

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/228435522>

# EstimaODE: Apoio a Estimativas de Tamanho e Esforço no Ambiente de Desenvolvimento de Software ODE

Article · January 2006

CITATIONS

4

READS

48

4 authors, including:



**Victorio Albani Carvalho**

Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo...

17 PUBLICATIONS 98 CITATIONS

SEE PROFILE



**Lucas de Oliveira Arantes**

Universidade Federal do Espírito Santo

5 PUBLICATIONS 15 CITATIONS

SEE PROFILE



**Ricardo de Almeida Falbo**

Universidade Federal do Espírito Santo

172 PUBLICATIONS 1,661 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



INTEROPERABILIDADE SEMÂNTICA DE INFORMAÇÕES EM SEGURANÇA PÚBLICA [View project](#)



Standards Harmonization [View project](#)



## **EstimaODE: Apoio a Estimativas de Tamanho e Esforço no Ambiente de Desenvolvimento de Software ODE**

**Victorio A. de Carvalho, Lucas O. Arantes, Ricardo de Almeida Falbo**

Departamento de Informática – Universidade Federal do Espírito Santo (UFES)  
Av. Fernando Ferrari s/n, Campus de Goiabeiras – 29.060-900 – Vitória – ES – Brasil

victorioalbani@yahoo.com.br, lucasdeoliveira@gmail.com,  
falbo@inf.ufes.br

***Abstract.** Software estimation is pointed as one of the most difficult tasks to be performed in software development. Although it will never be an exact science, useful techniques for software estimation do exist. Accurate project estimates generally are produced using more than one technique, and by comparing and reconciling their results. Also there is a consensus that estimation requires experience and access to good historical information. Given software estimation complexity, project managers need tools to support them in doing this task. Ideally, these tools should allow applying different approaches for estimating. This paper presents EstimaODE, a tool that offers functionalities for estimating size and effort, which is integrated to the software engineering environment ODE and uses its framework for flexible similarity computing.*

***Resumo.** Estimar é uma das tarefas mais difíceis de se realizar em um processo de software. Apesar de não ser uma ciência exata, há técnicas muito úteis para apoiar a realização de estimativas, sendo que estimativas acuradas geralmente são conseguidas quando se utiliza mais de uma delas e seus resultados são comparados e ajustados. Além disso, dados históricos são igualmente importantes. Neste contexto, gerentes de projeto necessitam de ferramentas para apoiá-los. Este artigo apresenta EstimaODE, uma ferramenta que apóia a realização de estimativas. EstimaODE está integrada ao ambiente de desenvolvimento de software ODE e utiliza sua infra-estrutura flexível para computar similaridade entre itens.*

### **1. Introdução**

O sucesso ou fracasso de um projeto de software está relacionado com sua entrega no prazo acordado com o cliente, dentro do orçamento previsto e com qualidade. A elaboração de estimativas acuradas do trabalho a ser realizado representa um fator chave do sucesso de um projeto e pode ser considerada a fundação para todas as outras atividades de planejamento e gerência de projeto [Valente e Falbo, 2002].

A realização de estimativas envolve, de fato, a estimativa de diversas grandezas, a saber: tamanho, esforço, recursos, tempo e custos do projeto. A execução dessa tarefa, no entanto, não é fácil, sendo a subjetividade envolvida uma de suas principais dificuldades. Para garantir a realização de boas estimativas, é aconselhável que o gerente de projeto possua experiência, utilize diferentes abordagens para realizar estimativas, incluindo dados históricos, e pondere complexidade e riscos [Pressman, 2005].



O tamanho do produto de software tem sido apontado como um ponto de partida para a realização de estimativas [Fiorini et al., 1998]. Ao longo dos anos, pesquisas vêm sendo desenvolvidas para se determinar formas apropriadas de se medir essa grandeza. Existem várias abordagens para se estimar tamanho, sendo que algumas têm merecido bastante atenção da comunidade de Engenharia de Software, tais como a Análise de Pontos de Função [Vazquez et al., 2005] e a Análise de Pontos de Caso de Uso [Schneider e Winters, 2001]. Tomando por base estimativas de tamanho, é possível chegar a estimativas de esforço e, a partir dessas, fica mais fácil estimar recursos, tempo e custo. Assim, de maneira geral, a qualidade das estimativas de recursos, tempo e custo é fortemente influenciada pela qualidade das estimativas de tamanho e esforço.

Tendo em vista a importância das estimativas de tamanho e esforço, é desejável que os gerentes de projeto tenham à disposição ferramentas para apoiá-los, preferencialmente permitindo aplicar diferentes abordagens, de modo que eles possam confrontar valores e chegar a um valor adequado [Arantes et al., 2006].

Este artigo discute o apoio à realização de estimativas no ambiente de desenvolvimento de software ODE [Falbo et al., 2003], procurando destacar a importância da integração neste contexto. ODE provê diversos serviços em sua infraestrutura, incluindo serviços de persistência de objetos e cálculo de similaridade de itens, e possui diversas ferramentas, sendo que algumas delas provêm informações importantes para a realização de estimativas, tal como a ferramenta de definição de processos [Bertollo et al., 2006]. Para apoiar a realização de estimativas em ODE, foi desenvolvida uma ferramenta integrada, denominada EstimaODE. Em sua versão atual, EstimaODE contempla três formas de estimativas, sendo a primeira de esforço e as demais de tamanho: (i) utilizando decomposição do processo de software e dados de projetos similares passados, (ii) Análise de Pontos de Função e (iii) Análise de Pontos de Casos de Uso. A arquitetura de EstimaODE procura dar um tratamento uniforme às estimativas realizadas usando diferentes técnicas, de modo que fique mais fácil comparar resultados e estender a ferramenta para apoiar outras técnicas de estimativa, não só de tamanho e esforço, mas também de grandezas como tempo e custo. Além disso, EstimaODE utiliza a infra-estrutura para caracterização e cálculo de similaridade entre itens de software provida por ODE.

O artigo está estruturado da seguinte forma: a seção 2 discute brevemente o tema principal do trabalho: estimativas. A seção 3 apresenta sucintamente o ambiente ODE, com destaque para a sua infra-estrutura de caracterização e cálculo de similaridade de itens de software. A seção 4 apresenta EstimaODE, tratando seu histórico de desenvolvimento, arquitetura e estágio atual. A seção 5 trata de trabalhos correlatos e, finalmente, na seção 6, são apresentadas as conclusões do trabalho.

