
Referências Bibliográficas

- [1] Java user groups.
<http://java.sun.com/community/usergroups/>.
- [2] Job trends: struts, hibernate e spring.
<http://www.indeed.com/jobtrends?q=spring+javava>.
- [3] Magnus Alvestad. Memory leaks where the classloader cannot be garbage collected.
<http://opensource.atlassian.com/confluence/spring/pages/viewpage.action?pageId=2669>.
- [4] Scott Ambler. The object-relational impedance mismatch.
<http://www.agiledata.org/essays/impedanceMismatch.html>.
- [5] Roland Barcia. Tired of hand coding jdbc? use ibatis as a data mapping framework instead.
http://www.ibm.com/developerworks/websphere/techjournal/0510_olbarcia/0510_olbarcia.html, 2005.
- [6] Grady Booch. *Object-Oriented Analysis and Design with Applications*. Addison-Wesley Professional, 1993.
- [7] Adrian Brock. Jboss classloader history.
<http://www.jboss.org/community/wiki/JBossClassLoaderHistory>.
- [8] Phillip Calçado. Objetos fantoches.
<http://fragmental.com.br/wiki/index.php?title=Fantoches>.
- [9] Microsoft Corporation. Extension methods (c programming guide).
<http://msdn.microsoft.com/en-us/library/bb383977.aspx>.
- [10] Aslak Hellesoy Dan North. Pico container design patterns.
<http://www.picocontainer.org/patterns.html>.
- [11] Joshua Bloch: Second Edition. *Effective Java*. Prentice Hall PTR, 2008.
- [12] Ehcache. Hibernate caching.
<http://ehcache.org/documentation/hibernate.html>.

- [13] Jason English. História do java: Java technology in your future.
<http://java.sun.com/java2/whatis/1996/storyofjava.html>.
- [14] Ralph Johnson John M. Vlissides Erich Gamma, Richard Helm. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional Computing Series, 1994.
- [15] Eric Evans. *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison-Wesley Professional, 2003.
- [16] Fog Creek forum. Why doesn't java have an immutable date class?
<http://discuss.fogcreek.com/joelonsoftware/default.asp?cmd=showixPost=73959>.
- [17] Martin Fowler. Anemic domain model.
<http://martinfowler.com/bliki/AnemicDomainModel.html>.
- [18] Martin Fowler. Fluent interface.
<http://martinfowler.com/bliki/FluentInterface.html>.
- [19] Martin Fowler. Inversion of control containers and the dependency injection pattern.
<http://martinfowler.com/articles/injection.html>.
- [20] Paulo Silveira Fábio Kung. Repository: seu modelo mais orientado a objeto.
<http://blog.caelum.com.br/2007/06/09/repository-seu-modelo-mais-orientado-a-objeto/>.
- [21] Simson Gnanam. How inheritance works in hibernate.
<http://simsonlive.wordpress.com/2008/03/09/how-inheritance-works-in-hibernate/>.
- [22] Brian Goetz. To mutate or not to mutate?
<http://www.ibm.com/developerworks/java/library/j-jtp02183.html>.
- [23] Brian Goetz. Java theory and practice: To mutate or not to mutate?
<http://www.ibm.com/developerworks/java/library/j-jtp02183.html>, 2003.
- [24] Paul Hammant. Inversion of control history.
<http://www.picocontainer.org/inversion-of-control-history.html>.
- [25] Eric Herman. Composition vs inheritance.
<http://www.tiedyedfreaks.org/eric/CompositionVsInheritance.html>.
- [26] JBoss Hibernate. Batch processing.
<http://docs.jboss.org/hibernate/stable/core/reference/en/html/batch.html>.
- [27] JBoss Hibernate. Configuração do c3p0 no hibernate.
<https://www.hibernate.org/214.html>.
- [28] JBoss Hibernate. Improving performance.
<http://docs.jboss.org/hibernate/stable/core/reference/en/html/performance.html>.

