

**FELIPE RODRIGUES DO PRADO
JOÃO PAULO NAKAJIMA PEREIRA**

**DESENVOLVIMENTO MODULAR DE SOFTWARE
UTILIZANDO OSGI**

**UNIVERSIDADE DO VALE DO SAPUCAÍ
POUSO ALEGRE
2015**

**FELIPE RODRIGUES DO PRADO
JOÃO PAULO NAKAJIMA PEREIRA**

**DESENVOLVIMENTO MODULAR DE SOFTWARE
UTILIZANDO OSGI**

Trabalho de Conclusão de Curso Sistemas de
Informação da Universidade do Vale do
Sapucaí como requisito para obtenção de título
de bacharel em SISTEMAS DE
INFORMAÇÃO

Orientador: Me. Márcio Emílio Cruz Vono de
Azevedo.

**UNIVERSIDADE DO VALE DO SAPUCAÍ
POUSO ALEGRE
2015**

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
BSD	<i>Berkeley Software Distribution</i>
CSS	<i>Cascading Style Sheet</i>
EJB	<i>Enterprise JavaBeans</i>
HTML	<i>HyperText Markup Language</i>
ICC	<i>Inatel Competence Center</i>
IDE	<i>Integrated Development Environment</i>
IoT	<i>Internet of Things</i>
JDBC	<i>Java Database Connectivity</i>
JEE	<i>Java Platform, Enterprise Edition.</i>
JMS	<i>Java Message Service</i>
JPA	<i>Java Persistence API</i>
JVM	<i>Java Virtual Machine</i>
NTT	<i>Nippon Telegraph and Telephone</i>
OSGi	<i>Open Services Gateway initiative</i>
SENAC	Serviço Nacional de Aprendizagem Comercial
SQL	<i>Structured Query Language</i>
SRA	<i>Systems Research and Applications Corporation</i>
UNIFEI	Universidade Federal de Itajubá
UNIVAS	Universidade do Vale do Sapucaí
XHTML	<i>Extensible HyperText Markup Language</i>
XML	<i>Extensible Markup Language</i>
W3C	<i>World Wide Web Consortium</i>

SUMÁRIO

INTRODUÇÃO.....	6
REFERÊNCIAS.....	9

INTRODUÇÃO

O desenvolvimento de um *software* não envolve apenas métodos de programação. Para desenvolvê-lo é necessário preocupar-se com planejamento, engenharia, metodologia e tecnologia a ser utilizada. Esses fatores influenciam na manutenção, atualização e expansão do mesmo. Dessa forma, definir tais fatores é fundamental para que o *software* seja flexível a mudanças. Pensando dessa maneira, será demonstrado um modelo de desenvolvimento modular, utilizando a especificação OSGi.

OSGi, abreviação para *Open Services Gateway Initiative*, possibilita o desenvolvimento de *softwares* em módulos, disponibilizando o gerenciamento dos mesmos e fornece facilidades na manutenção e expansão da aplicação.

Segundo Fernandes (2009), um sistema modular possui algumas propriedades. Deve ser autocontido, os módulos podem ser incluídos, retirados, instalados ou desinstalados. Outra propriedade é ter alta coesão. Um módulo deve cumprir apenas sua finalidade, ou seja, deve fazer somente as funções que lhe foram atribuídas. O baixo acoplamento é outra propriedade muito importante. Um módulo não precisa se preocupar com implementações de outros módulos que interagem com ele, além de permitir alterá-lo sem a necessidade de atualizar os outros.

Muitos *softwares* são desenvolvidos de maneira semelhante à modularização, divididos em partes, tendo cada parte uma responsabilidade. Porém não são realmente modularizados, ou seja, não atendem ao conceito de modularização como descrito acima. Dois exemplos de projetos de *softwares* que foram desenvolvidos utilizando a especificação OSGi e que atendem a definição de modularização, são, IDE¹ Eclipse, ferramenta de desenvolvimento de softwares e o GlassFish, servidor de aplicações JEE².