## 2. Estimativas

Antes do início de um projeto, o gerente, juntamente com a equipe de desenvolvimento, deve estimar o trabalho a ser realizado (tamanho / esforço), os recursos necessários, o tempo de duração (cronograma) e, por fim, o custo do projeto. Apesar da realização de estimativas ser, muitas vezes, um pouco arte, um pouco ciência, essa importante atividade não deve ser conduzida desordenadamente. Para se tentar alcançar boas estimativas, há algumas opções, dentre elas [Pressman, 2005]: (i) Usar técnicas de



decomposição; (ii) Usar um ou mais métodos de estimativas; (iii) Basear as estimativas em projetos similares que já tenham sido concluídos.

Técnicas de decomposição adotam uma abordagem “dividir para conquistar” na realização de estimativas, decompondo um projeto em suas funções (decomposição do produto) ou atividades (decomposição do processo) principais. Decompondo o escopo do projeto, o gerente de projeto tem unidades menores a estimar, que são mais fáceis de estimar do que o projeto como um todo.

Métodos de estimativa definem uma abordagem sistemática para se estimar alguma grandeza relativa ao projeto de software. Os métodos da Análise de Pontos de Função (APF) [Vazquez et al., 2005] e da Análise de Pontos de Casos de Uso (APCU) [Schneider e Winters, 2001], por exemplo, definem abordagens sistemáticas para se medir e estimar o tamanho de um projeto de software segundo uma medida funcional. Com medidas funcionais, pode-se utilizar modelos empíricos, que usam fórmulas derivadas de estudos experimentais, para prever esforço como uma função do tamanho. Outra opção consiste em utilizar dados de projetos anteriores similares ao projeto em questão para derivar uma medida de esforço a partir de uma medida de tamanho. De fato, dados históricos podem ser usados diretamente como fonte para estimativas, por exemplo, para estimar unidades do projeto derivadas a partir da decomposição do escopo do projeto. Deve-se observar que, no caso do uso de modelos empíricos, fatores culturais da organização não são considerados. Além disso, algumas vezes, os dados usados para produzir esses modelos são derivados de um conjunto limitado de projetos. Por outro lado, quando estimativas são baseadas em dados históricos, os fatores culturais são considerados, pois os projetos foram desenvolvidos na própria organização [Valente e Falbo, 2002].

Quando se faz estimativas, busca-se exatidão, isto é, espera-se que a previsão feita se verifique quando da execução do plano. Entretanto, nem sempre isso é possível, já que a precisão de uma estimativa é fortemente dependente da habilidade de quem a produz, do conhecimento acumulado em experiências anteriores semelhantes e da completeza das informações que se tem à cerca do elemento objeto da estimativa [Fiorini et al., 1998]. Assim sendo, para se conseguir um maior nível de exatidão, o ideal é combinar a aplicação de diversas técnicas. Pela comparação e conciliação das estimativas derivadas pela aplicação de diferentes técnicas, é mais provável que o gerente de projeto consiga derivar uma estimativa precisa [Pressman, 2005]. Vale destacar, neste contexto, a importância de dados históricos, seja para apoiar a realização de estimativas futuras, seja para avaliar a precisão conseguida com cada técnica empregada [Pfleeger, 2004] [Fiorini et al., 1998].

Finalmente, deve-se realçar que as estimativas devem ser feitas repetidamente durante o ciclo de vida do projeto. À medida que o projeto avança e, potencialmente, alguns de seus aspectos se modificam, as estimativas devem ser aprimoradas com base nas informações mais completas sobre as características do projeto [Pfleeger, 2004].

Tendo em vista a complexidade envolvida na realização de estimativas, é muito importante prover apoio automatizado para os gerentes de projeto realizarem esta atividade. Idealmente, uma ferramenta com tal propósito deve permitir a combinação de diversas técnicas de estimativas, prover acesso a dados históricos e ser sensível ao contexto do projeto. Tendo em mente essas considerações, foi desenvolvida



EstimaODE, uma ferramenta para apoiar gerentes no planejamento de projetos no contexto do ambiente de desenvolvimento de software ODE [Falbo et al., 2003]. Dada a grande importância de dados históricos neste contexto, EstimaODE utiliza-se da infraestrutura de apoio à caracterização e cálculo de similaridades de itens de software de ODE. Na seção que se segue, o ambiente ODE e sua infra-estrutura de caracterização e cálculo de similaridade entre itens de software são brevemente apresentados.

### 3. O Ambiente ODE

ODE (*Ontology-based software Development Environment*) [Falbo et al., 2003] é um Ambiente de Desenvolvimento de Software (ADS) centrado em processo, que está sendo desenvolvido no Laboratório de Engenharia de Software da Universidade Federal do Espírito Santo (LabES/UFES). ODE está fundamentado em ontologias, partindo-se do pressuposto que, se as ferramentas de um ADS são construídas baseadas em ontologias, a integração das mesmas pode ser facilitada, pois os conceitos envolvidos são bem definidos pelas ontologias. Dentre as ontologias que compõem a base ontológica de ODE, tem-se as ontologias de processo de software, de qualidade de software, de artefatos de software, de gerência de configuração, de riscos de software e de requisitos de software. Essas ontologias são usadas, dentre outros, para estruturar o ambiente e sua infra-estrutura de gerência de conhecimento e para estabelecer uma forma padrão de comunicação entre os agentes que atuam no ambiente [Falbo et al., 2005].

ODE possui várias ferramentas, dentre elas: de apoio à definição de processos de software, de apoio à gerência de riscos, de apoio à documentação, de apoio à gerência de recursos humanos e de apoio à modelagem usando a UML. Uma primeira versão parcial do ambiente foi implantada em uma organização de desenvolvimento de software no final de outubro de 2004, por meio de uma parceria universidade-empresa, visando a apontar oportunidades de melhoria nas diversas ferramentas do ambiente, tomando por base situações reais dessa organização.

ODE oferece serviços de gerência de conhecimento [Natali e Falbo, 2003] que podem ser utilizados pelas suas diversas ferramentas. Como uma questão fundamental para a gerência de conhecimento é a recuperação de itens de conhecimento relevantes para uma dada situação, percebeu-se que era importante prover uma estrutura flexível e poderosa para caracterizar itens de software e calcular similaridade entre eles. Assim, recentemente foi desenvolvida uma infra-estrutura com essa finalidade, utilizando técnicas de Raciocínio Baseado em Casos [Wangenheim e Wangenheim, 2003], cujo diagrama de classes é apresentado na Figura 1.

