

Qualidade de Componentes de Software



Universidade Federal de Juiz de Fora
Núcleo de Pesquisa em Qualidade de Software

Fernanda Campos

Regina Braga

fcampos@dcc.ufjf.br

EQPS Indaiatuba, SP - 2003



Introdução

- O NPQS em parceria com a COPPE/Sistema vem desenvolvendo pesquisas na área de Desenvolvimento Baseado em Componentes.
- Uma das preocupações do grupo se refere à qualidade que estes componentes devem ter tanto no processo de desenvolvimento, quanto no produto final e na usabilidade destes componentes.



Objetivos do projeto

- Especificar um conjunto de características de qualidade para o processo, produto e o uso de componentes de software.
- Desenvolver um conjunto de componentes de software para os domínios agropecuário e da educação a distância onde os requisitos de qualidade definidos no projeto sejam seguidos e implementados e sirvam de exemplo e modelo para outros desenvolvedores.



Componentes de software

- Componentes de software reutilizáveis são artefatos autocontidos, claramente identificáveis, que descrevem ou realizam uma função específica e têm interfaces claras em conformidade com um dado modelo de arquitetura de software, documentação apropriada e um grau de reutilização definido.
(Braga, 1999)



Componentes de software

- A construção de software através de componentes, leva a uma maior reutilização e conseqüentemente um avanço na qualidade, produtividade e custos.
- Visa a solução de um problema através da composição de partes reutilizáveis que possam ser conectadas umas às outras.
- A abordagem desenvolvimento baseado em componentes - DBC é uma proposta promissora para uma significativa melhoria da produção de software nos próximos anos.



Desenvolvimento Baseado em Componentes

- A abordagem de desenvolvimento através da integração de componentes enfatiza a busca, seleção e adaptação de componentes em todas as fases de desenvolvimento para a construção de uma aplicação específica.
- Essa visão tem se mostrado muito importante por focar os serviços prestados pelos componentes separadamente de seus detalhes de implementação, maximizando a reutilização e evitando o uso de informações não significativas durante a montagem da arquitetura.



Vantagens do DBC

- **Maior confiabilidade**, já que os componentes reutilizados que são empregados nos sistemas em operação devem ser mais confiáveis do que os componentes novos.
- **Redução dos riscos de processo**, pois se recorrermos a um componente já existente, serão menores as incertezas sobre os custos relacionados ao reuso desse componente do que sobre os custos de desenvolvimento.
- **Uso efetivo de especialistas** no desenvolvimento dos componentes.
- **Conformidade de padrões**, já que alguns padrões, como os padrões de interface com o usuário, podem ser implementados como um conjunto de componentes-padrão.
- **Desenvolvimento acelerado**, pois de modo geral, é mais importante fornecer um sistema para o mercado o mais rápido possível do que se prender aos custos gerais de desenvolvimento.



Problemas do DBC

- **Aumento nos custos de manutenção**, pois se o código-fonte do componente não estiver disponível, então os custos de manutenção poderão aumentar, uma vez que os elementos reutilizados no sistema podem se tornar crescentemente incompatíveis com as mudanças do sistema.
- **Falta de ferramentas de apoio**, já que os conjuntos de ferramentas CASE não apóiam o desenvolvimento com reuso.
- **A síndrome do “não-foi-inventado-aqui”**, que diz respeito a alguns engenheiros de software preferirem reescrever os componentes ao invés de reutilizá-los.
- **Manutenção de uma biblioteca de componentes**, pois implementar uma biblioteca de componentes e assegurar que os desenvolvedores de software utilizem essa biblioteca pode ser dispendioso.
- **Encontrar e adaptar componentes reutilizáveis**, pois os componentes de software devem ser encontrados em uma biblioteca, compreendidos e, algumas vezes, adaptados, a fim de trabalharem em um novo ambiente.



O usuário quer saber

- Que evidência existe, que este componente é adequado para trabalhar em minha aplicação?
- O componente foi testado de uma forma relevante para minha utilização pretendida?
- Quanta utilização relevante este componente já teve em áreas similares a qual minha aplicação pertence?
- Quais são as implicações de usar este componente?



O desenvolvedor quer saber

- Qual evidência existe que este componente é adequado para trabalhar em aplicações reais do usuário?
- O componente foi testado numa variedade de situações suficiente?
- O componente é projetado para ter um desempenho eficiente numa variedade razoável de contextos?



