

Resumo dos capítulos 3 e 4 da tese: Suporte ao Desenvolvimento e Uso de Frameworks e Componentes

Ricardo Pereira e Silva

OS capítulos específicos da tese em questão dissertam sobre o desenvolvimento e uso de *frameworks* orientados a objetos (OO), esclarecendo as metodologias de desenvolvimento existentes bem como as maneiras de documentá-la de modo a proporcionar o uso adequado dos *frameworks* de acordo com as diferentes necessidades dos seus usuários.

Frameworks OO

Frameworks Orientados a Objetos correspondem a um conjunto de classes que se adaptam ao desenvolvimento de aplicações de um domínio específico, oferecendo estruturas flexíveis para os pontos em que aplicações de um mesmo domínio diferem entre si.

Duas das principais características de um *framework* são alterabilidade e extensibilidade:

- **Alterabilidade:** Permite que os conceitos específicos das aplicações sejam suportados a partir de estruturas flexíveis do *framework*;
- **Extensibilidade:** Garante que o *framework* possa ser estendido no seu conjunto de funcionalidades, ou seja, tenha sua manutenibilidade garantida;

Um framework sempre depende de outros artefatos de software para existir, basicamente, as aplicações específicas de um domínio que o originaram e as aplicações geradas a partir dele.

Metodologias de Desenvolvimento de Frameworks OO

Muitos elementos do processo de desenvolvimento de um framework são comuns à diversas metodologias, podemos elencar estes elementos da seguinte forma:

- **Aquisição de conhecimento de domínio:** A primeira etapa é o estudo de aplicações desenvolvidas para o mesmo domínio do *framework* de modo a identificar generalidades e especificidades dentro das aplicações.
- **Construção da estrutura de classes:** Após a análise, é realizada a modelagem de classes. Esta modelagem pode ser dividida em cinco etapas iterativas:
 - **Generalização:** Criação de classes para as estruturas idênticas das aplicações;
 - **Flexibilização:** Localização das especificidades dos *frameworks*;
 - **Aplicação de meta padrões:** Criação de métodos *template* que podem ser flexibilizados através de métodos *hooks* para os pontos de flexibilização encontrados;
 - **Aplicação de Padrões de Projeto:** Incluir classes de um padrão (*design pattern*) já catalogado e conhecido para uma determinada estrutura do *framework*;
 - **Aplicação de Princípios de OO:** Aplicação de princípios específicos de Orientação a Objetos como uso de herança e composição.
- **Implementação:** Criação do *framework* em uma linguagem de programação;
- **Avaliação:** O *framework* deve ser avaliado a partir do desenvolvimento de aplicações do domínio para o qual foi criado;
- **Refinamento:** A partir do desenvolvimento de novas aplicações, o conhecimento sobre o

domínio aumentará, possibilitando a melhoria do *framework*.

Uso de Frameworks OO

Um framework deve facilitar o desenvolvimento de aplicações de um domínio. Se entender o framework for muito complexo, pode ser mais fácil desenvolver a aplicação sem o framework.

Uma das formas de garantir uma documentação que atenda às necessidades dos usuários do framework é adaptando a documentação à estas necessidades. Segue abaixo uma proposta de documentação específica para três tipos de usuários de frameworks:

Tipo de Usuário	Tipo de Documentação
Precisa decidir qual framework utilizar para o desenvolvimento de uma aplicação	Descrição do que pode e do que não pode ser feito com o framework.
Precisa descobrir como desenvolver uma aplicação	Receitas específicas de como utilizar um determinado recurso e/ou desenvolver uma aplicação.
Precisam conhecer em detalhes o projeto do framework para o desenvolvimento de aplicações complexas	<p>Podem ser utilizados diagramas, descrições mais detalhadas da estrutura das classes, código fonte e outras ferramentas.</p> <p>As questões chave para este entendimento são:</p> <ul style="list-style-type: none">• Quais classes usar: Quais classes concretas do framework podem ser utilizadas e quais devem ser criadas pelo desenvolvedor;• Quais métodos: devem ser redefinidos pelo desenvolvedor (métodos abstratos) e quais devem ser definidos pelo framework. Entra aqui também a definição dos métodos <i>template</i>;• O que os métodos fazem: nos métodos definidos pelo framework, qual o seu comportamento?

Conclusão

É essencial ao desenvolvimento de um framework o conhecimento do domínio a ser trabalhado de modo a identificar estruturas genéricas e as especificidades de cada uma. Isto é o coração do framework.

Após esta etapa e após a implementação de estruturas que suportem adequadamente a flexibilidade desejada é essencial a atenção dada à documentação, pois, pela própria existência de um *framework* como um artefato inter-relacionado com outros, a compreensão do seu uso deve ser clara, garantindo que o desenvolvimento de aplicações será efetivamente mais rápido e eficaz com o *framework* do que sem ele.