Qualquer item de software que precise ser caracterizado deverá implementar a interface `ObjetoCaracterizavel`. Projetos e módulos, por exemplo, precisam ser caracterizados e, portanto, têm de implementar essa interface. Assim, poderão ter associados a eles uma `Caracterizacao`. Uma caracterização de um objeto define valores para características, sendo que essas podem ser de três tipos: `CaracteristicaValorOrdenado`, `CaracteristicaValorNaoOrdenado` e `CaracteristicaConhecimento`. Uma característica de valor ordenado é aquela cujos possíveis valores têm uma ordem intrínseca estabelecida entre eles, tal como complexidade do projeto (alta, média ou baixa). Uma característica de valor não



ordenado é aquela cujos possíveis valores não têm uma ordem pré-estabelecida entre eles, tal como a forma de desenvolvimento (desenvolvimento interno, desenvolvimento totalmente terceirizado, desenvolvimento com equipe mista). Por fim, características de conhecimento são características que têm como possíveis valores objetos de conhecimento. Esses objetos são instâncias das ontologias de ODE. Por exemplo, a ontologia de processo de software [Falbo e Bertollo, 2005] define o conceito Paradigma (implementado como a classe *KParadigma*, herdeira de *Conhecimento*). Assim, a característica de conhecimento que considera paradigmas de desenvolvimento tem como possíveis valores instâncias da classe *KParadigma* (orientado a objetos, estruturado etc). Para as características de valor não ordenado e de conhecimento há, ainda, a opção de se permitir, ou não, a atribuição vários valores.

Para o cálculo de similaridade entre objetos caracterizados (por exemplo, projetos), são usadas tabelas de similaridade (*SimilaridadeValorNaoOrdenado* para as características de valor não ordenado e *SimilaridadeConhecimento* para as características de conhecimento), e os valores máximo e mínimo de uma característica de valor ordenado (em conjunto com o valor de um *PossivelValorOrdenado*).

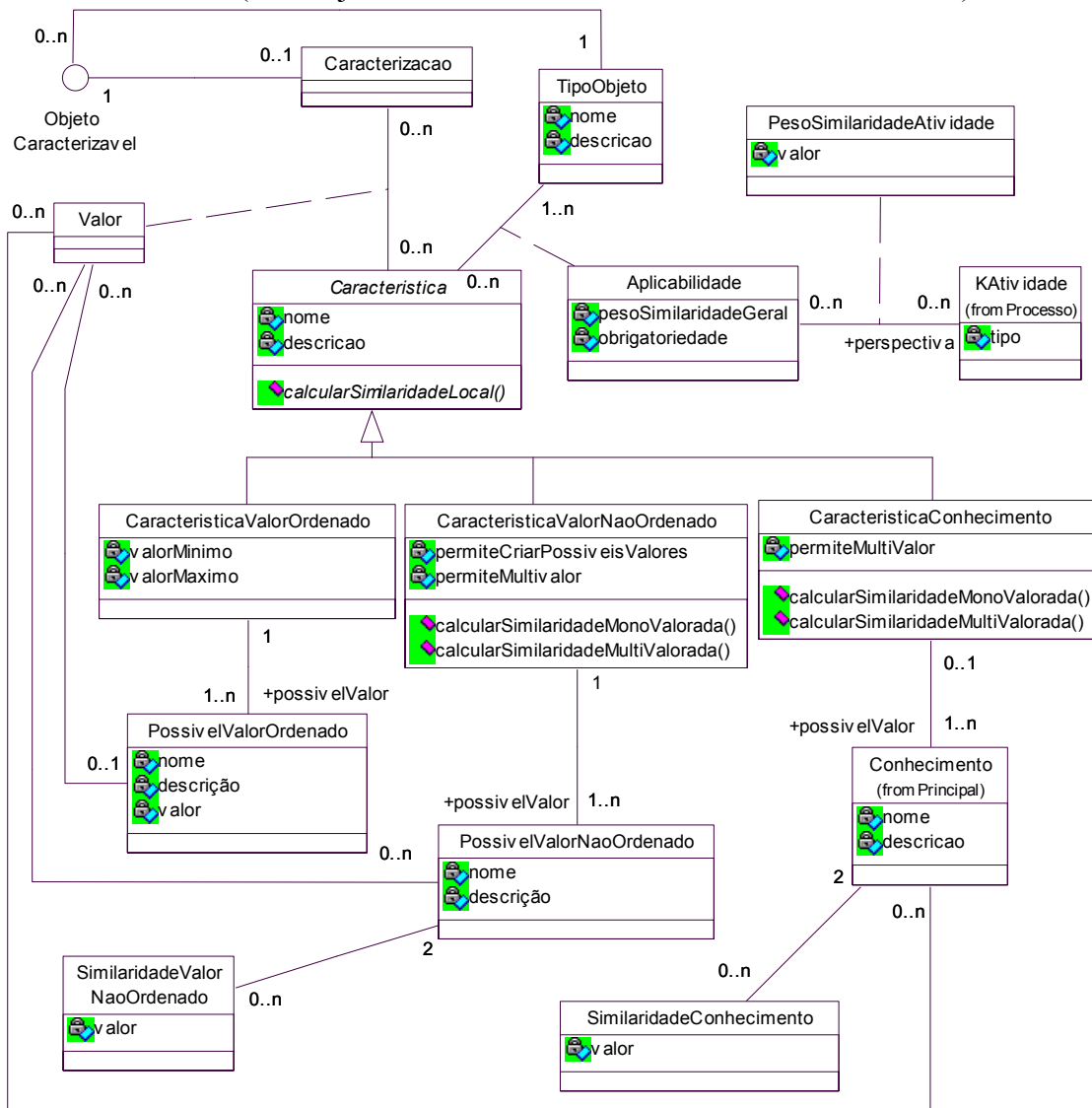


Figura 1 – A infra-estrutura para caracterizar e calcular a similaridade de itens em ODE.



Como diferentes tipos de objetos são classificados por diferentes características, é importante indicar a aplicabilidade de cada característica a um tipo de objeto. Por exemplo, a caracterização de projetos (um tipo de objeto) tem de obrigatoriamente incluir a característica de conhecimento paradigma. Assim, todos os projetos são obrigados a informar o paradigma adotado. Em `Aplicabilidade` é registrado, ainda, o peso a ser atribuído a essa característica quando do cálculo da similaridade geral entre itens de um tipo de objeto.

Finalmente, é interessante observar que, dependendo da atividade do desenvolvimento, a similaridade pode ter de ser computada de maneira diferente. Por exemplo, o nível de experiência da equipe com o processo de software adotado (uma característica) é mais importante para se calcular a similaridade entre projetos sob a ótica da atividade de planejamento do que sob a ótica da atividade de análise de requisitos. Na primeira, estamos interessados em recuperar, dentre outros, estimativas e riscos, e o nível de experiência da equipe com o processo de software adotado tem um grande impacto sobre os itens que se deseja recuperar. Já na análise de requisitos, estamos interessados em recuperar modelos de requisitos e, para tal, o nível de experiência da equipe com o processo de software adotado tem menor impacto. Para tratar esse aspecto, é permitido informar o peso de similaridade por atividade.

