

# FAST CASE

## UMA FERRAMENTA CASE PARA O DESENVOLVIMENTO VISUAL DE SISTEMAS ORIENTADOS A OBJETOS

Denis S. Silveira  
*e-mail:* [denis@cos.ufrj.br](mailto:denis@cos.ufrj.br)

Eber A. Schmitz  
*e-mail:* [eber@mbi.nce.ufrj.br](mailto:eber@mbi.nce.ufrj.br)

Instituto de Matemática / Núcleo de Computação Eletrônica  
Universidade Federal do Rio de Janeiro  
Caixa Postal 2324, CEP:20001-970,  
Rio de Janeiro, RJ - Brasil

### RESUMO

Este artigo apresenta uma ferramenta CASE que suporta um subconjunto da UML. A grande motivação para o nosso trabalho vem da ausência de uma ferramenta simples para ser utilizada no ensino das técnicas de análise e projeto orientados a objetos. O **FAST CASE** provê todas as funcionalidades necessárias para a construção dos modelos, tais como, o Modelo de Requisitos, suportando a técnica de casos de usos e, os Modelos de Análise e Projeto, suportando os modelos de Classes, Interações e de Ciclo de vida. O sistema, que é distribuído livremente, já vem sendo utilizado há mais de um ano por mais de uma centena de alunos.

**Palavras-chave:** CASE, UML, Engenharia de Software, Orientação à Objetos, Arquitetura de Software, Engenharia de Requisitos;

### 1 APRESENTAÇÃO

Uma construção complexa, tanto do ponto de vista da engenharia como da arquitetura, é na grande maioria das vezes centrada em modelos. Utilizamos modelos na construção de sistemas complexos, porque em geral não conseguimos compreender um sistema em sua totalidade. Um modelo é uma descrição da realidade que ressalta alguns aspectos em detrimento de outros. Cada tipo de modelo utiliza uma notação precisa e um conjunto de regras sintáticas e semânticas.

Os modelos de sistema de informação podem ser classificados em vários estilos, por exemplo baseado em fluxo de dados, relacionais e orientado a objetos. A orientação a objeto, na década de 90 se afirmou como o estilo de projeto mais utilizado no desenvolvimento de software. As grandes vantagens deste estilo de programação são: uma melhor representação do mundo real devido ao mapeamento simples que existe entre os objetos computacionais e os do mundo real, um mecanismo simples para a modularização e a possibilidade de construção de componentes extensíveis.

A adoção da UML (*Unified Modeling Language*) pela OMG (*Object Management Group*) em 1997 como padrão para a descrição de modelos de sistemas orientados a objeto, trouxe como consequência um crescimento do uso da UML como linguagem de modelagem de sistemas orientados a objetos. [2]

O uso de uma linguagem de modelagem tão complexa e extensível quanto a UML requer o suporte de ferramentas. Mesmo que os primeiros esboços de um modelo sejam feitos manualmente, a tarefa de manter, sincronizar e fornecer consistência em um certo número de diagramas torna-se bastante difícil sem o uso de uma ferramenta.

Encontramos atualmente no mercado um grande número de sofisticadas ferramentas CASE que dão suporte à UML. A sofisticação destas ferramentas acarretam em um tempo maior no aprendizado, o que por sua vez implica num aumento do tempo de projeto, gerando um custo maior e consequentemente tornando as organizações menos competitivas. Seria muito proveitoso para as empresas e para o meio acadêmico o surgimento de uma ferramenta simples o suficiente, porém robusta, para proporcionar uma melhor qualidade de ensino das técnicas de análise e projeto orientados a objetos. Este trabalho mostra uma ferramenta CASE que suporta um subconjunto da UML e que tem como público alvo o ambiente universitário, onde se ensinam as técnicas de análise e projeto e construção de sistemas de informação orientando a objetos.

## 2 MOTIVAÇÃO

A garantia de qualidade de software é uma “*atividade abrangente*” aplicada todos os passos do processo de produção de software e que envolve procedimentos para uma efetiva aplicação de métodos e ferramentas. O ensino adequado destes procedimentos requer que as atividades de ensino disponham de ferramentas adequadas.[3]

Encontramos atualmente no mercado um grande número de ferramentas, de custo acessível, para apoio às atividades de programação. Encontramos também ferramentas CASE que dão suporte às etapas de análise e projeto de sistema, que além de serem bastante sofisticadas, não tem um custo que permita seu uso pelos alunos de graduação. A grande motivação para o nosso trabalho vem, exatamente, da ausência de uma ferramenta simples e de baixo custo, que possa ser utilizada para proporcionar uma melhor qualidade de ensino das técnicas de análise e projeto orientados a objetos.