- [29] Allen Holub. Why getter and setter methods are evil.
<http://www.javaworld.com/javaworld/jw-09-2003/jw-0905-toolbox.html>.
- [30] Guy Steele Gilad Bracha James Gosling, Bill Joy. *Java Language Specification: Third Edition*. Addison-Wesley, 2005.
- [31] Craig Larman. Protected variation: The importance of being closed. *IEEE Software*, 2001.
- [32] Julien Lecomte. Yahoo ui compressor and java class loader.
<http://www.julienlecomte.net/blog/2008/10/80/>.
- [33] Robert C. Martin. The principles of ood.
<http://www.butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod>.
- [34] Robert C. Martin. Oo design quality metrics: An analysis of dependencies. *comp.lang.c++.*, 1994.
- [35] Jon Masamitsu. Our collectors (sun).
http://blogs.sun.com/jonthecollector/entry/our_collectors.
- [36] Sun Microsystems. Código fonte da classe string.
<http://hg.openjdk.java.net/jdk6/jdk6/jdk/file/536cbf2d9d0e/src/share/classes/java/lang/String.java>.
- [37] Sun Microsystems. The garbage-first garbage collector.
http://java.sun.com/javase/technologies/hotspot/gc/g1_intro.jsp.
- [38] Sun Microsystems. Java se 6 hotspot[tm] virtual machine garbage collection tuning.
http://java.sun.com/javase/technologies/hotspot/gc/gc_tuning_6.html.
- [39] Sun Microsystems. Javadoc do método valueOf(int) da classe integer.
[http://java.sun.com/javase/6/docs/api/java/lang/Integer.html#valueOf\(int\)](http://java.sun.com/javase/6/docs/api/java/lang/Integer.html#valueOf(int)).
- [40] Sun Microsystems. Jaxp compatibility guide for the j2se 5 platform.
http://java.sun.com/j2se/1.5.0/docs/guide/xml/jaxp/JAXP-Compatibility_150.html.
- [41] Sun Microsystems. Memory management in the java hotspot virtual machine. 2006.
- [42] Sun Microsystems. Java persistence blueprints.
<https://blueprints.dev.java.net/bpcatalog/ee5/persistence/>, 2007.
- [43] Oracle. Tuning the memory management system - jrockit documentation.
http://download.oracle.com/docs/cd/E13150_01/jrockit_jvm/jrockit/geninfo/diagnos/memman.html.
- [44] Vincent Partington. Jpa implementation patterns: Wrap-up.
<http://blog.xebia.com/2009/07/13/jpa-implementation-patterns-wrap-up/>.
- [45] Vincent Partington. Jpa implementation patterns: Data access objects.
<http://blog.xebia.com/2009/03/09/jpa-implementation-patterns-data-access-objects/>, 2009.

- [46] Vincent Partington. Jpa implementation patterns: Service facades and data transfers objects.
<http://blog.xebia.com/2009/05/11/jpa-implementation-patterns-service-facades-and-data-transfers-objects/>, 2009.
- [47] Srini Penchikala. Jdbc 4.0 enhancements in java se 6.
<http://onjava.com/pub/a/onjava/2006/08/02/jjdbc-4-enhancements-in-java-se-6.html>, 2006.
- [48] Bruno R. Preiss. Mark-and-sweep garbage collection.
<http://www.brpreiss.com/books/opus5/html/page424.html>.
- [49] Tony Printezis. Garbage collection in the java hotspot virtual machine (generational algorithm).
<http://www.devx.com/Java/Article/21977>.
- [50] Mark Radford. Singleton - the anti-pattern!
<http://accu.org/index.php/journals/337>.
- [51] Paulo Silveira. Java 6, as apis de xml, webservices e classloaders.
<http://blog.caelum.com.br/2007/12/17/java-6-as-apis-de-xml-webservices-e-classloaders/>.
- [52] Paulo Silveira. A java.net.socketexception broken pipe.
<http://blog.caelum.com.br/2009/10/19/a-java-net-socketexception-broken-pipe/>.
- [53] Paulo Silveira. Jpa com hibernate: Herança e mapeamentos.
<http://blog.caelum.com.br/2007/03/04/jpa-com-hibernate-heranca-e-mapeamentos/>.
- [54] Paulo Silveira. Os 7 hábitos dos desenvolvedores hibernate e jpa altamente eficazes.
<http://blog.caelum.com.br/2008/01/28/os-7-habitos-dos-desenvolvedores-hibernate-e-jpa-altamente-eficazes/>.
- [55] Paulo Silveira. Singletons e static: perigo a vista.
<http://blog.caelum.com.br/2006/08/08/singletons-e-static-perigo-a-vista/>.
- [56] Paulo Silveira. Enfrentando a lazyinitializationexception no hibernate.
<http://blog.caelum.com.br/2009/10/13/enfrentando-a-lazyinitializationexception-no-hibernate/>, 2009.
- [57] Paulo Silveira. Guj: Acesso ao entitymanager: através de um dao ou diretamente?
<http://www.guj.com.br/posts/list/133946.java>, 2009.
- [58] Bernie Sumption. Inheritance is evil, and must be destroyed.
<http://www.berniecode.com/writing/inheritance/>.
- [59] Attila Szegedi. A day in the life of a memory leak hunter.
<http://www.szegedi.org/articles/memleak.html>.

- [60] Frank Yellin Tim Lindholm. The java virtual machine specification: Loading, linking, and initializing.
http://java.sun.com/docs/books/jvms/second_edition/html/ConstantPool.doc.html.
- [61] Frank Yellin Tim Lindholm. *Java Virtual Machine Specification: Second Edition*. Prentice Hall PTR, 1999.
- [62] Robert Tolksdorf. Programming languages for the java virtual machine jvm.
<http://www.is-research.de/info/vmlanguages/>.
- [63] Bill Venners. Design principles from design patterns - a conversation with erich gamma, part iii.
<http://www.artima.com/lejava/articles/designprinciples.html>.
- [64] Bill Venners. Inheritance versus composition: Which one should you choose?
<http://www.javaworld.com/javaworld/jw-11-1998/jw-11-techniques.html>.
- [65] Simon Willison. A re-introduction to javascript.
https://developer.mozilla.org/en/A_re-introduction_to_JavaScript, 2006.
- [66] Benjamin Zorn. The measured cost of conservative garbage collection. *Software — Practice and Experience - VOL. 23(7)*, 1993.