Para determinar o grau de similaridade global (GSG) entre dois objetos caracterizados (Objeto1 e Objeto2) é necessário calcular, primeiro, os graus de similaridades locais (GSL) de cada uma das características dos objetos em questão. Esse cálculo varia de acordo com o tipo da característica, a saber:

- (i) Para características de valor ordenado, o GSL é dado por:  $GSL = 1 - |(\text{valorObjeto1} - \text{valorObjeto2}) / (\text{valorMaximo} - \text{valorMinimo})|$ , onde `valorObjetoN` é o valor do `PossivelValorOrdenado` na caracterização do objeto N referente à característica de valor ordenado considerada e `valorMaximo` e `valorMinimo` correspondem aos atributos homônimos da mesma característica
- (ii) Para características de valor não ordenado ou características de conhecimento que não aceitem vários valores, o GSL dos objetos em questão é obtido na correspondente tabela de similaridade (respectivamente, os atributos `valor` das classes `SimilaridadeValorNaoOrdenado` e `SimilaridadeConhecimento`).
- (iii) Para características de valor não ordenado ou características de conhecimento que aceitem vários valores, o grau de similaridade local é dado por:  $GSL = \text{numInterseção} / \text{numUnião}$ , onde `numInterseção` é o número de elementos da interseção dos conjuntos de valores dos objetos em questão e `numUnião` é o número de elementos da união desses mesmos conjuntos.

O grau de similaridade global (GSG) entre dois objetos caracterizados baseia-se no conceito de “vizinho mais próximo ponderado” [Wangenheim e Wangenheim, 2003], sendo dado por:  $GSG = (\sum GSL_i * P_i) / \sum P_i$ , onde `GSLi` é o grau de similaridade local da característica `i` e `Pi` é seu peso no cálculo. O peso de cada característica no cálculo da similaridade global é definido de acordo com a ótica sob a qual a busca está sendo realizada: geral (atributo `pesoSimilaridadeGeral` de `Aplicabilidade`) ou por uma atividade específica (atributo `valor` de `PesoSimilaridadeAtividade`).



#### **4. EstimaODE: A Ferramenta de Apoio à Realização de Estimativas de ODE**

Conforme apontado na seção anterior, o ambiente ODE possui diversas ferramentas, dentre elas EstimaODE, uma ferramenta de apoio a estimativas de tamanho e esforço. Para melhor compreender as funcionalidades e o tratamento de estimativas de EstimaODE, é importante relatar seu histórico de desenvolvimento.

##### **4.1. Histórico de Desenvolvimento de EstimaODE**

Na primeira versão de ODE implantada na empresa parceira, o apoio à realização de estimativas restringia-se a uma ferramenta de apoio à Análise de Pontos de Função (APF), sendo que se encontrava em desenvolvimento uma segunda ferramenta de apoio à Análise de Pontos de Casos de Uso (APCU). Dentre as oportunidades de melhoria (OM) apontadas no primeiro relatório de avaliação entregue no final de dezembro de 2004, duas delas apontavam a área de estimativas, a saber [Arantes et al., 2006]:

- OM1. Uma vez que a organização tinha pouca experiência no uso da técnica de APF, problemas foram detectados no sentido de se obter um indicativo sobre a real precisão das estimativas.
- OM2. Na versão disponibilizada, toda vez que se desejava fazer uma recontagem de pontos de função, os dados da última contagem não eram considerados, sendo essa tarefa considerada uma nova contagem. Assim, ficava difícil gerar um histórico de evolução das contagens de pontos de função, de forma a avaliar os erros de estimativa a partir das características de cada projeto.

Considerando OM1, a solução adotada considerou o uso de dados de projetos passados para fornecer um indicativo de precisão. Uma vez que a organização tinha uma base de dados de projetos com informações acerca do esforço despendido em projetos anteriores, a estratégia consistiu em usar a ferramenta para realizar a contagem de PFs desses projetos. A partir do número de PFs encontrado, comparado a dados reais de esforço, chegou-se a um fator de produtividade (PF / homens-hora) para a organização. Usando esse fator, passou-se a trabalhar com estimativas de esforço (homens-hora), uma grandeza com a qual os gerentes de projeto estavam mais habituados a trabalhar e, portanto, mais fácil de se avaliar a precisão das estimativas obtidas.

Para tratar OM2, foi necessário reestruturar a ferramenta, de modo que fosse possível registrar o histórico da evolução das estimativas. Tomando por base essa reestruturação, foi definida uma estrutura geral para tratar estimativas diversas, integrando diversas formas de estimativas em uma única ferramenta, denominada, então, EstimaODE. Essa nova estrutura foi desenvolvida e foram incorporadas estimativas com base em dados históricos por decomposição do processo e aplicando a técnica de APCU.

Contudo, tais soluções ainda não se mostraram suficientes. Em novo relatório entregue em agosto de 2005, uma nova oportunidade de melhoria foi apontada:

- OM3. O fator de produtividade deveria a ser calculado pela ferramenta, tomando por base projetos similares e não todos os projetos da organização.

Assim, para dar maior precisão no cálculo de similaridade, foi desenvolvida a infra-estrutura apresentada na seção anterior e a arquitetura de EstimaODE reformulada.

