

**UNIVERSIDADE ESTADUAL DO OESTE DO PARANÁ**  
**CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS**  
**CURSO DE INFORMÁTICA**  
**DISCIPLINA: BANCO DE DADOS I**

**METODOLOGIA DE PROJETO**  
**DE BANCO DE DADOS (UML)**

**Cascavel**

**2009**

**ALESSANDRO RODRIGO FRANCO**

**FERNANDO MARTINS**

**RAFAEL ALMEIDA DE OLIVEIRA**

**METODOLOGIA DE PROJETO**

**DE BANCO DE DADOS (UML)**

Trabalho apresentado a disciplina de Banco de Dados I, para o Curso de Informática, do Centro de Ciências Exatas e Tecnológicas, da Universidade Estadual do Oeste do Paraná.

**Orientador: Prof. José Maria Olguin**

**Cascavel**

**2009**

## SUMÁRIO

<b>RESUMO.....</b>	<b>5</b>
<b>ABSTRACT.....</b>	<b>6</b>
<b>INTRODUÇÃO.....</b>	<b>7</b>
<b>1. PROJETO E IMPLEMENTAÇÃO DE BANCO DE DADOS.....</b>	<b>8</b>
1.1. Levantamento e Análise de Requisitos.....	8
1.2. Projeto Conceitual do Banco de Dados.....	9
1.2.1. Projeto do Esquema Conceitual.....	9
1.2.2. Projeto das Transações.....	9
1.3. Escolha de um SGBD.....	10
1.4. Mapeamento do Modelo de Dados.....	11
1.5. Projeto Físico do Banco de Dados.....	11
1.6. Implementação e Sintonização do Sistema de Banco de Dados.....	12
<b>2. USO DE DIAGRAMAS UML COMO APOIO PARA A ESPECIFICAÇÃO DE PROJETO DE BANCO DE DADOS .....</b>	<b>14</b>
2.1. Os Diferentes Diagramas UML.....	14
<b>3. MODELAGEM E PROJETO DE UM BANCO DE DADOS UNIVERSITÁRIO.....</b>	<b>17</b>
Figura 1: Exemplo de Diagrama de Casos de Uso.....	17
Figura 2: Exemplo Diagrama de Estados.....	18
Figura 3: Exemplo Diagrama de Sequência.....	19

<b>4. RATIONAL ROSE.....</b>	<b>20</b>
4.1. Rational Rose Data Modeler.....	20
4.2. Usando o Modelador de Dados da Rational Rose.....	20
<b>CONCLUSÃO.....</b>	<b>21</b>
<b>REFERÊNCIAS.....</b>	<b>22</b>

## RESUMO

Um sistema confiável, íntegro, seguro, com facilidade de uso depende de vários fatores durante seu ciclo de vida desde a fase de levantamento de requisitos até a fase final onde a implementação é finalizada e são realizados os testes em buscas de falhas.

Um ponto importante é a fase de projeto do sistema. Ela é crucial para o sucesso de um sistema. Não queremos discutir nenhuma metodologia específica, mas o projeto de banco de dados de uma maneira ampla e objetiva, como é tratado em grandes organizações onde projetos e implementações de aplicações suprem centenas ou milhares de usuários.

Um projeto que envolva banco de dados com até 20 usuários, não é complicado. A partir daí é necessária uma abordagem mais sistemática para a atividades de projeto de banco de dados global.

Para exemplificar o assunto explanado nesse artigo usaremos um exemplo ilustrativo parcial de um banco de dados universitário. Por fim, introduziremos ainda, uma excelente ferramenta, chamada Rational Rose.

O objetivo desse artigo é expor de maneira clara metodologias para projetos de banco de dados usando diagramas e técnicas de UML.

## **ABSTRACT**

A reliable, integrated, safe, easy to use depends on several factors during its life cycle from the stage of gathering requirements to the final stage where the implementation is completed and tests are performed in search of flaws.

An important point is the design phase of the system. It is crucial to the success of a system. We do not discuss any specific methodology, but the project database of a broad and objective way, as is treated in large organizations where projects and application deployments supply hundreds or thousands of users.

A project involving database with up to 20 users, is not complicated. From there we need a more systematic approach to the project activities of global database.

To illustrate the subject explained in this article we will use an illustrative example of a partial database university. Finally, we introduce also an excellent tool called Rational Rose.

The aim of this article is to clearly methodologies for projects database using diagrams and UML techniques.