O uso da modularização, com certeza, traz grandes benefícios para o desenvolvimento e manutenção de um *software*. Poder parar parte de uma aplicação para fazer uma manutenção ou poder instalar novas funcionalidades, garantindo que todas as outras partes restantes continuem funcionando normalmente, seria uma característica notável da aplicação, principalmente em grandes empresas, onde não se tem o luxo de parar o sistema para atualizar

1 IDE – Abreviação para *Integrated Development Environment*.

2 JEE – Abreviação para *Java Platform, Enterprise Edition*.

um relatório.

Fernandes (2009) forneceu uma visão geral sobre OSGi, mostrando seus benefícios e salientando a importância da plataforma Java possuir um melhor suporte à modularidade, até demonstrando com um exemplo simples as premissas e vantagens do OSGi.

Mayworm (2010) demonstra a tecnologia OSGi no contexto de aplicações distribuídas, permitindo a disponibilização de seus serviços remotamente, integrando com diferentes *frameworks* de *middleware* para o desenvolvimento de aplicações empresariais.

Malcher (2008) apresenta um modelo de componentes adaptável para se usar em ambientes distribuídos. Também analisa alguns aspectos, como modelo de distribuição, transparência, descoberta de novos módulos disponíveis, desempenho e *performance* dos mesmos. Afirma ainda que para se escolher entre um modelo de distribuição – *deployment* local ou execução remota – é necessário analisar o contexto e o objetivo da aplicação, pois cada um se adapta melhor a determinadas situações e ambientes de execução.

Portanto, após despertar um grande interesse pelo modelo de desenvolvimento modular, será realizado estudos sobre a especificação OSGi, e ainda, o desenvolvimento de uma aplicação modular que demonstre tal modelo de desenvolvimento.

O presente trabalho possui como objetivo geral:

- Demonstrar o modelo de desenvolvimento modular utilizando a especificação OSGi para aplicações empresariais;

e objetivos específicos:

- Pesquisar as melhores práticas e *frameworks* que contribuem para a produtividade no desenvolvimento modularizado;
- Desenvolver uma aplicação que exemplifique o modelo de desenvolvimento modularizado através da especificação OSGi;
- Obter através dos resultados uma conclusão sobre as vantagens do modelo de desenvolvimento modular.

Diante dos vários segmentos e constantes mudanças empresariais, novos *softwares* são desenvolvidos e precisam estar preparados para acompanhar as modificações que ocorrem nos processos de uma empresa. Dessa forma o desenvolvimento separado em módulos seria uma

solução para atender empresas de diferentes setores em vez de criar um *software* para cada ramo empresarial. Além disso a utilização de módulos em um sistema torna mais flexível sua manutenção, pois a mesma é realizada nos módulos, não afetando o restante da aplicação.

O trabalho tem relevância na visão acadêmica diante da aprendizagem da tecnologia OSGi, além de agregar conceitos sobre o desenvolvimento de softwares de forma modularizada. E ainda, por ser pouco utilizada no mercado devido à sua complexidade de aprendizagem, este trabalho visa reunir informações e disponibilizar uma documentação com práticas de programação e vantagens que essa tecnologia pode oferecer.

Desenvolvedores de *softwares* procuram sempre a utilização de novas práticas e tecnologias produtivas. O modelo de desenvolvimento modular através da tecnologia OSGi oferece diversos benefícios que ajudam na expansão e manutenção de uma aplicação. Isto também se torna útil para as empresas, pois para novas funcionalidades seriam criados módulos, e a manutenção do sistema seria realizada somente no módulo específico, sem a necessidade de parar o restante do sistema.

Como a adoção dessa nova prática e tecnologia proporciona a reutilização de código, consequentemente não haverá a necessidade do desenvolvimento repetido de funcionalidades, economizando tempo, energia elétrica e recursos computacionais, algo que se tornou de grande valor atualmente, visando o desenvolvimento harmonioso e sustentável entre o ser humano e a natureza.

