



**Aluno:** Fagner Fernandes Candido da Silva (ffcs@cin.ufpe.br)  
**Orientador:** Henrique Emanuel Mostaert Rebêlo (herm@cin.ufpe.br)

Recife, 30 de Setembro de 2015.

## Sumário

1. Introdução .....	3
2. Objetivos .....	4
3. Cronograma .....	4
4. Possíveis avaliadores .....	4
5. Referências .....	5
6. Assinaturas .....	5

## 1. Introdução

O processo de desenvolvimento de software vem sofrendo drásticas mudanças ao longo das últimas décadas. Essas mudanças, segundo Matkovic e Tumbas [1], são consequência da evolução não só, do processo de criação e adaptação de produtos de software, como também dos modelos e das metodologias utilizadas na engenharia de software. Para Laddad [2] os projetos de software estão cada vez mais complexos e a tendência é que essa complexidade aumente nos próximos anos.

Uma forma de lidar com essa complexidade crescente é a modularização dos sistemas. Quando se divide um problema em partes menores, a concepção da solução ocorre de modo mais rápido e eficiente. Entretanto existem preocupações (*concerns*) que afetam as funcionalidades do sistema em vários níveis, conforme descrito por Laddad.

Segurança, logging, pooling de recursos, caching, monitoramento de performance, concorrência e gerenciamento de transações são exemplos de funcionalidades que afetam vários módulos do sistema, sendo considerados assim características transversais (do inglês, *cross-cutting*).

Em 1996, Kiczales et al. [3] concebeu uma nova técnica de programação chamada programação orientada a aspectos (do inglês, *aspect-oriented programming*) com intuito de solucionar esse problema cujo qual o paradigma orientado a objetos não pode resolver. Esse paradigma de programação orientada a aspectos é considerado ideal para esse tipo de separação, pois ele introduz uma nova unidade de modularização chamada de aspecto, onde cada aspecto pode ser responsável por cada uma dessas funcionalidades transversais.

De acordo com Bien [4] com a introdução da plataforma Java Enterprise Edition (JEE) 5, em 2006, o desenvolvimento web atual foi bastante simplificado. Graças a Java Specification Request 299 (JSR 299), foi criado o CDI (do inglês, *Contexts and Dependency Injection*) permitiu que várias dessas funcionalidades transversais pudessem ser resolvidas utilizando as próprias APIs (do inglês, *Application Programming Interface*) da plataforma enterprise do Java.

Sendo assim, é necessário avaliar se o uso do paradigma de programação orientada a aspectos ainda é relevante no cenário de desenvolvimento web atual.

## 2. Objetivos

Este trabalho tem como objetivo principal avaliar a viabilidade do uso da programação orientada a aspectos no desenvolvimento de software no ambiente web atual. Para tanto, iremos utilizar como base a aplicação Healthwatcher. Ela será reformulada para o padrão de desenvolvimento web atual através da utilização de tecnologias aderentes a plataforma JEE 6.

Após essa conversão, essa mesma aplicação será alterada para que possa utilizar aspectos, mais especificamente, a implementação AspectJ. Essa implementação foi escolhida devido a linguagem de programação utilizada para desenvolver o Healthwatcher ter sido Java. Além disso, segundo a própria fundação Eclipse [4], o AspectJ é fácil de aprender e de usar.

## 3. Cronograma

Nessa seção, a tabela 1 descreve o cronograma de atividades previsto para desenvolvimento desse trabalho de graduação ao longo desse semestre.

Atividade	Setembro	Outubro	Novembro	Dezembro	Janeiro
Elaboração da Proposta					
Healthwatcher JEE6					
Healthwatcher JEE6 com Aspectos					
Pesquisa bibliográfica					
Testes e experimentos					
Análise dos resultados					
Elaboração do relatório					
Preparação para apresentação e defesa					

Tabela 1- Cronograma de Trabalho de Graduação

## 4. Possíveis avaliadores

Os possíveis avaliadores para o resultado a ser obtido no final de todas as etapas descritas nesse documento são:

- Leopoldo Motta Teixeira;

- Fernando Castor.

## 5. Referências

[1] MATKOVIĆ, Predrag; TUMBAS, Pere. A Comparative Overview of the Evolution of Software Development Models. **International Journal Of Industrial Engineering And Managem Ent**. Online, p. 163-172. 2010. <[http://www.iim.ftn.uns.ac.rs/casopis/volume1/ijiem\\_vol1\\_no4\\_4.pdf](http://www.iim.ftn.uns.ac.rs/casopis/volume1/ijiem_vol1_no4_4.pdf)>. Acesso 09 set. 2010.

[2] LADDAD, Ramnivas. **AspectJ in Action**. 2. ed. United States Of America: Manning Publications Co, 2010.

[3] KICZALES et al. **Aspect-Oriented Programming**. <http://www.cs.ubc.ca/~gregor/papers/kiczales-ECOOP1997-AOP.pdf>. Acesso 22 set. 2015.

[4] BIEN, Adam. **Simplicity by Design**. <http://www.oracle.com/au/products/database/o11java-195110.html>. Acesso 10 set. 2015.

[5] ECLIPSE. **AspectJ**. <https://eclipse.org/aspectj/>. Acesso 10 set. 2015.

## 6. Assinaturas

---

Fagner Fernandes Candido da Silva  
**Orientando**

---

Henrique Emanuel Mostaert Rebêlo  
**Orientador**