## 3 ARQUITETURA DO *FAST CASE*

O *FAST CASE* foi projetado para proporcionar um uso simples e eficiente. Tivemos a preocupação em proporcionar aos projetistas, mesmo sendo iniciantes, um perfeito entendimento do produto que eles estarão utilizando. O resultado deste esforço é um produto de muito fácil compreensão.

### 3.1 Projeto

O *FAST CASE* trabalha com modelos de sistemas. O conjunto de modelos correspondentes a um sistema é chamado de projeto. Cada projeto é armazenado em um diretório distinto, ou seja, não é possível armazenar informações de dois projetos em um mesmo diretório.

Na identificação do problema, modelamos as principais funcionalidades que o *FAST CASE* iria suportar em relação ao projeto sendo manipulado. Na busca da simplicidade e extensibilidade procuramos definir funções simples, que pudessem ser estendidas ou usadas por outras funções.

### 3.2 Diagramas

O *FAST CASE* suporta a construção e manutenção dos seguintes diagramas da UML: Casos de uso, Classes, Seqüência e Estado.

Os diagramas de casos de uso devem descrever as funcionalidades que o sistema se propõe a resolver do ponto de vista dos atores externos. Um ator é qualquer entidade externa:

um usuário, um dispositivo ou um outro sistema. Esses diagramas mostram a interação entre um usuário e o sistema que ele está utilizando.

O diagrama de classe, parte central dos métodos orientados a objetos, descreve os tipos de objetos do sistema e os relacionamentos estáticos que ocorrem entre eles. O diagrama de classe é estático pois a estrutura descrita por ele é sempre válida em qualquer instante do ciclo de vida de um projeto. Ele mostra os atributos e as operações de uma classe e as restrições associadas a ela. Na modelagem orientada a objetos, classes, objetos e seus relacionamentos são vistos como os elementos primários da modelagem.

Um diagrama de sequência mostra como vários objetos interagem em uma situação específica. Um diagrama de sequência mostra a interação das mensagens trocadas pelos objetos em uma sequência de tempo descrita no caso de uso.

O diagrama de estados mostra a sequência de estados de um objeto ou uma interação durante sua vida em resposta à um estímulo recebido, junto com suas respostas e ações. A UML propõe que um diagrama de estado seja construído para cada classe. Uma vez que todas as instâncias de uma classe tem o mesmo comportamento, elas devem compartilhar o mesmo diagrama de estados. O diagrama de estados da UML utiliza, com algumas pequenas modificações, a notação e a semântica apresentada por David Harel. [1]