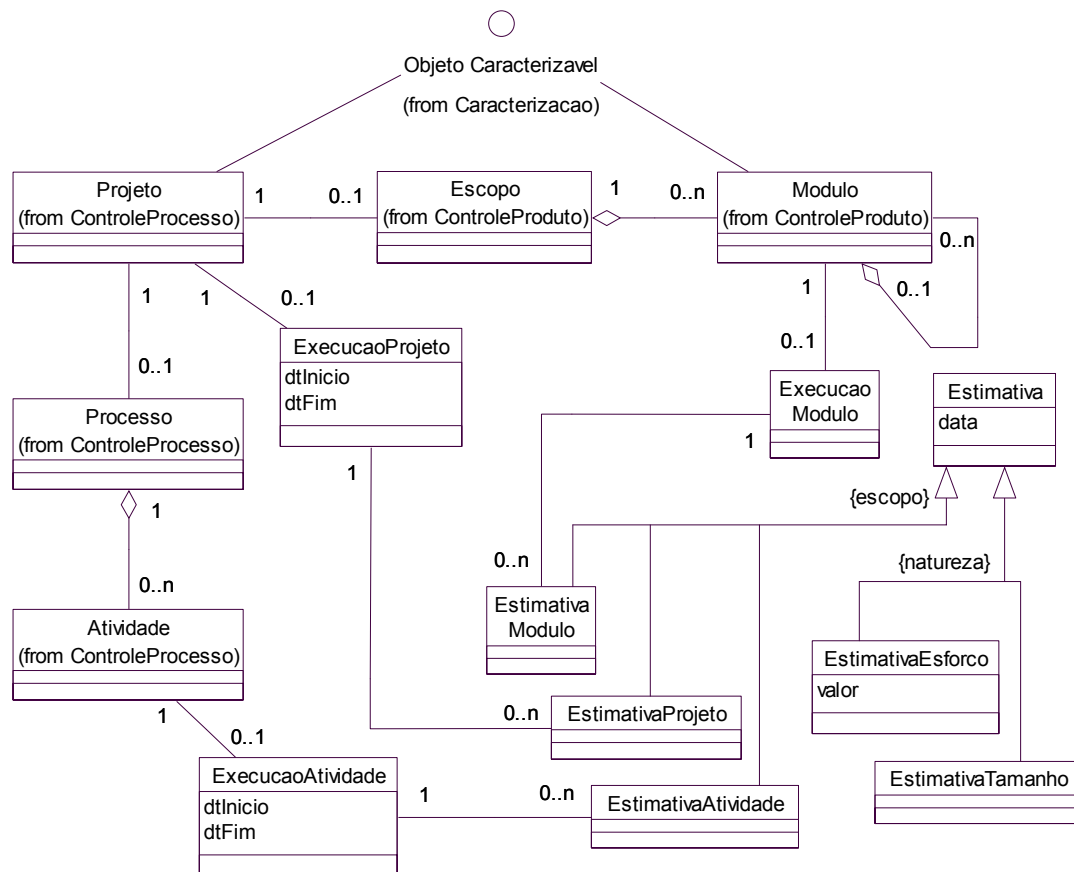




## 4.2. A Arquitetura Básica de EstimaODE

Para que EstimaODE pudesse satisfazer os requisitos colocados, foram considerados os seguintes aspectos em sua arquitetura: (i) a possibilidade de tratar diferentes abordagens de estimativas com uniformidade; (ii) o registro de evolução de cada uma das estimativas existentes; (iii) o cálculo de similaridade entre itens sendo estimados.

Foi desenvolvida, então, uma infra-estrutura em que diferentes tipos de estimativas, organizados pela natureza da estimativa sendo realizada (atualmente, de esforço e tamanho), ou pelo escopo da estimativa (atualmente, para o projeto como um todo, para um módulo do projeto ou para atividades específicas), são abrigados como especializações da classe `Estimativa`, como mostra a Figura 2. As classes `ExecucaoProjeto`, `ExecucaoModulo` e `ExecucaoAtividade` isolam informações sobre a execução e as estimativas das respectivas classes de ODE, `Projeto`, `Modulo` e `Atividade`. Isso é importante, uma vez que essas classes são utilizadas por diversas ferramentas do ambiente e, portanto, idealmente, devem ser mantidas o mais simples possível.



**Figura 2 – A infra-estrutura para tratar estimativas de EstimaODE.**

Uma característica importante dessa infra-estrutura é poder ser facilmente estendida, incorporando-se abordagens de estimativas como especializações de seus tipos em diferentes níveis. Por exemplo, é possível introduzir o tratamento de estimativas com escopo de tempo e custo, como subclasses de `Estimativa` na hierarquia de naturezas. Analogamente, é possível introduzir o tratamento de tipos de estimativas com escopo ainda não considerado, tal como iteração (em que as estimativas são feitas



para cada iteração em um processo de software iterativo), introduzindo subclasses de `Estimativa` na hierarquia de escopo. Finalmente, podem ser tratadas formas específicas de se realizar algum tipo de estimativa, como subclasses do tipo específico. Por exemplo, estimativas segundo a APF ou a APCU são tratadas como especializações de `EstimativaTamanho`. Dessa forma, estimativas, mesmo sendo de diferentes abordagens, são tratadas de maneira uniforme no contexto do ambiente, facilitando a criação, recuperação e comparação das estimativas.

A infra-estrutura proposta permite, ainda, que um mesmo elemento de escopo (atualmente projeto, módulo e atividade) seja estimado segundo diversas abordagens, buscando valores mais precisos, ou estimado várias vezes segundo a mesma abordagem, traçando uma evolução dos valores previstos ao longo do tempo. Manter um histórico dos valores estimados no decorrer dos projetos é de grande importância para que se possa avaliar o nível de acerto desses valores, comparando-os aos dados reais obtidos ao final do projeto.

