

# PROPOSTA DE DESENVOLVIMENTO DE SOFTWARE COM ORIENTAÇÃO A OBJETOS

Adriana L.Q. Mattoso  
Hélcio Blum  
COPPE/Sistemas - UFRJ

---

Durante os anos 80, o paradigma de orientação a objetos tem sido amplamente utilizado e discutido. Não existe, entretanto, um consenso com relação a um método orientado a objetos que possa ser adotado durante todo o ciclo de vida. Uma proposta de desenvolvimento de software com orientação a objetos é apresentada. Esta baseia-se em outras encontradas na literatura e busca aplicar os conceitos de orientação a objetos em todas as fases do ciclo de vida.

---

## 1. INTRODUÇÃO

Este trabalho tem por objetivo apresentar um roteiro para desenvolvimento de software tomando como base os conceitos de orientação a objetos. Sua origem se deu em estudos feitos nesta área, que levaram à motivação de experimentar-se na prática algumas das idéias encontradas na literatura. O roteiro apresentado está sendo utilizado em dois projetos em desenvolvimento na COPPE/Sistemas: um "Editor Gráfico para Apoio no Projeto de Sistemas" e um "Sistema de Gerência de Banco de Dados Orientado a Objetos".

O trabalho está dividido em três partes. Inicialmente são apresentados um método e um ciclo de vida ao qual este é aplicado. Posteriormente são apresentados alguns roteiros para documentação que tenham relação direta com o fato de se estar usando orientação a objetos. Finalmente são apresentadas

usando orientação a objetos. Finalmente são apresentadas algumas conclusões referentes à experiência de elaboração deste trabalho, além de algumas considerações referentes a possíveis problemas a serem enfrentados quando da utilização da proposta aqui apresentada.

## 2. MÉTODO

O método descrito é fruto da combinação de algumas propostas já existentes sobre o assunto e busca manter uma coerência com o paradigma da orientação a objetos.

O ciclo de vida adotado é baseado na filosofia de desenvolvimento por versões sucessivas [FAI 85]. A adoção desta estratégia tem por objetivo permitir que o produto seja especificado como um todo e projetado e construído em partes.

Para suportar tal estratégia, este ciclo de vida prevê atividades de integração das partes que forem sendo desenvolvidas e a consequente avaliação da versão resultante desta integração.

A seguir é apresentado o método proposto através da descrição das fases e atividades que compõem o ciclo de vida. Neste, foram intencionalmente omitidas algumas atividades que são inerentes ao processo de desenvolvimento, mas que não têm relação direta com a orientação a objetos.

## **FASE 1 - Definição**

### **Atividade: Entendimento do Problema e Definição de Objetivos**

Deve-se procurar entender todos os aspectos envolvidos no problema a ser resolvido, a fim de se definir os objetivos do produto a ser desenvolvido. Esta atividade tem como produto a elaboração dos capítulos 1 e 2 da Proposta de Desenvolvimento.

### **Atividade: Análise e Especificação de Requisitos**

A análise de requisitos deve ser feita tomando-se como base uma descrição informal das características do produto que se deseja implementar. Para tal, o texto deve ser examinado de acordo com os passos seguintes:

- i) destacar os substantivos existentes buscando-se identificar os objetos e classes, levando-se em conta a ocorrência de substantivos sinônimos que possam representar o mesmo objeto ou classe;
- ii) destacar os adjetivos associados aos objetos e classes identificados, buscando-se identificar seus atributos;
- iii) destacar os verbos associados aos objetos e classes identificados, buscando-se identificar as operações por estes sofridas ou realizadas.

A especificação dos requisitos deve tomar como base a análise anterior, e deve ser feita de acordo com os passos seguintes:

- i) reunir os objetos que tenham os mesmos atributos e operações em classes de objetos;

- ii) definir a interface das classes de objetos descrevendo as operações que podem ser aplicadas sobre seus objetos;
- iii) verificar a existência de atributos e/ou operações comuns a mais de uma classe de objetos, definindo uma hierarquia entre elas de acordo com o seguinte critério:
  - existindo uma classe de objetos cujo conjunto de atributos e/ou operações seja subconjunto dos atributos e/ou operações de uma outra classe qualquer, deve ser definida uma hierarquia entre elas de tal forma que a classe que envolva o conjunto com a quantidade maior de atributos e/ou operações assuma o papel de subclasse da outra, herdando desta tais características;
  - existindo uma interseção dos conjuntos de atributos e/ou operações de mais de uma classe, deve ser definida uma superclasse que as generalize, passando as classes originais a serem subclasses desta e a herdarem também desta as características comuns.