Qualidade de componentes

- Vários fatores influenciam na qualidade de uma aplicação baseada em componentes:
- Deve ser possível encontrar componentes reutilizáveis apropriados.
- O responsável pelo reuso dos componentes precisa ter certeza de que os componentes se comportarão como especificado e de que serão confiáveis.
- Os componentes devem ter a documentação associada para ajudar o usuário a compreendê-los e adaptá-los a uma nova aplicação.



Qualidade de componentes

- Um componente pode ser visto como se fosse um software em miniatura, possuindo com isso uma arquitetura, sendo constituído por vários artefatos, podendo ser construído a partir de outros componentes e possuindo características de um software completo.
- As características dos componentes influenciam na qualidade do produto de software final e é por isso que a qualidade do mesmo é fundamental.
- O ideal é que ele fosse certificado, isto é o mesmo fosse avaliado se satisfaz os requisitos do sistema com alta qualidade e confiabilidade.



Referências sobre qualidade de componentes

- Características de Qualidade utilizando o Modelo *Fuzzy*, Arnaldo Dias Belchior e Régis Patrick Silva Simão,
- Características de Qualidade relacionadas ao código-fonte, Xia Cai, da Universidade Chinesa de Hong Kong.
- Qualidade para componentes COTS (*Commercial off-the-shelf*) Manuel F. Bertoa e Antonio Vallecillo, da Universidade de Málaga, na Espanha.
 - Todos baseados na norma ISO/IEC 9126.



Abordagem adotada

- Qualidade de Processo:
 - características de qualidade presentes no processo de desenvolvimento do componente.
- Qualidade de Produto:
 - características de qualidade presentes no componente propriamente dito.
- Satisfação do Usuário:
 - características de qualidade retiradas de uma análise do comportamento do componente durante sua utilização.



Características de qualidade de produto - exemplos

- **Acoplamento entre Objetos**
 - Qual o número de outras classes às quais uma classe é acoplada?
- **Número de Classes**
 - Qual o número de classes ou interfaces que são declaradas ao todo em um componente?
- **Precisão**
 - Qual a porcentagem de resultados obtidos com a precisão especificada pelos requisitos dos usuários?
- **Acurácia Computacional**
 - Qual o número de resultados precisos retornados pelas operações do componente de acordo com as especificações do usuário?



Características de qualidade de produto - exemplos

- **Controle de Erros**
 - O componente pode lidar com situações de erro?
- **Tratamento de Exceções**
 - Existe algum mecanismo no componente para tratamento de exceções?
- **Tempo de Resposta**
 - Qual o tempo gasto desde que uma requisição é recebida por um componente até uma resposta ser enviada?
- **Capacidade**
 - Qual o montante de informação de entrada que pode ser processada com sucesso pelo componente em um dado período de tempo?



Características de qualidade de produto - exemplos

- **Tamanho do componente**
 - Qual o tamanho do componente, em termos de espaço que ele ocupa na memória?
- **Compatibilidade de dados**
 - O formato dos dados manipulados pelo componente está em conformidade com algum padrão ou convenção?
- **Interfaces fornecidas**
 - Quantas são as interfaces fornecidas pelo componente?
- **Interfaces requeridas**
 - Quantas interfaces o componente requisita de outros componentes para operar?



Características de qualidade de produto - exemplos

■ Tolerância à falhas

- O componente consegue manter seu nível de desempenho em caso de falha?

■ Frequência de reuso

- O componente é reutilizado muitas vezes?

■ Diversidade de reuso

- Qual a diversidade das situações nas quais o componente é reutilizado?



Características de qualidade satisfação do usuário - exemplos

- **Apreensibilidade**
 - Quão fácil é aprender a usar e administrar o componente?
- **Documentação do usuário**
 - Qual a qualidade da documentação do usuário, em termos de completude, clareza e utilidade?
- **Operacionalidade**
 - Quão fácil é operar, configurar e administrar um componente?
- **Modificabilidade**
 - Quão fácil é customizar, modificar e corrigir o componente?
- **Testabilidade**
 - Quão fácil é testar o componente?



Etapas do desenvolvimento dos componentes

- construção de componentes de negócio;
- disponibilização dos componentes em um repositório de componentes no ambiente *Odyssey*;
- especificação de uma aplicação para integrar estes componentes e mostrar o funcionamento dos mesmos em conjunto.