REFERÊNCIAS

ANICHE, Maurício. **Test-Driven Development: Teste e Design no Mundo Real**. São Paulo: Casa do Código, 2012.

ANICHE, Maurício. **TDD**. 2014. Disponível em <http://tdd.caelum.com.br/>. Acesso em 11 de março, 2015.

APACHE. **OSGi Frequently Asked Questions**. 2015. Disponível em <http://felix.apache.org/documentation/tutorials-examples-and-presentations/apache-felix-osgi-faq.html>. Acesso em 16 de junho, 2015.

APACHE. **What is maven**. 2015. Disponível em <http://maven.apache.org/what-is-maven.html>. Acesso em 16 de junho, 2015.

APPOLINÁRIO, Fábio. **Dicionário de metodologia: um guia para a produção do conhecimento científico**. São Paulo: Atlas, 2004.

BORBA, Paulo. **Aspectos de Modularização**. 2015. Disponível em <http://www.di.ufpe.br/~java/graduacao961/aulas/aula4/aula4.html>. Acesso em 21 de junho, 2015.

BARTLETT, Neil. **OSGi In Practice**. 2009.

BAUER, Christian; KING, Gavin. **Hibernate in action**. Greenwich: Manning Publications Co., 2005.

BOSSCHAERT, David. **OSGi in Depth**. Shelter Island: Manning Publications Co, 2012

CAELUM. **Apostila Desenvolvimento Web com HTML, CSS e JavaScript**. 2015. Disponível em <https://www.caelum.com.br/apostila-html-css-javascript/javascript-e-interatividade-na-web/>. Acesso em 08 de março, 2015.

CAELUM. **Apostila Java e Orientação a Objetos**. 2015. Disponível em <http://www.caelum.com.br/apostila-java-orientacao-objetos/>. Acesso em 08 de março, 2015.

CLARO, Daniela Barreiro; SOBRAL, João Bosco Manguiera. **Programação em Java**. Santa Catarina: Copyleft Pearson Education, 2008.

DEITEL, Harvey Matt; DEITEL, Paul John. **Java How to Program**. 8. ed. Edson Furmankiewicz. São Paulo: Pearson Prentice Hall, 2010.

DEVMEDIA. **Introdução ao Spring Framework**. 2015. Disponível em <http://www.devmedia.com.br/introducao-ao-spring-framework/26212>. Acesso em 10 de março, 2015.

DEVMEDIA. **Novidades do GlassFish 3.1**. 2011. Disponível em: <http://www.devmedia.com.br/novidades-do-glassfish-3-1-artigo-java-magazine-91/21124>. Acesso em 19 de junho, 2015.

DURHAM, Alan; JOHNSON, Ralph. A Framework for Run-time Systems and its Visual Programming Language. In: **OBJECT-ORIENTED PROGRAMMING SYSTEMS, LANGUAGES, AND APPLICATIONS**. San Jose, CA, 1996, p. 20-25.

FERNANDES, Leonardo. **OSGi e os benefícios de uma Arquitetura Modular**. 37.ed. 2009. p. 27-35.

FILHO, Walter dos Santos. **Introdução ao Apache Maven**. Belo Horizonte: Eteg Tecnologia da Informação Ltda, 2008.

GAMA, Kiev. **Uma visão geral sobre a plataforma OSGi**. 2008. Disponível em <https://kievgama.wordpress.com/2008/11/24/um-pouco-de-osgi-em-portugues/>. Acesso em 09 de março, 2015.

GIL, Antônio Carlos. **Como elaborar projetos de pesquisa**. São Paulo: Atlas, 2002.

GONCALVES, Antonio. **Beginning Java EE 6 Platform with GlassFish 3**. Nova York: Springer Science+Business Media, LCC. 2010.

KIOSKEA. **Condução de reunião**. 2014. Disponível em <http://pt.kioskea.net/contents/579-conducao-de-reuniao>. Acesso em 16 de abril, 2015.