Esta atividade tem como produto a elaboração do capítulo 3 da Proposta de Desenvolvimento.

#### **Atividade: Planejamento dos Ciclos de Versões**

Definir e planejar a versão a ser construída.

## **FASE 2 - Projeto**

### **Atividade: Projeto da Arquitetura**

O projeto da arquitetura deve ser feito tomando-se como base a especificação de requisitos elaborada na fase anterior. Para cada classe de objetos identificada deve-se:

- i) definir os métodos para as operações identificadas especificando a sua lógica;
- ii) identificar e definir as variáveis de instância, especificando o seu tipo.

Esta atividade tem como produto a elaboração dos capítulos 1 e 2 da Especificação de Projeto.

### **Atividade: Escolha do Ambiente de Programação**

A escolha do ambiente de programação deve anteceder ao detalhamento do projeto uma vez que, dependendo do ambiente escolhido, é possível conhecer-se as classes nele existentes.

Este conhecimento é importante para a análise de reutilização de classes de objetos feita durante o detalhamento do projeto, já que em função do ambiente de programação a ser adotado, pode existir ou não uma biblioteca de classes disponíveis para serem reutilizadas.

### **Atividade: Detalhamento do Projeto**

O detalhamento do projeto deve ser feito tomando-se como base o projeto da arquitetura. Para cada classe de objetos deve-se:

- i) verificar se já existe na biblioteca de classes, alguma classe que possa ser usada para a implementação dos seus objetos;

- ii) caso exista, indicar a classe que será usada e as eventuais extensões que possam vir a ser necessárias (definição de novas variáveis de instância e/ou métodos, ou definição de nova subclasse);
- iii) caso não exista, decompor os seus objetos a fim de obter outros de menor complexidade e, recursivamente, executar para cada um destes as atividades Projeto da Arquitetura e Detalhamento do Projeto;
- iv) representar as trocas de mensagens entre classes de objetos, através de um diagrama.

Esta atividade tem como produto a elaboração do capítulo 3 da Especificação de Projeto.

### **FASE 3 - Construção**

#### **Atividade: Implementação**

Implementar os componentes do produto considerando que:

- i) caso seja utilizada uma linguagem de programação orientada a objetos, a implementação resume-se à criação das classes de objetos (variáveis de instância e métodos) no ambiente de programação, indicando sua(s) superclasse(s);
- ii) se forem usadas linguagens híbridas, a implementação deve ser feita através da codificação das classes de objetos em um programa propriamente dito, utilizando-se o tipo de dado que simula classes de objetos e reconhece, em alguns casos, a herança entre elas;
- iii) se forem usadas linguagens convencionais, a implementação deve ser feita através da codificação das classes de objetos em um programa propriamente dito, criando-se tipos de dados que simulem as classes de objetos e a herança entre elas.

### **Atividade: Depuração**

Para depurar cada classe de objetos codificada, deve ser simulada a sua execução de acordo com os seguintes passos:

- i) substituir as mensagens enviadas a outras classes por uma mensagem externa (exibição para o depurador), e pela inicialização dos parâmetros que estariam contidos na mensagem de volta;
- ii) ativar cada uma de suas operações através do envio de mensagens a elas, e verificar os resultados obtidos.

### **Atividade: Planejamento da Avaliação do Produto**

A avaliação do produto deve ser feita em função do processo de integração, isto é, o produto deve ser avaliado à medida que suas partes forem sendo integradas.

Existem duas estratégias para a integração das classes de objetos que compõem o produto. A primeira delas (de-fora-para-dentro) parte dos objetos mais externos (fronteiras do produto) e envolve a integração sucessiva dos objetos mais internos a este. Neste caso, para avaliar cada camada do produto deve ser usada a mesma técnica aplicada na atividade Depuração.

A outra alternativa de integração (de-dentro-para-fora) parte dos objetos mais internos (núcleo do produto) e envolve a integração sucessiva dos objetos mais externos a este. Neste caso, para avaliar cada camada do produto, devem ser simuladas as mensagens que são aceitas pelas classes de objetos que a compõem.