Exemplo da modelagem e documentação dos componentes

■ Domínio Agropecuário

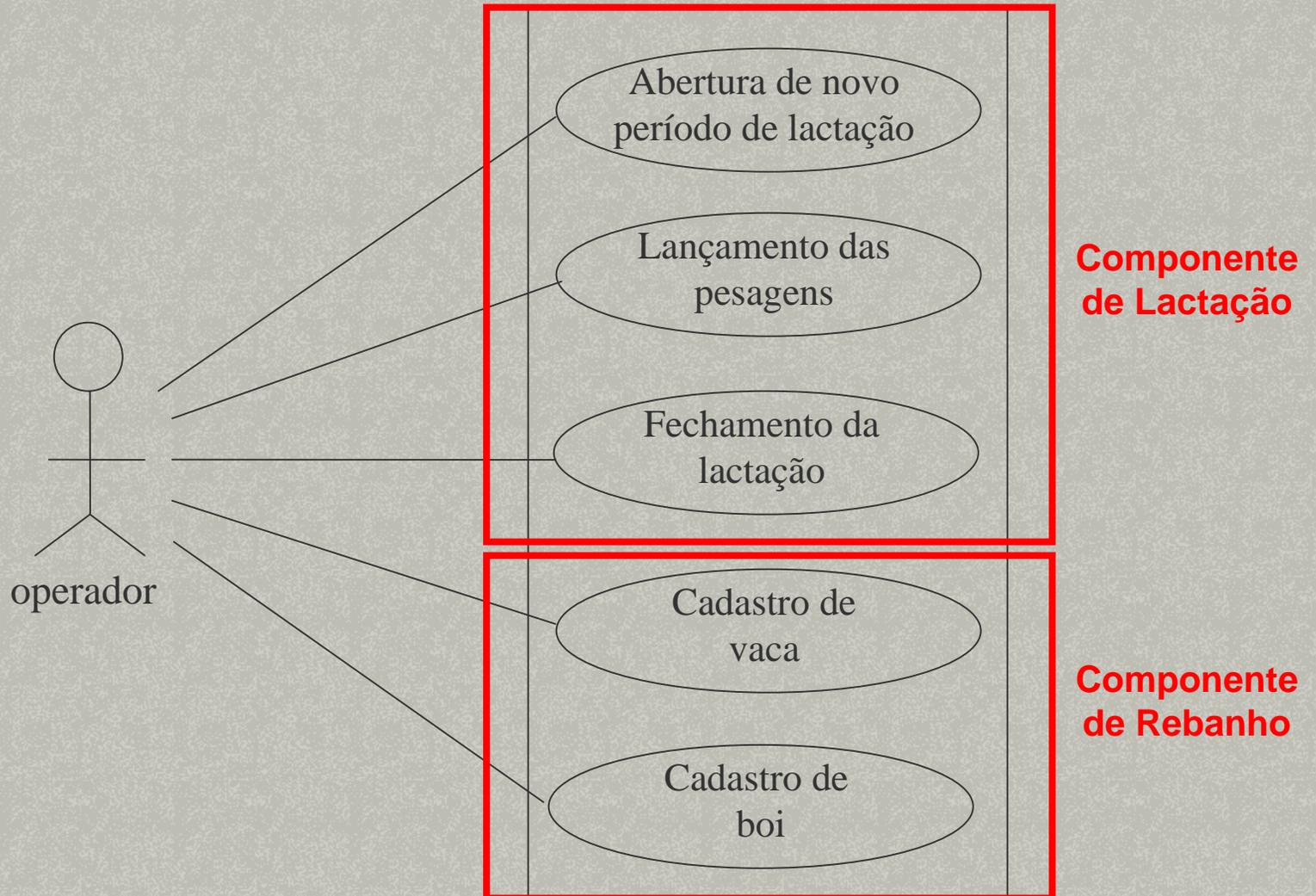
- Controle Leiteiro

■ Análise do Domínio

- Documentação
- Análise de softwares
 - GV Farm
 - Pró-Leite
- Consultoria com especialistas