### 4.3. Estágio Atual de EstimaODE

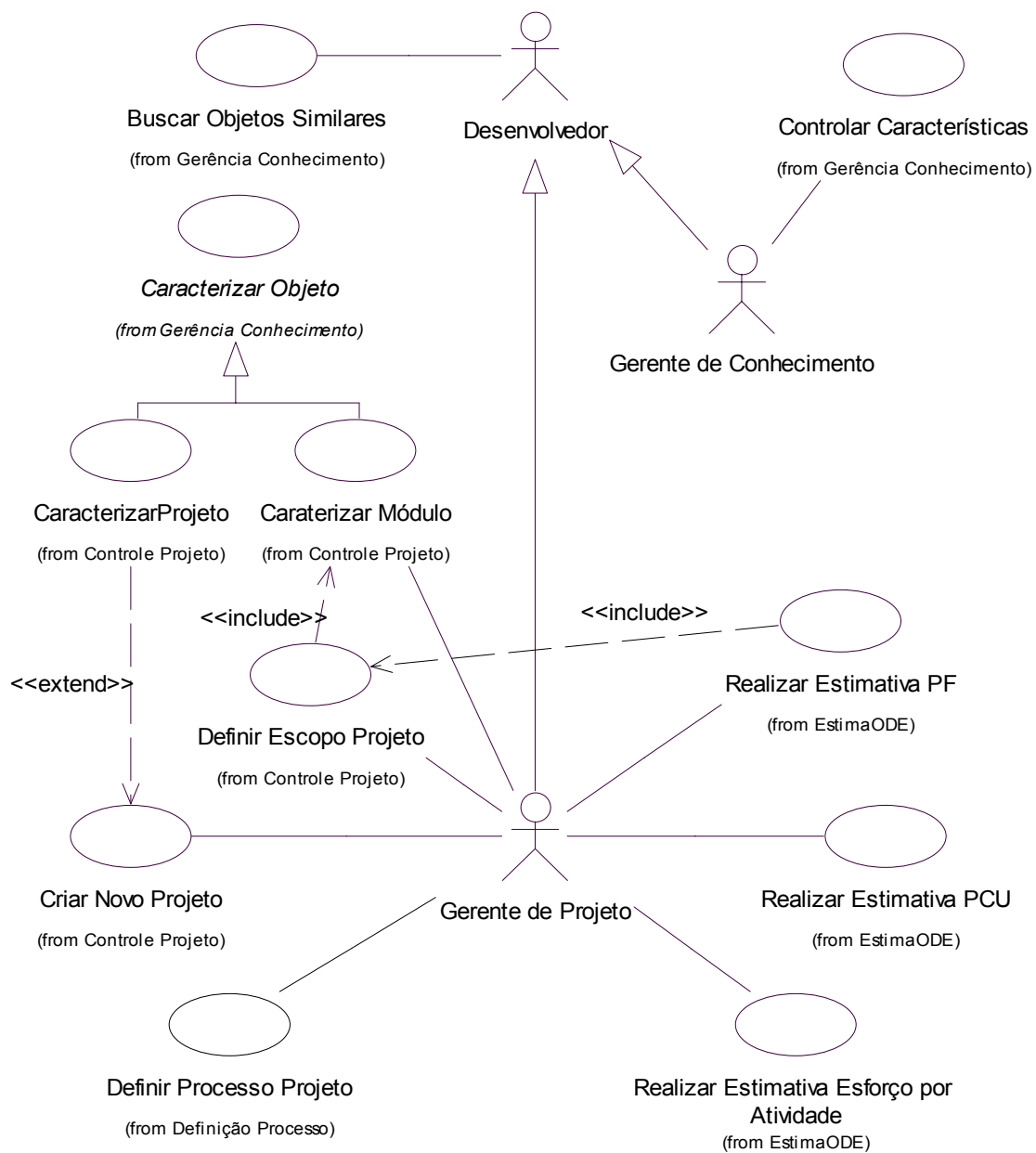
A infra-estrutura de EstimaODE para tratar estimativas foi aplicada a três formas de estimativas: (i) usando Análise de Pontos de Função, para o projeto como um todo ou para módulos selecionados, (ii) aplicando a técnica de Análise de Pontos de Casos de Uso, também para o projeto como um todo ou para módulos, e (iii) com base em dados históricos, para o projeto ou aplicando decomposição do processo do projeto (por atividade). A Figura 3 mostra o modelo de casos de uso correspondente, incluindo funcionalidades do ambiente ODE relacionadas.

Conforme discutido na seção 3, ODE é um ambiente com apoio de gerência de conhecimento. Assim, desenvolvedores realizando suas tarefas cotidianas podem buscar itens de software similares para apoiar seu trabalho. Ainda em relação à gerência de conhecimento, uma vez que a caracterização tem papel decisivo no desempenho da gerência de conhecimento do ambiente, cabe ao gerente de conhecimento da organização controlar o cadastro das características e seus possíveis valores (Figura 1).

O Gerente de Projeto, quando cria um novo projeto, deve caracterizá-lo. Como primeira atividade do planejamento desse projeto, o Gerente de Projeto deve definir seu escopo, quando funcionalidades são atribuídas a módulos. Esses módulos podem ser caracterizados quando criados ou posteriormente. A seguir, um processo deve ser definido para o projeto, utilizando a ferramenta de definição de processos.

Os demais casos de uso são aqueles tratados pela ferramenta EstimaODE. Ao se realizar uma estimativa utilizando a Análise de Pontos de Função (APF), os seguintes passos desse método devem ser executados, a saber [Vazquez et al., 2005]:

1. Determinar o tipo de contagem: Uma contagem de PFs pode ser feita para um projeto de desenvolvimento ou manutenção, ou para uma aplicação instalada. Na versão atual, apenas contagens para projetos de desenvolvimento e manutenção são possíveis em EstimaODE. Essa informação é registrada quando um projeto é criado e, portanto, EstimaODE já a utiliza diretamente. Neste passo, EstimaODE permite, ainda, que se defina a forma de cálculo a ser adotada, dando as seguintes opções: não ajustada segundo o padrão do IFPUG, ajustada segundo o padrão do IFPUG e estimativa segundo a abordagem proposta pela NESMA.



**Figura 3 – Funcionalidades de EstimaODE e outras relacionadas.**

- Identificar o escopo da contagem: Visa a definir a funcionalidade que será incluída em uma contagem de PF específica. Em EstimaODE, duas opções de escopo de contagem são dadas: o projeto inteiro ou um subconjunto de funções do projeto. No segundo caso, devem ser selecionados os módulos que farão parte do escopo dessa contagem. Esses módulos foram definidos previamente durante a definição do escopo do projeto (caso de uso “Definir Escopo Projeto” na Figura 3) e podem ser agora refinados. Neste passo, deve-se definir, ainda, qual o tipo de função de cada módulo (Entrada Externa - EE, Saída Externa - SE e Consulta Externa - CE) e se deve informar os arquivos de dados (Arquivos Lógicos Internos - ALI e Arquivos de Interface Externa - AIE).



3. Contar funções de dados e funções transacionais: quando a forma de cálculo escolhida no passo 1 for o padrão IFPUG, para cada função de transacional (EE, SE, CE) deve-se informar os números de arquivos e de itens de dados referenciados, a partir dos quais a ferramenta calcula a complexidade da função, como ilustra a Figura 4. Procedimento semelhante é feito para as funções de dados (AIE, ALI), informando-se os números de registros lógicos e de itens de dados.
4. Determinar o fator de ajuste: quando a forma de cálculo escolhida for a contagem ajustada do padrão IFPUG, é necessário determinar o fator de ajuste: Neste passo são consideradas 14 características gerais de sistemas (p.ex., processamento distribuído, volume de dados etc). O fator de ajuste é determinado por meio de um cálculo que leva em conta o nível de influência de cada uma dessas características no projeto em questão. Ainda que a ferramenta permita a determinação do fator de ajuste, desaconselha-se o seu uso, pois o uso de dados organizacionais para derivar estimativas de esforço a partir dos pontos de função não ajustados já realiza esse ajuste.
5. Calcular o número de PFs: De acordo com o tipo de contagem e a forma de cálculo escolhidos no passo 1, EstimaODE calcula o número de PFs. Por exemplo, para projetos de desenvolvimento, usando a contagem estimativa da NESMA, a seguinte fórmula é utilizada:  $nPF = 7 * nALI + 5 * nAIE + 4 * nEE + 5 * nSE + 4 * nCE$ .