KNOERNSCHILD, Kirk. **Java Application Architecture: Modularity Patterns with Examples Using OSGi**. Crawfordsville: Pearson Education, 2012.

LUCENA, Fábio Nogueira de. **Introdução ao Equinox**. 2010. Disponível em <https://code.google.com/p/exemplos/wiki/equinox>. Acesso em 09 de março, 2015.

MADEIRA, Marcelo. **OSGi – Modularizando sua aplicação**. 2009. Disponível em <https://celodemelo.wordpress.com/2009/11/12/osgi-modularizando-sua-aplicacao/>. Acesso em 09 de março, 2015.

MALCHER, Marcelo Andrade da Gama. **OSGi Distribuído: deployment local e execução remota**. Monografia de Seminários de Sistemas Distribuídos. Pontifícia Universidade Católica do Rio de Janeiro, 2008.

MAUJOR. **Site sobre CSS e Padrões Web: Por que CSS?**. 2015. Disponível em: <http://www.maujor.com/index.php>. Acesso em: 08 de março, 2015.

MAYWORM, Marcelo. **OSGi Distribuída: Uma Visão Geral**. 42.ed. 2010. p. 60-67.

MILANI, André. PostgreSQL: **Guia do Programador**. São Paulo: Novatec Editora, 2008.

Mozilla Developer Network. **CSS**. 2015. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/CSS>. Acesso em 08 de março, 2015.

Mozilla Developer Network. **HTML**. 2014. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTML>. Acesso em 07 de março, 2015.

Mozilla Developer Network. **JavaScript**. 2015. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>. Acesso em 08 de março, 2015.

Mozilla Developer Network. **JavaScript language resources**. 2014. Disponível em https://developer.mozilla.org/en-US/docs/Web/JavaScript/Language_Resources. Acesso em 08 de março, 2015.

OSGI ALLIANCE. **OSGi**. 2015. Disponível em <http://www.osgi.org/Main/HomePage>. Acesso em 08 de março, 2015.

ORACLE. **Difference between GlassFish Open Source and Commercial Editions**. 2011. Disponível em https://blogs.oracle.com/GlassFishForBusiness/entry/difference_between_glassfish_open_source. Acesso em 20 de junho, 2015.

Pivotal Software. **Preface**. 2015. Disponível em <http://docs.spring.io/osgi/docs/current/reference/html/preface.html>. Acesso em 10 de março, 2015.

Pivotal Software. **Spring Framework**. 2015. Disponível em <http://projects.spring.io/springframework/>. Acesso em 10 de março, 2015.

SILVA, Maurício Samy. **CSS3: Desenvolva aplicações web profissionais com uso dos poderosos recursos de estilização das CSS3**. São Paulo: Novatec Editora, 2012.

SILVA, Maurício Samy. **HTML5: A linguagem de marcação que revolucionou a web**. São Paulo: Novatec Editora, 2011.

SILVA, Maurício Samy. **JavaScript: Guia do Programador**. São Paulo: Novatec Editora, 2010.

SOUZA, Arthur Câmara; AMARAL, Hugo Richard; LIZARDO, Luis Eduardo O. **PostgreSQL: uma alternativa para sistemas gerenciadores de banco de dados de código aberto**. In: Anais do Congresso Nacional Universidade, EAD e Software Livre, 2012.

USP. **Fundamentos do projeto de software**. 2015. Disponível em <http://www.pcs.usp.br/~pcs722/98/Objetos/bases.html>. Acesso em 21 de junho, 2015.

VAZ, Rodrigo Cardoso. **JUnit – Framework para testes em Java**. Palmas: Centro Universitário de Palmas, 2003.

W3C. **About W3C**. 2015. Disponível em <http://www.w3.org/Consortium/>. Acesso em 07 de

março, 2015.

W3C. **What is CSS?**. 2015. Disponível em <http://www.w3.org/Style/CSS/>. Acesso em 08 de março, 2015.