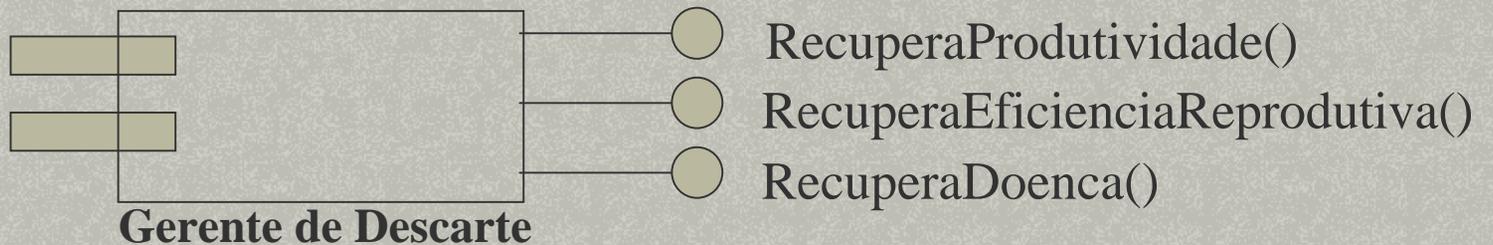
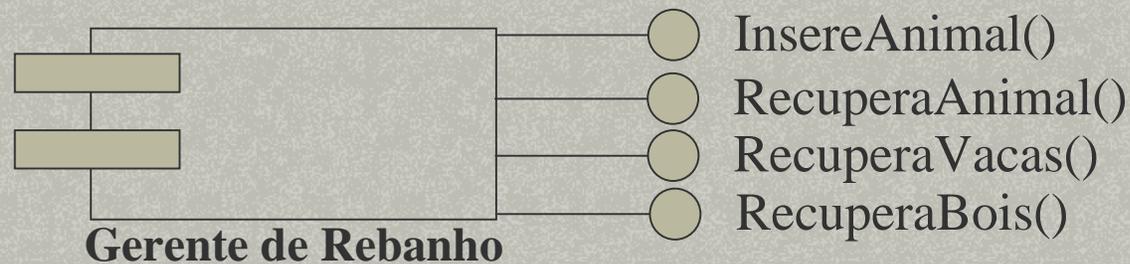
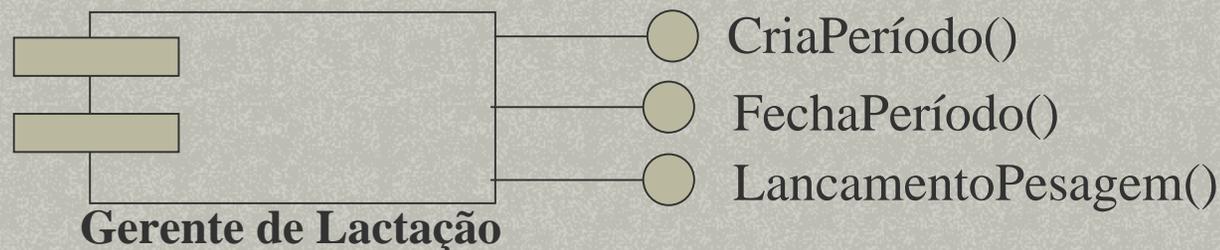