Finalmente, quando solicitado pelo gerente de projeto, uma estimativa de esforço pode ser gerada a partir do número de PFs calculado, usando um fator de produtividade (número de homens-hora por PF) definido pela ferramenta a partir de projetos similares.

	De 1 a 4	De 5 a 15	16 ou mais
Até 1	Simple	Simple	Média
Apenas 2	Simple	Média	Complexa
3 ou mais	Média	Complexa	Complexa

Descrição da Função	Arquivos Referenciados	Itens de Dados	Complexidade
Autorizar Criação de SubGrupo	2	3	Simple
Entrar em Grupo	4	8	Complexa
Solicitar Criação de SubGrupo	2	8	Média
Criar SubGrupo	2	8	Média
Cadastrar Usuário	2	24	Complexa

Figura 4 – Análise de Pontos de Função em EstimaODE.



É importante destacar o quanto a integração de EstimaODE a ODE facilita uma análise de pontos de função. Diversas informações e funcionalidades disponíveis no ambiente são utilizadas, evitando retrabalho e inconsistências. Conforme anteriormente mencionado, ao se criar um novo projeto, informa-se se o mesmo é um projeto de desenvolvimento ou manutenção. Essa informação é diretamente utilizada pela ferramenta para definir o tipo de contagem. As informações sobre os módulos que compõem o escopo de um projeto, bem como a própria funcionalidade de decomposição do produto, são diretamente utilizadas por ocasião da definição do escopo de contagem. Por fim, a infra-estrutura de caracterização e cálculo de similaridade entre itens é utilizada no cálculo do fator de produtividade de projetos similares.

Outro método de estimativa contemplado por EstimaODE é a Análise de Ponto de Caso de Uso (APCU) [Schneider e Winters, 2001]. A APCU define um processo para contagem de Pontos de Casos de Uso (PCUs) que compreende os seguintes passos: (1) Determinar o peso dos atores; (2) Determinar o peso dos casos de uso; (3) Determinar a contagem de PCUs não ajustados; (4) Determinar os fatores de ajuste relacionados à complexidade técnica do projeto e a experiência das pessoas envolvidas; (5) Calcular os pontos de caso de uso, considerando os fatores de ajuste.

EstimaODE provê funcionalidades para apoiar todos esses passos, sendo que é importante destacar, novamente, que o fato de EstimaODE estar integrada a ODE facilita o trabalho. ODE possui uma ferramenta de apoio à modelagem de sistemas usando a UML e, portanto, casos de uso e atores não precisam ser informados. Eles são recuperados do repositório central do ambiente por EstimaODE. Além disso, de maneira análoga à funcionalidade de apoio à APF, uma estimativa de esforço pode ser derivada a partir do número de PCUs calculado, usando um fator de produtividade (número de homens-hora por PCU), definido pela ferramenta a partir de projetos similares.

A terceira e última forma de estimativa atualmente contemplada por EstimaODE é a estimativa de esforço com base em dados históricos de projetos similares já concluídos. Essa forma de estimativa leva em conta que, para se realizar uma estimativa de esforço confiável, é interessante considerar valores obtidos dos esforços reais despendidos em projetos anteriores similares ao projeto em questão. Em EstimaODE, essa análise pode ser feita considerando o projeto como um todo ou cada uma de suas atividades (decomposição do processo).

Para a estimativa do projeto como um todo, EstimaODE exibe uma lista de projetos similares ao projeto corrente com seus respectivos valores reais de esforço. Quando o escopo é uma atividade, leva-se em conta o processo do projeto em questão. Suas atividades são listadas e, selecionada uma atividade, são exibidos dados dessa atividade e uma lista com valores de esforço despendido em atividades do mesmo tipo em projetos similares. Novamente, é importante destacar que EstimaODE faz uso, neste caso, do processo definido pela ferramenta de definição de processos de ODE, bem como de dados de apropriação de esforço, mostrando, mais uma vez, a importância da integração de ferramentas em um ADS.

## 5. Trabalhos Correlatos

Apesar da grande importância de estimativas, pouca atenção se tem dado a ferramentas para apoiar essa tarefa de maneira integrada. Em recente pesquisa feita junto a 16



empresas situadas em Vitória – ES, todas utilizavam planilhas eletrônicas para apoiar essa atividade. Além disso, a maioria das ferramentas de apoio à realização de estimativas disponíveis pesquisadas apóia uma forma específica de estimativa e não está integrada a um ambiente de desenvolvimento de software, tornando a entrada de dados uma tarefa repetitiva e propensa a inconsistências.

As ferramentas *Function Point WorkBench* [Charismatek Software Metrics, 2006] e *FP Recorder* [Chis Pty, 2006], por exemplo, apóiam a análise de pontos de função, permitindo decompor o sistema em módulos (decomposição do produto). Apesar de se limitar a esse escopo e tipo de estimativa, essas ferramentas apresentam alguns aspectos bastante interessantes, não tratados por EstimaODE. Um desses aspectos é permitir agrupar itens de estimativa, a partir de estimativas dos subitens, no caso módulos. No caso de *Function Point WorkBench*, também é fornecida ao usuário uma funcionalidade para cadastrar arquivos e itens de dados de cada arquivo, de modo que, durante a estimativa, basta selecionar o arquivo que a ferramenta conta quantos itens de dados estão associados, automatizando as estimativas. Assim como EstimaODE, *Function Point WorkBench* permite exportar estimativas em um formato XML e gerar relatórios. Além disso, ao fazer uma estimativa, permite que essa tarefa seja baseada em modelos já cadastrados.

No que se refere à integração com outras ferramentas, a ferramenta *Estimate Easy UC* [Duversa Software, 2006], que apóia a análise de pontos de casos de uso, permite que atores e casos de uso sejam importados de modelos XMI, formato XML padrão para intercâmbio de modelos UML e modelos no formato da ferramenta Rational Rose, dentre outros. Essa facilidade também é fornecida por EstimaODE, de forma indireta, uma vez que a ferramenta de apoio à modelagem UML de ODE, denominada OODE, permite a importação de modelos XMI e, conforme discutido na seção 3.3, EstimaODE tem acesso aos dados de modelos de objetos armazenados em ODE.

Um trabalho que merece destaque neste contexto, é a ferramenta *CustPlan* [Barcellos et al., 2003] da Estação TABA, um ambiente de desenvolvimento de software com apoio de gerência de conhecimento. *CustPlan* apóia o planejamento de custos e de cronogramas baseando-se no uso de técnicas de Engenharia de Software e no aprendizado da organização por meio de suas experiências passadas, através da gerência do conhecimento. As técnicas empregadas incluem a Análise de Pontos de Função e o COCOMO II. Entretanto, não há uma referência ao uso de abordagens diferentes para o mesmo tipo de estimativa, preocupação grande de EstimaODE. Como ambos os ambientes TABA e ODE têm infra-estruturas de gerência de conhecimento, ambos oferecem facilidades para buscar estimativas e experiências de projetos similares.