Contém a descrição detalhada das classes de objetos que implementam a interface como das internas ao produto. A apresentação deve ser feita graficamente através de diagramas e na forma de um dicionário de classes de objetos. Neste deve estar contidos, para cada classe,

## **FASE 4 - Integração e Avaliação**

### **Atividade: Avaliação do Produto**

Integrar e avaliar o produto de acordo com o definido na atividade Planejamento da Avaliação do Produto.

## **3. ROTEIROS DE DOCUMENTAÇÃO**

### **Proposta de Desenvolvimento**

1. Definição do problema
2. Objetivos do produto
3. Especificação de requisitos

#### **3.1 Contexto do produto**

Contém o contexto do produto apresentado graficamente através de um diagrama, e especificado através da descrição de sua funcionalidade e interface.

#### **3.2 Decomposição inicial**

Contém uma decomposição do contexto definido anteriormente que deve ser apresentado graficamente através de um diagrama, e especificado através da descrição da funcionalidade e interface de cada componente do produto.

### **Especificação de projeto**

1. Introdução
2. Projeto da arquitetura

Contém a descrição tanto das classes de objetos que implementam a interface como das internas ao produto. A apresentação deve ser feita graficamente através de diagramas e na forma de um dicionário de classes de objetos. Neste devem estar contidos, para cada classe,

sua hierarquia (superclasses e subclasses), as trocas de mensagens, variáveis de instância, métodos e, quando possível, suas instâncias.

### 3. Projeto detalhado

Contém, para cada classe de objetos, a definição dos tipos sobre os quais são implementadas suas variáveis de instância, e o detalhamento dos algoritmos de cada método a ser implementado.

## 4. CONCLUSÕES

O método proposto é resultante da combinação de propostas para projeto e desenvolvimento com orientação a objetos encontrados na literatura. Uma das adaptações feitas foi a aplicação, na atividade de análise e especificação de requisitos, de algumas técnicas propostas para projeto.

As dificuldades encontradas na elaboração desta proposta referem-se, de uma forma geral, à falta de definição de padrões de documentação e técnicas para testar, integrar e avaliar sistemas orientados a objetos.

Acredita-se que com a experiência adquirida nos dois projetos em desenvolvimento será possível avaliar o método proposto e ajustá-lo em função das dificuldades encontradas na sua utilização.

Recomenda-se que outros projetos que venham a ser desenvolvidos seguindo esta proposta, levem em consideração o grau de inovação inerente à utilização do método na elaboração de seus cronogramas.

## AGRADECIMENTOS

Gostaríamos de agradecer a Luiz Carlos Monte pelas críticas e sugestões feitas ao método proposto.

## BIBLIOGRAFIA

- [BOO 83] Booch, G. "Software Engineering with Ada"; B. Cummings Publishing; 1983
- [BOO 86] Booch, G. "Object-Oriented Development"; IEEE Trans. on Soft. Eng., vol. SE-12, n.2; fev./86
- [COH 84] Cohen, A. "Data Abstraction, data encapsulation and Object-Oriented Programming"; SIGPLAN Notices, V19-1; 1984
- [COX 84] Cox, B. "Object-Oriented Programming - A power tool for craftsman"; Unix Review; fev-mar/84
- [FAI 85] Fairley, R. "Software Engineering Concepts"; McGraw Hill; 1985
- [GOL 84] Goldberg, A. "SMALLTALK-80: The interactive programming environment"; Addison-Wesley; 1984
- [JAC 87] Jacobson, I. "Object-Oriented Development in an Industrial Environment"; Proc. OOPSLA; 1987
- [MON 87] Monte, L. C. "Avaliação da orientação a objetos em Projetos"; monografia do curso de Controle da Qualidade - COPPE Sistemas/UFRJ; 1987
- [ROC 87] Rocha, A.R. "Análise e Projeto Estruturado de Sistemas"; Ed. Campus; 1987
- [ROT 87] Rotemberg, H. "Programação Orientada a Objetos: Um enfoque da Engenharia de Software"; Dissertação de mestrado; PUC/RJ; 1987
- [TAK 88] Takahashi, T. "Introdução a Programação Orientada a Objetos"; III EBAI; 1988