Diagrama de Casos de Uso





Componentes do domínio





Componente de suporte

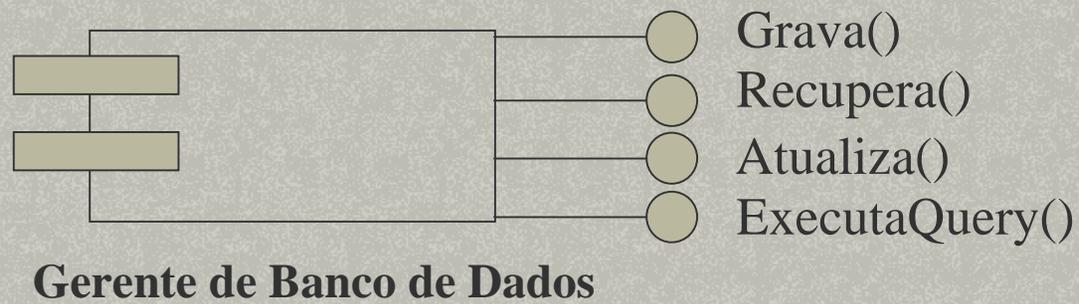
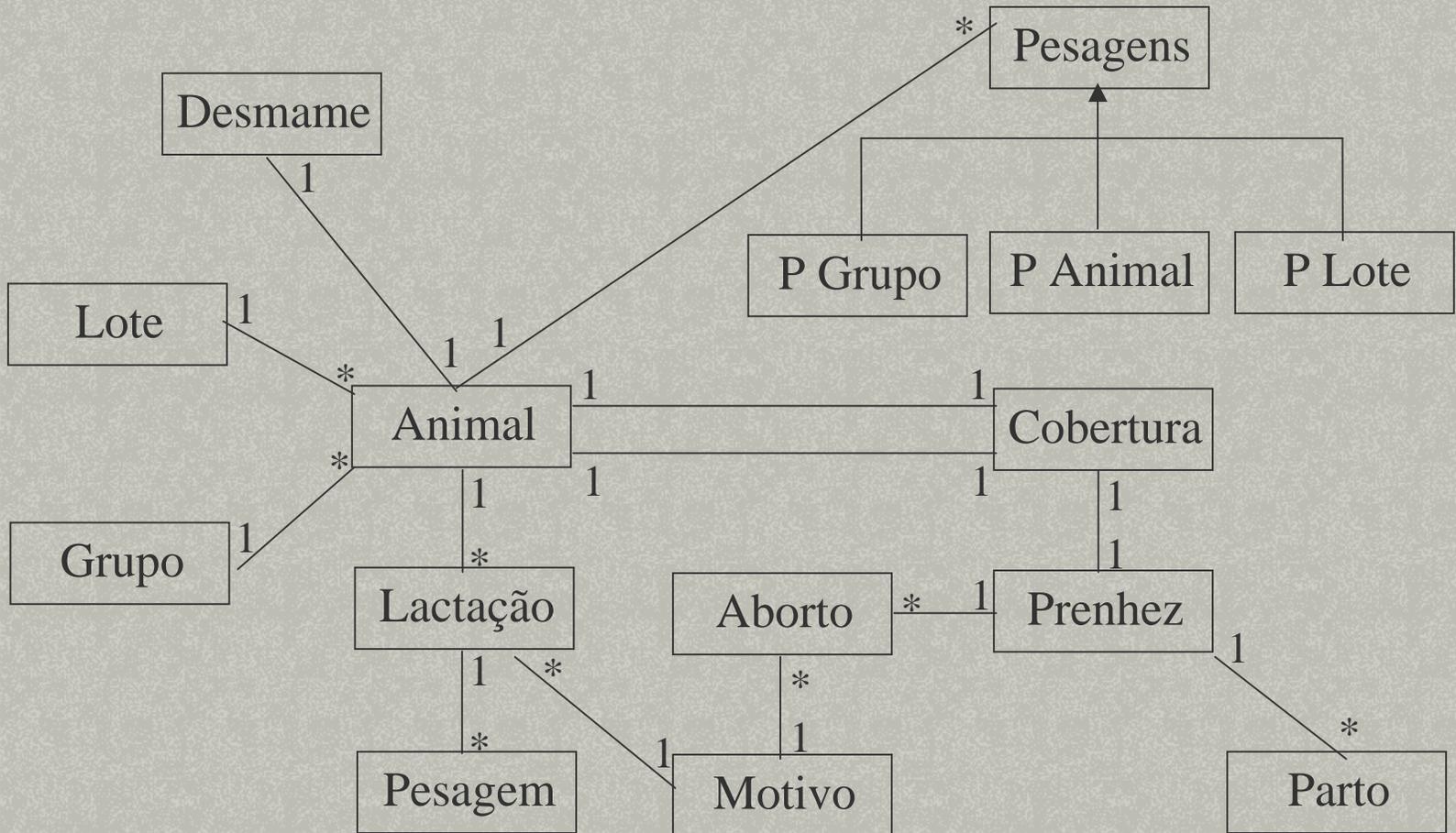




Diagrama de Classes





Classes

| <<Type>> Animal |
|--------------------|
| Brinco: Integer |
| Nome: String |
| Dt_Nasc: DateTime |
| Raca: String |
| Sexo: String |
| InsererDados() |