Finalmente, na linha de estimativas apoiadas por gerência de conhecimento, há também ProKnowHow [Valente e Falbo, 2002], uma ferramenta proprietária de uma organização, que utiliza uma abordagem GQ(I)M para oferecer mecanismos de busca para itens de conhecimento relacionados a estimativas. Assim, a principal diferença entre os apoios de gerência de conhecimento oferecidos por ProKnowHow e EstimaODE está na abordagem da primeira, que parte de objetivos, passando a indicadores e métricas, e chegando aos dados de métricas, que correspondem aos valores estimados.



## 6. Conclusões

Efetuar estimativas não é uma tarefa fácil e requer apoio. Idealmente, esse apoio deve ser provido de forma automatizada utilizando dados históricos de projetos similares e, sobretudo, permitindo o uso de diferentes abordagens.

Este artigo apresentou a ferramenta EstimaODE, procurando discutir como ela se beneficia da estrutura do ambiente de desenvolvimento de software ODE, evitando que muito trabalho tenha de ser refeito para se usar a ferramenta de estimativas e fazendo com que dados históricos de projetos similares sejam usados mais facilmente.

Algumas potenciais melhorias já detectadas incluem a necessidade de ampliar o leque de tipos de estimativas segundo as duas perspectivas consideradas em EstimaODE: escopo e natureza. Quanto ao escopo, está em estudo uma funcionalidade para permitir a realização de estimativas com base nas iterações de um projeto que adote um modelo de ciclo de vida iterativo. No que se refere à natureza, está-se considerando estimativas de tempo e custo, além de outras formas de se estimar tamanho e esforço.

Vale enfatizar que, uma vez que EstimaODE está integrada ao ambiente ODE, foi possível aproveitar algumas facilidades provenientes dessa integração, a saber:

- ODE é um ambiente centrado em processo. Assim, no momento em que são realizadas as estimativas, já há um processo definido, decomposto em atividades. Dessa forma, a amarração do projeto a suas estimativas e também das atividades que compõem o seu processo às suas respectivas estimativas é feita de forma direta e natural.
- ODE possui um repositório único de dados. Desse modo, todos os projetos anteriores, com suas atividades, estimativas e esforços reais estão em um único lugar. Isso permite a fácil recuperação de qualquer um desses dados, como projetos similares ou esforços registrados, para informar melhor o gerente que realiza a estimativa.

## Agradecimentos

Este trabalho foi realizado com o apoio do CNPq e da CAPES, entidades do Governo Brasileiro dedicadas ao desenvolvimento científico e tecnológico, da FAPES, Fundação de Apoio à Ciência e Tecnologia do Espírito Santo, e das empresas VixTeam e Projeta, parceiras que têm financiado o projeto e dado feedback de sua aplicação a casos reais.

## Referências

- Arantes, L.O., Carvalho, V.A., Falbo, R.A. (2006) “Uma Ferramenta Integrada de Apoio a Estimativas de Tamanho e Esforço em um Ambiente de Desenvolvimento de Software”, IX Workshop Iberoamericano de Ingeniería de Requisitos y Desarrollo de Ambientes de Software – IDEAS'2006, La Plata, Argentina (short paper).
- Bertollo, G., Segrini, B., Falbo, R.A. (2006) “Definição de Processos de Software em um Ambiente de Desenvolvimento de Software Baseado em Ontologias”, V Simpósio Brasileiro de Qualidade de Software, SBQS'2006, Vila Velha, Brasil.
- Barcellos, M.P., Figueiredo, S.M., Rocha, A.R.C., Travassos, G.H. (2003) “Utilização de Métodos Paramétricos, Analogias, Julgamento de Especialistas e Conhecimento



- Organizacional no Planejamento de Tempo e Custos de Projetos de Software”, II Simpósio Brasileiro de Qualidade de Software, 17 – 31, Fortaleza, Brasil.
- Charismatek Software Metrics (2006), Function Point WORKBENCH  
[http://www.charismatek.com.au/\\_public1/index.htm](http://www.charismatek.com.au/_public1/index.htm)
- Chis Pty Ltd (2006), FP Recorder, <http://www.fprecorder.com>
- Duessa Software (2006), Estimate Easy UC, <http://www.duessa.com>
- Falbo, R. A., Natali, A. C. C., Mian, P.G., Bertollo, G., Ruy, F.B. (2003) “ODE: Ontology-based software Development Environment”, In: Memórias de IX Congresso Argentino de Ciencias de la Computación, p. 1124-1135, La Plata, Argentina.
- Falbo, R.A., Bertollo, G. (2005) “Establishing a Common Vocabulary for Software Organizations Understand Software Processes”, International Workshop on Vocabularies, Ontologies and Rules for the Enterprise, Enschede, The Netherlands.
- Falbo, R.A., Ruy, F.B., Dal Moro, R. (2005) “Using Ontologies to Add Semantics to a Software Engineering Environment” Proc. of the 17th International Conference on Software Engineering and Knowledge Engineering, Taipei, China.
- Fiorini, S. T., Staa, A., Baptista, R. M. (1998) *Engenharia de Software com CMM*, Brasport.
- Natali, A.C.C. e Falbo, R.A. (2003) “Gerência de Conhecimento em ODE”, Anais do XVII Simpósio Brasileiro de Engenharia de Software, p. 270-285, Manaus, Brasil.
- Pfleeger, S.L. (2004) *Engenharia de Software: Teoria e Prática*, Prentice Hall, 2a edição.
- Pressman, R. S. (2005) *Software Engineering: A Practitioner’s Approach*, Mc Graw Hill, 6<sup>th</sup> edition.
- Schneider, G. and Winters, J. P., (2001) *Applying Use Cases: A Practical Guide*, 2nd edition, Addison-Wesley.
- Valente, F. F. R. e Falbo, R. A. (2002) “Uso de Gerência de Conhecimento para Apoiar a Realização de Estimativas”, In: Proceedings of the XXVIII Latin-American Conference on Informatics - CLEI'2002, Montevideo, Uruguay, November 2002.
- Vazquez, C.E., Simões, G.S., Albert, R.M. (2005) *Análise de Pontos de Função: Medição, Estimativas e Gerenciamento de Projetos de Software*, Editora Érica, 3<sup>a</sup> edição.
- Wangenheim, C.G. e Wangenheim A. (2003) *Raciocínio Baseado em Casos*, Ed. Manole.