<http://java.sun.com/blueprints/corej2eepatterns/Patterns/SessionFacade.html>>.
43. IBM. *DPD Chronology*. Disponível em: <http://www-03.ibm.com/ibm/history/exhibits/dpd50/dpd50_chronology2.html>.
44. RICHARDS, Mark; MONSON-HAEFEL, Richard; CHAPPELL, David A. *Java Message Service*. 2. ed. Califórnia: O'Reilly, 2009. p. 11-12.
45. Idem, p. 4.
46. Idem, p. 87-105.
47. Idem, p. 63-86.
48. GOTTFRID, Derek. *Self-Service, Prorated Supercomputing Fun!* 2007. Disponível em: <<http://open.blogs.nytimes.com/2007/11/01/self-service-prorated-super-computing-fun>>.
49. CARR, Nicholas. *The Big Switch: Rewiring the World, from Edison to Google*. Nova Iorque: W.W. Norton, 2008.
50. MALLYA, Subraya. *Netflix's Move to Amazon EC2 - explained*. 2010. Disponível em: <<http://www.prudentcloud.com/cloud-computing-technology/netflixs-move-to-amazon-ec2-explained-20102010>>.
51. HOFF, Todd. *Netflix: Use Less Chatty Protocols in the Cloud - Plus 26 fixes*. 2010. Disponível em: <<http://highscalability.com/blog/2010/12/20/netflix-use-less-chatty-protocols-in-the-cloud-plus-26-fixes.html>>.
52. LOPES, Sérgio. *Vivendo no Cloud: a Infraestrutura Externa da Caelum em 11 soluções*. 2011. Disponível em: <<http://blog.caelum.com.br/vivendo-no-cloud-a-infraestrutura-externa-da-caelum-em-11-solucoes>>.
53. KALLEN, Nick. *Big Data in Real-Time at Twitter*. 2010. Disponível em: <<http://www.slideshare.net/nkallen/q-con-3770885>>.
54. SHOUP, Randy. *eBays Architectural Principles*. 2008. Disponível em: <<http://www.slideshare.net/deimos/randy-shoup-ebays-architectural-principles>>.
55. BROWNE, Julian. *Brewer's CAP Theorem*. 2009. Disponível em: <<http://www.julianbrowne.com/article/viewer/brewers-cap-theorem>>.
56. THARAKAN, Royans K. *Brewer's CAP Theorem on Distributed Systems*. 2010. Disponível em: <<http://www.royans.net/arch/brewers-cap-theorem-on-distributed-systems>>.
57. GOOGLE APP ENGINE WEBSITE. *Choosing a Datastore for Google App Engine*. Disponível em: <<http://code.google.com/appengine/docs/java/datastore/hr/>>.

58. PRITCHETT, Dan. *BASE: An Acid Alternative*. 2008. Disponível em: <<http://queue.acm.org/detail.cfm?id=1394128>>.

CAPÍTULO 7

1. SIEGEL, David. *Pull: the power of the Semantic Web to transform your business*. Nova Iorque: Portfolio, 2009.
2. ANDREWS, Gregory R. *Foundations of Multithreaded, Parallel, and Distributed Programming*. Massachusetts: Addison-Wesley, 2000.
3. TANENBAUM, Andrew S. *Computer networks*. 2. ed. Nova Jersey: Prentice Hall, 1988.
4. SOWA, John. *Sowa's Law of Standards*. 1991. Disponível em: <<http://www.jfsowa.com/computer/standard.htm>>.
5. RICHARDSON, Leonard. *Justice will take us millions of intricate moves*. 2008. Disponível em: <<http://www.crummy.com/writing/speaking/2008-QCon/act1.html>>.
6. W3C. *Simple Object Access Protocol (SOAP) 1.1*. 2000. Disponível em: <<http://www.w3.org/TR/2000/NOTE-SOAP-20000508>>.
7. INNOQ. *Web Services Standards as of Q1 2007*. Disponível em: <<http://www.innoq.com/resources/ws-standards-poster/>>.
8. JORDAN, Diane; EVDEMON, John. *Web Services Business Process Execution Language Version 2.0*. 2007. Disponível em: <<http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf>>.
9. SNELL, James. *Restful RPC*. 2010. Disponível em: <<http://www.snellspace.com/wp/2010/05/restful-rpc>>.
10. TILKOV, Stefan. *Devoxx 2010 Enterprise It vs. WWW. 2010*. Disponível em: <<http://www.devoxx.com/display/Devoxx2K10/Enterprise+IT+vs.+WWW>>
11. THOLOMÉ, Eric. *Well Earned Retirement for Soap Search*. 2009. Disponível em: <<http://googlecode.blogspot.com/2009/08/well-earned-retirement-for-soap-search.html>>.
12. TILKOV, Stefan. *Einstieg in REST*. 2010. Disponível em: <<http://www.heise.de/developer/artikel/Episode-17-Einstieg-in-REST-921652.html>>.
13. *Ws-* is to REST as Theory is to Practice*. 2007. Disponível em: <<http://www.25hoursaday.com/weblog/2007/11/15/WSIsToRESTAsTheoryIsToPractice.aspx>>.