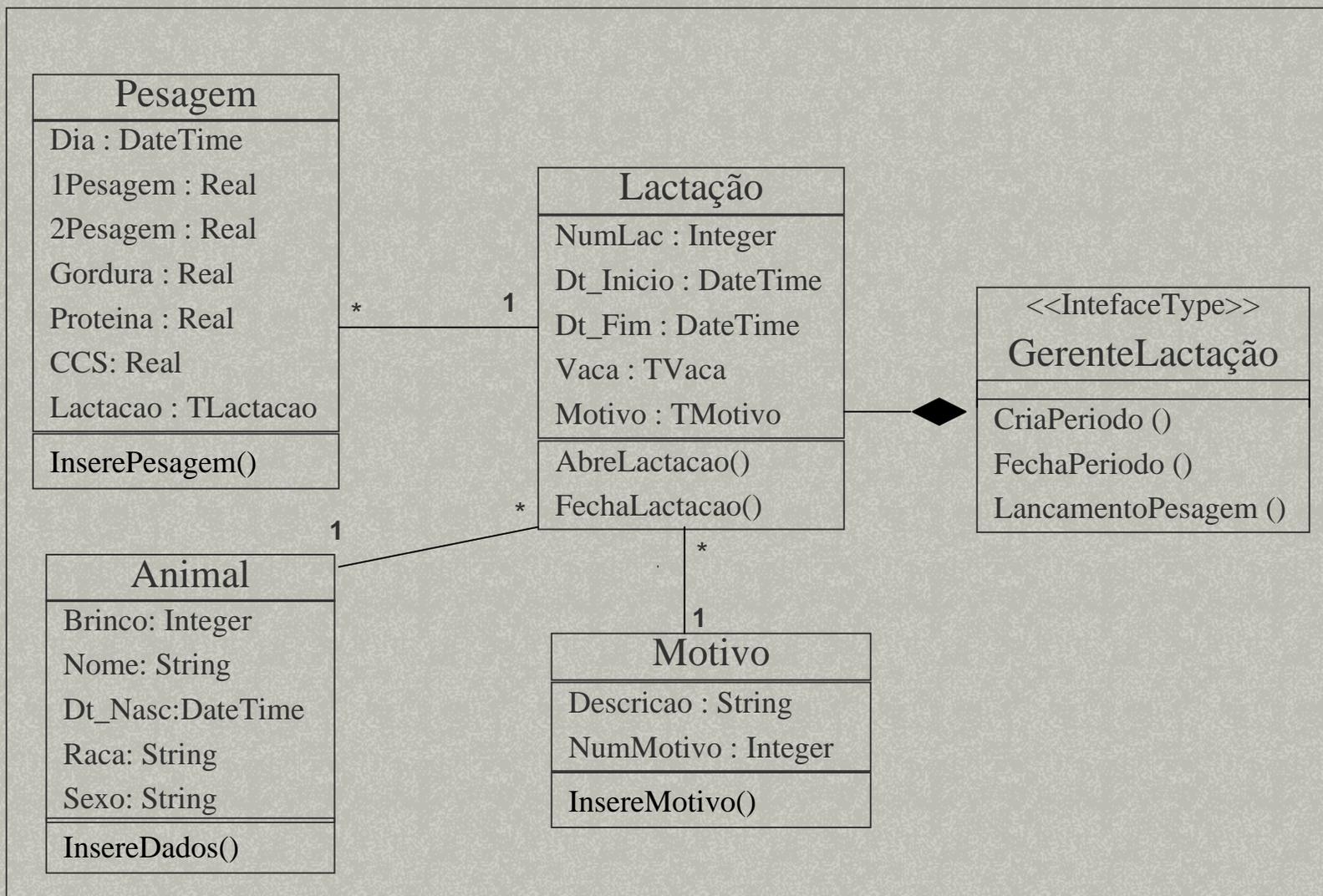
| <<Type>> Lactação |
|----------------------|
| NumLac : Integer |
| Dt_Inicio : DateTime |
| Dt_Fim : DateTime |
| AbreLactacao() |
| FechaLactacao() |

| <<Type>> Pesagem |
|---------------------|
| Dia : DateTime |
| 1Pesagem : Real |
| 2Pesagem : Real |
| Gordura : Real |
| Proteina : Real |
| CCS: Real |
| InsererPesagem() |

| <<Type>> Motivo |
|---------------------|
| Descricao : String |
| NumMotivo : Integer |
| InsererMotivo() |