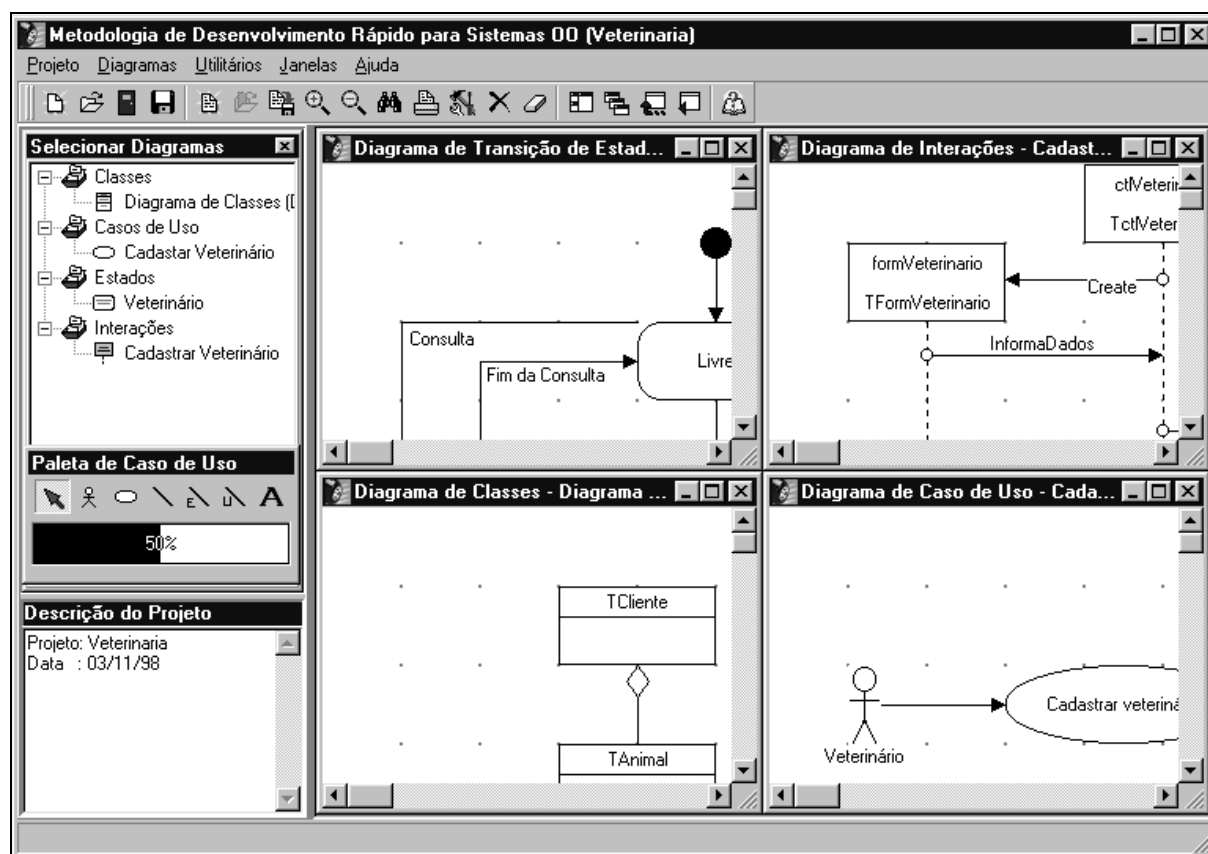


Figura 1 Ambiente de Desenvolvimento do FAST CASE

### 3 Elementos dos diagramas

Além de disponibilizar todas as funções gráficas (*mover, zoom, tamanho, preview, etc*), o **FAST CASE** possibilita a associação de uma definição a cada elementos gráficos que

compõem. Esta definição pode ser exportada formato HTML, permitindo a importação do modelo por todos os sistemas compatíveis com este formato.

### 3.4 Outras funcionalidades

Em adição às funcionalidades básicas acima citadas, o **FAST CASE** oferece ao projetista as seguintes facilidades que simplificam a operação do sistema:

- *Zoom*: permite a ampliação do tamanho dos objetos gráficos desde 50% até 150% do tamanho original.
- *Definição de métodos*: permite a inclusão de novos métodos, tanto no diagrama de classes como no de interação.
- *Recuperação de dados*: no caso de um erro imprevisto durante a operação do sistema, é possível recuperar a situação do sistema através de um arquivo de segurança (*modo texto*) que é criado durante a operação do sistema.
- *Importação de código fonte*: o sistema permite importar o código fonte (*DELPHI*) de um sistema. O projetista pode associar uma classe do modelo à uma classe da implementação, facilitando a engenharia reversa do sistema.
- *Exportação da base de dados de projeto*: permite a criação de um arquivo no formato HTML contendo as definições de todos os objetos dos modelos componentes de um projeto.

## 4 CONCLUSÕES

A maior contribuição desse trabalho é uma ferramenta CASE para ambiente de ensino. Com o **FAST CASE** os alunos podem ter o apoio de uma ferramenta que suporta os conceitos da orientação a objetos nas fase de análise e projeto. Esta ferramenta já está disponibilizada, sem custos, para os nossos cursos de graduação, tendo sido usada por mais de uma centena de alunos em três períodos consecutivos. Como ferramenta extensível, o **FAST CASE** servirá como plataforma de desenvolvimento para vários projetos, que estão sendo iniciados, na área de ferramentas CASE.

## 5 REFERÊNCIAS BIBLIOGRÁFICAS

- [1] BOOCH, G., JACOBSON, I., RUMBAUGH, J. Unified Modeling Language – Notation Guide. Rational Software Corporation, 1997.
- [2] FOWLER, M., SCOTT, K., UML Distilled - Applying the Standard Object Modeling Language, Addison Wesley Object Technology Series, 1997
- [3] PRESSMAN, R. S., Engenharia de Software, Makron Books 1995