14. W3C. *SOAP Version 1.2 Part 0: Primer (Second Edition)*. 2007. Disponível em: <<http://www.w3.org/TR/2007/REC-soap12-part0-20070427>>.
15. ORCHARD, David. *Versioning XML Vocabularies*. 2003. Disponível em: <<http://www.xml.com/pub/a/2003/12/03/versioning.html>>.
16. SINGHAM, Roy et al. *The Thoughtworks Anthology: essays on Software Technology and Innovation*. Carolina do Norte: Pragmatic Bookshelf, 2008.
17. GOOGLE. *Protocol buffers*. Disponível em: <<http://code.google.com/apis/protocolbuffers/docs/overview.html>>.
18. TILKOV, Stefan et al. *SOA Manifesto*. 2009. Disponível em: <<http://www.soa-manifesto.org>>.
19. HOHPE, Gregor; WOLF, Bobby. *Enterprise integration patterns: designing, building, and deploying messaging solutions*. Boston: Addison-Wesley, 2003.
20. TILKOV, Stefan. *Einstieg in REST*. 2010. Disponível em: <<http://www.heise.de/developer/artikel/Episode-17-Einstieg-in-REST-921652.html>>.
21. TILKOV, Stefan. *REST und HTTP: Einsatz der architektur des web für integrationsszenarien*. dpunkt Verlag, 2009.
22. RICHARDSON, Leonard; RUBY, Sam. *RESTful Web Services*. Farnham: O'Reilly, 2007.
23. FIELDING, Roy. *Architectural Styles and the Design of Network Based Software Architectures*. 2000. Disponível em: <<http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>>.
24. WEBBER, Jim; PARASTATIDIS, Savas; ROBINSON, Ian, *REST in practice: Hypermedia and systems architecture*. O'Reilly Media, 2010.
25. ALLAMANJARU, Subbu. *RESTful Web Services Cookbook*. Califórnia: O'Reilly, 2010.
26. NOTTINGHAM, Mark. *Caching Tutorial for Web Authors and Webmasters*. 1998-2010. Disponível em: <http://www.mnot.net/cache_docs/>.
27. FOWLER, Martin. *Segmentation by Freshness*. 2008. Disponível em: <<http://martinfowler.com/bliki/SegmentationByFreshness.html>>.
28. TILKOV, Stefan; HADLEY, Marc; SANDOZ, Paul. *JSR 311 Final: Java API for RESTful Web Services*. 2008. Disponível em: <<http://www.infoq.com/news/2008/09/jsr311-approved>>
29. CHINNICI, Roberto; POTOCIAR, Marek. *JAX-RS 2.0: The java API for RESTful Web Services*. 2011. Disponível em: <<http://jcp.org/en/jsr/detail?id=339>>.

