

# **APLICAÇÃO DOS CONCEITOS DE PROGRAMAÇÃO ORIENTADA A ASPECTO NO DESENVOLVIMENTO DE SISTEMAS**

**Rafael Alessandro Casachi**  
**Almir Rogério Camolesi**

*POA; Programação; Desenvolvimento; Aspecto*

## **INTRODUÇÃO**

A evolução no desenvolvimento de sistemas nos últimos anos, influenciou na criação de ferramentas mais eficazes como a Programação Orientada a Aspecto (POA) criada por Gregor Kiczales em 1997 na PARC.

A POA consiste em um método de separação de códigos em relação ao comportamento. Ou seja, os códigos funcionais, que fazem parte do esqueleto do sistema, são separados dos códigos sistêmicos, que são aqueles que oferecem suporte aos interesses funcionais.

O código sistêmico é programado em um aspecto que irá agrupar os pontos de atuação e os adendos deste código.

Os pontos de atuação definiram os pontos no código principal onde será aplicado o aspecto, a este ponto chamamos de ponto de junção.

O adendo é o comportamento de um aspecto. Os adendos são semelhantes aos métodos na programação orientada a objetos, ou seja, são blocos de códigos que serão executados no momento definido no adendo. Assim, um combinador unirá os dois tipos de interesses em um único código para ser compilado.

## **OBJETIVO**

O objetivo deste trabalho é mostrar que a aplicação dos conceitos do paradigma de programação a aspectos no desenvolvimento de softwares pode ser mais eficiente, mais organizado e mais rápido.

## **MATERIAL E MÉTODOS**

Os métodos utilizados para o desenvolvimento do trabalho foi a consulta de materiais técnicos para a elaboração de um estudo sobre o paradigma de programação orientado a aspectos e o desenvolvimento de um estudo de caso que simulasse o desenvolvimento de software com o paradigma.

## **RESULTADOS E DISCUSSÃO**

A utilização da programação orientada a aspecto no desenvolvimento de sistemas melhora a

visibilidade dos códigos, pois possibilita a modularização de códigos sistêmicos. Como os códigos estão separados em blocos fora da aplicação principal, fica mais fácil a reutilização de códigos, diminuindo a quantidade de linhas de código, auxiliam também na legibilidade do programa.

Com o programa organizado desta forma, podemos evitar o problema de diversos interesses emaranhados em uma única classe e de interesses espalhados por diversas classes. A solução desses problemas possibilita que o analista foque na qualidade dos requisitos levantados.

O processo de produção e de manutenção é beneficiado com o este método, pois cada interesse é independente do resto da aplicação, o que torna a implantação de novas funcionalidades mais simples e a manutenção torna-se mais eficaz. Assim, o custo e o prazo de entrega dos projetos são melhorados e é garantida a qualidade dos serviços.

## CONCLUSÃO

A utilização do paradigma de programação orientada a aspecto na construção de um sistema melhora o processo de análise de requisitos, diminui os custos de programação e manutenção, melhora a legibilidade do código, organiza o aplicativo e diminui o tempo de desenvolvimento de um projeto.

## REFERÊNCIAS

KICZALES, Gregor; LAMPING, John; MENDHEKAR, Anurag; MAEDA, Chris; LOPES, Cristina Videira; LOINGTIER, Jean-Marc. Aspect-Oriented Programming. In: European Conference on Object-Oriented Programming (ECOOP), 06,1997. Finlândia. Anais Springer-Verlag LNCS 1241, 06, 1997

KICZALES, Gregor; MEZINI, Mira. **Separation of Concerns with Procedures, Annotations, Advices and Pointcuts**. Vancouver, Canada: University of British Columbia

GOETTEN, Vicente J.; WINCK, Diogo V. **AspectJ – Programação Orientada a Aspectos com Java**. São Paulo: Novatec Editora, 2006.

KRUPA, Artur. **Analyse Aspect-Oriented Software Approach and Its Application**. Athabaska, Alberta, Canadá: Athabaska University, 2010.

HEILEMAN, Gregory L. **Data Structures, Algorithms, and Object-Oriented Programming**. Singapore: McGraw-Hill, 1996.

SAFONOV, Vladimir O. **Using Aspect-Oriented Programming for Trustworthy Software Development**. New Jersey: Wiley-Interscience, 2008.

BODKIN, Ron; LADDAD, Ramnivas. **Zen and the art of Aspect-Oriented Programming**. Linux Magazine, April, 2004.

THE ASPECT TEAM. **The AspectJ Programming Guide**. Xerox Corporation, Palo Alto Research Center (PARC), Palo Alto, CA, Estados Unidos. Disponível em: <"www.eclipse.org/aspectJ">. Acessado em: 06 abr. 2011.

UBAYASHI, Naoyasu; MASUHARA, Hidehiko; MORIYAMA, Genki; TAMAI, Tetsuo. **A Parameterized Interpreter for Modeling Different AOP Mechanisms**. Fukuoka, Japão: Kyushu Institute of Technology, 2005.