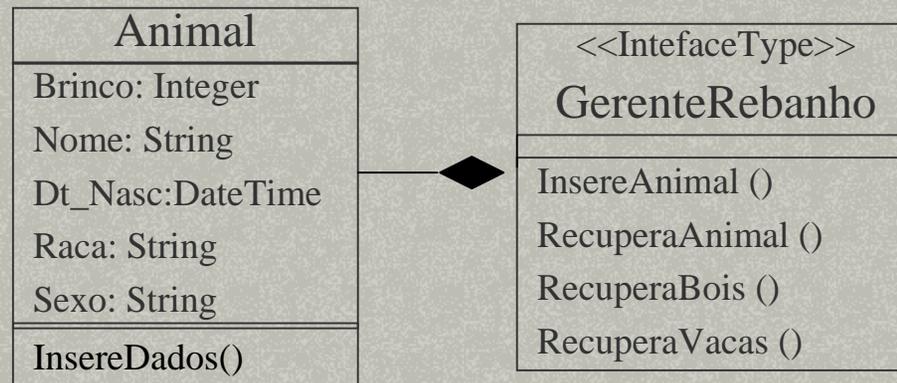


Especificação do componente



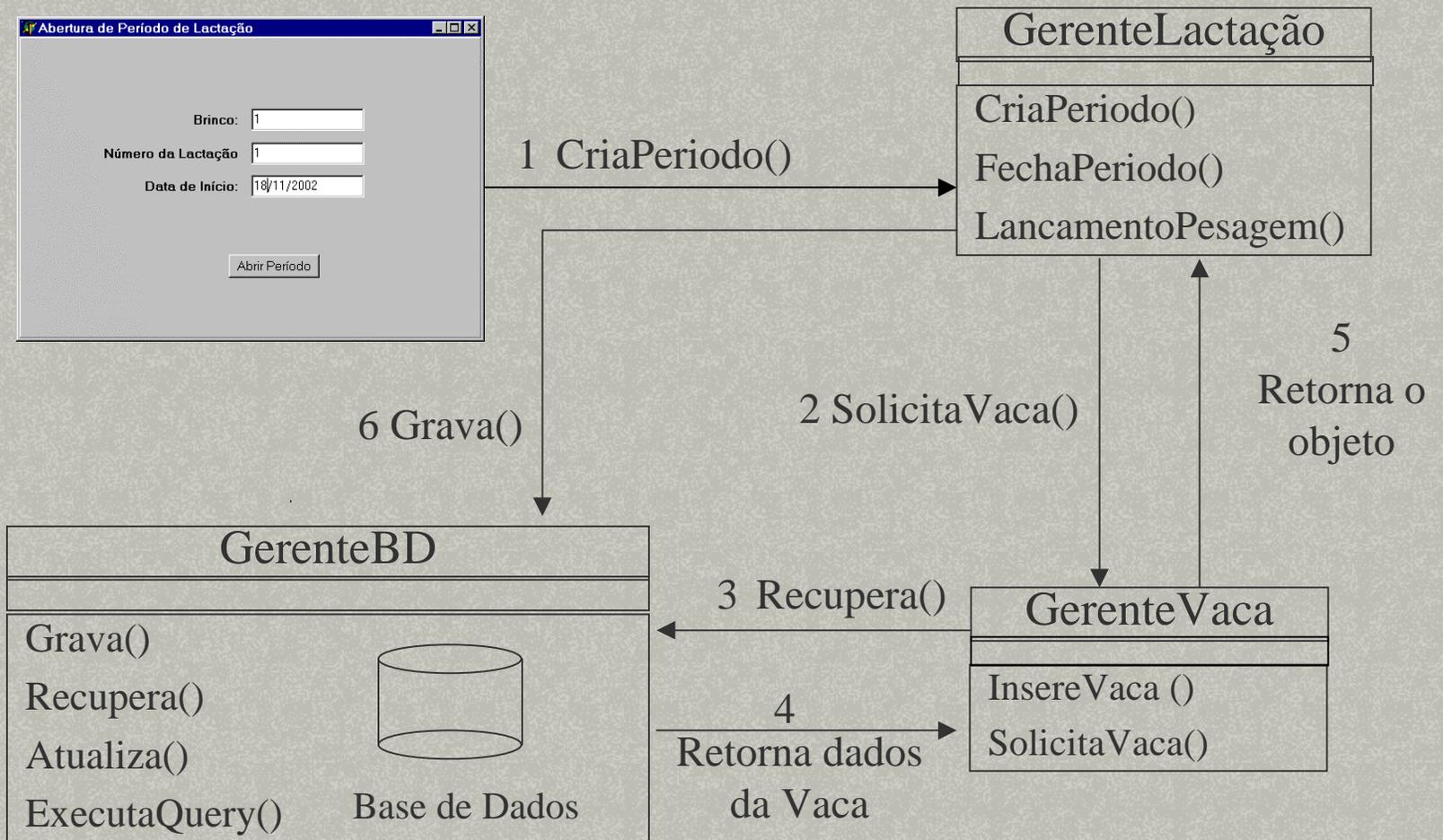


Especificação do componente



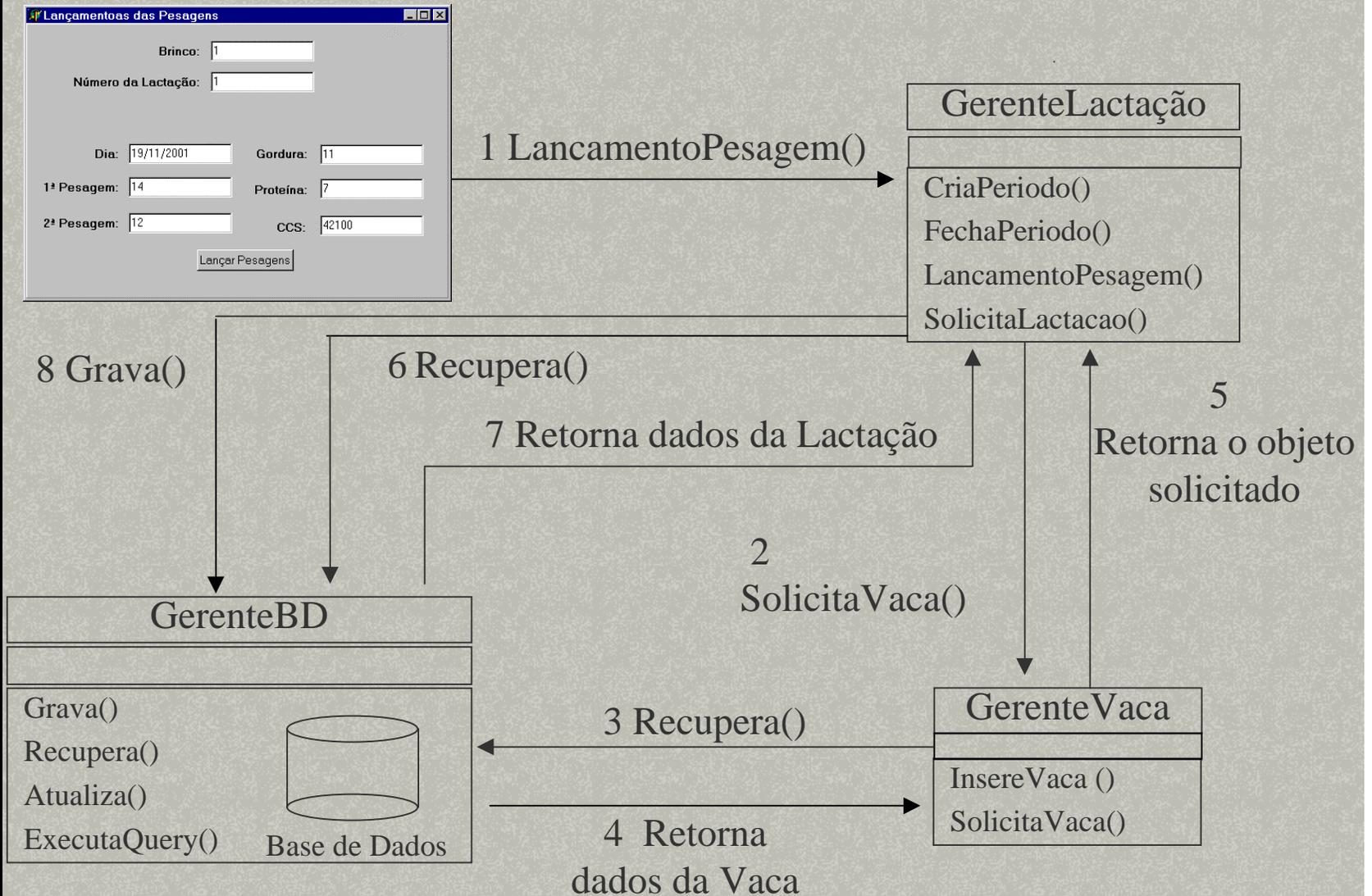


Arquitetura de uma aplicação que utiliza o componente de Lactação





Arquitetura de uma aplicação que utiliza o componente de Lactação





Universidade Federal de Juiz de Fora
Núcleo de Pesquisa em Qualidade de Software

Fernanda Campos

Regina Braga

fcampos@dcc.ufjf.br

EQPS Indaiatuba, SP - 2003