30. GREGORIO, J. et al. *URI Template*. 2010. Disponível em: <<http://tools.ietf.org/html/draft-gregorio-uritemplate-04>>.
31. ALLAMANJARU, Subbu. *RESTful URIs*. 2010. Disponível em: <<http://www.subbu.org/blog/2008/07/restful-uris>>.
32. NELSON, T. H. Complex Information Processing: a File Structure for the Complex, the Changing and the Indeterminate. In: *ACM '65 Proceedings of the 1965 20th national conference*. Nova Iorque: ACM, 1965.
33. PARASTATIDIS, Savas et al. The Role of Hypermedia in Distributed System Development. In: *WS-REST '10 Proceedings of the First International Workshop on RESTful Design*. Nova Iorque: ACM, 2010.
34. ROBINSON, Ian. *Using Typed Links to Forms*. 2010. Disponível em: <<http://iansrobinson.com/2010/09/02/using-typed-links-to-forms>>.
35. AMUNDSEN, Mike. *Designing a Hypermedia API with JSON*. 2010. Disponível em: <<http://www.amundsen.com/blog/archives/1054>>.
36. ZYP, K.; COURT, G. *A JSON Media Type for Describing the Structure and Meaning of JSON Documents*. 2009-2010. Disponível em: <<http://tools.ietf.org/html/draft-zyp-json-schema-03>>.
37. NOTTINGHAM, Mark. *Web Linking*. 2010. Disponível em: <<http://tools.ietf.org/html/rfc5988>>.
38. CLINTON, DeWitt et al. *Open search specification*. 2000. Disponível em: <<http://www.opensearch.org/Specifications/OpenSearch/1.1>>.
39. DUSSEAULT, L.; LAB, Linden; SNELL, J. *Patch Method for HTTP*. 2010. Disponível em: <<http://tools.ietf.org/html/rfc5789>>.
40. IANA. *MIME Media Types Registry*. 2007. Disponível em: <<http://www.iana.org/assignments/media-types/index.html>>.
41. MURATA, M.; ST. LAURENT, S.; KOHN, D. *XML Media Types*. 2001. Disponível em: <<http://www.ietf.org/rfc/rfc3023.txt>>.
42. MICROFORMATS. *Plain Old XML Considered Harmful*. 2008. Disponível em: <<http://microformats.org/wiki/plain-old-xml-considered-harmful>>.
43. BRAY, Tim. *Don't Invent XML Languages*. 2006. Disponível em: <<http://www.tbray.org/ongoing/When/200x/2006/01/08/No-New-XML-Languages>>
44. ÇELIK, Tantek. *Facebook adds hCalendar and hCard Microformats to Millions of Events*. 2011. Disponível em: <<http://microformats.org/2011/02/17/facebook>>

- adds-hcalendar-hcard>.
45. DANYLCHUK, Mike; SHETTI, Nitin. *Introducing a New Rich Snippets Format: Events*. 2010. Disponível em: <<http://googlewebmastercentral.blogspot.com/2010/01/introducing--new-rich-snippets-format.html>>.
 46. MICROFORMATS. *Google adds support for hCalendar and hRecipe Rich Snippets*. 2010. Disponível em: <<http://microformats.org/2010/04/28/google-adds-support-for-hcalendar-and-hrecipe-rich-snippet>>
 47. AMUNDSEN, Mike. *RESTful Profiling : XHTML-based Web APIs*. 2010. Disponível em: <<http://www.amundsen.com/blog/archives/1043>>.
 48. MANOLA, Frank; MILLER, Eric. *RDF Primer*. 2004. Disponível em: <<http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>>.
 49. GREGORIO, Joe. *Do we need WADL?* 2007. Disponível em: <<http://bitworking.org/news/193/Do-we-need-WADL>>.
 50. AMUNDSEN, Mike. *Why WADL when you can run?* 2010. Disponível em: <<http://www.amundsen.com/blog/archives/1067>>.
 51. MILLER, Libby; BRICKLEY, Dan. *The friend of a friend project*. 2000. Disponível em: <<http://www.foaf-project.org/>>.
 52. NOTTINGHAM, Mark. *HTTP Cache-Control Extensions for Stale Content*. 2010. Disponível em: <<http://tools.ietf.org/html/rfc5861>>.
 53. GRIGORIK, Ilya. *Asynchronous HTTP Cache Validations*. 2008. Disponível em: <<http://www.igvita.com/2008/10/07/asynchronous-http-cache-validations>>.