## INTRODUÇÃO

Um projeto de banco de dados possuem metas a serem cumpridas. Entre elas podemos destacar:

- Satisfazer os requisitos de informações especificadas por usuários e aplicações.
- Proporcionar uma estruturação natural e fácil para entender a informação.
- Dar suporte a quaisquer requisitos de processo e objetivos de desempenho, como tempo de resposta, tempo de processamento e espaço de armazenamento.

## **1. PROJETO E IMPLEMENTAÇÃO DE BANCO DE DADOS**

Para cumprir com as metas citadas acima precisa-se que se realize as fases do projeto, que são mostradas a seguir:

- Levantamento e análise de requisitos;
- Projeto conceitual de bando de dados;
- Escolha de um SGBD;
- Mapeamento do modelo de dados;
- Projeto físico do banco de dados;
- Implementação e sintonização do sistema de banco de dados.

### **1.1. Levantamento e Análise de Requisitos**

Para projetar de maneira eficaz um banco de dados, temos que realizar um levantamento e análise de requisitos. Esse processo é basicamente conhecer as expectativas dos usuários. A especificação dos requisitos são baseadas em identificar as partes do sistema de informação que interagirão com o sistema. Como parte dessa fase podemos citar:

- Identificação dos grupos de usuários do BD e áreas de aplicação.
- Estudo e análise da documentação e revisão dos demais documentos, como manuais, formulários e relatórios.
- Estudo do ambiente operacional, análise dos tipos de transação, fluxo de informação no sistema.
- Respostas a conjuntos de consultas realizadas por usuários, que envolvem as prioridades dos usuários e a importância que eles atribuem as diversas aplicações.

Esses procedimentos auxiliam na definição da estrutura e na refinação dos requisitos. Além disso, para transformar esses requisitos em uma forma mais estruturada são usadas técnicas para especificação de requisitos. Essa especificação possui métodos que utilizam técnicas de diagramação para organizar e apresentar os requisitos.

Essa fase requer muito tempo, e é crucial para o sucesso do sistema de informação. Corrigir um erro que ocorra nessa fase é muito mais caro que corrigir um erro de implementação.

## **1.2. Projeto Conceitual do Banco de Dados**

Nessa fase do projeto temos duas atividades em paralelo. Uma delas é o projeto do esquema conceitual, que é resultado da fase 1 e examina os resultados de dados. A outra é o projeto das transações e aplicações que produz especificações de alto nível com base nas aplicações de banco de dados, analisadas na fase 1.

### **1.2.1. Projeto do Esquema Conceitual**

Nessa fase é importante usar dados de alto nível com as seguintes características:

- *Expressividade*: para distinguir os diferentes tipos de dados, relacionamentos e restrições.
- *Simplicidade e inteligibilidade*: para que usuários amadores possam entender seus conceitos.
- *Sintético*: número reduzido de conceitos básicos, que sejam distintos.
- *Representação Diagramática*: exibição do esquema conceitual de modo que facilite a interpretação.
- *Formalismo*: especificação formal não ambígua dos dados. Os modelos conceituais devem ser definidos com precisão e sem ambigüidade.

A identificação dos componentes básicos do esquema como: tipos entidade, tipos relacionamento e atributos é de suma importância para o projeto de um esquema conceitual.

### **1.2.2. Projeto das Transações**

Essa fase que ocorre em paralelo com a fase anterior, tem o propósito de

projetar as características das aplicações (transações). Uma parte de suma importância do projeto é a descrição das características funcionais das aplicações no início do processo do projeto. Isso nos garante que o esquema do banco de dados incluirá todas as informações necessárias as transações.

Uma técnica amplamente utilizada na especificação de transações em nível conceitual é identificar suas entradas/saídas e seu comportamento funcional. Após a especificação do fluxo funcional interno de controle e dos parâmetros de entrada e saída os projetistas podem determinar uma transação de modo conceitual e independente do sistema.

As transações, basicamente, são agrupadas em três categorias:

- *Transações de Recuperação*: Usadas para recuperar dados para exibição em uma tela ou para produção de relatório;
- *Transações de Atualização*: Usadas para entrada ou modificação de dados.
- *Transações Mistas*: Usadas para aplicações mais complexas que fazem recuperação e atualização.

O projeto de transações é tão importante quanto o de esquema, mas é frequentemente considerado parte do software em vez de parte do projeto de banco de dados.

### **1.3. Escolha de um SGBD**

Vários fatores determinam a escolha de um SGBD, como, técnicos, econômicos, e ligados a política de organização. Vamos nos concentrar nos fatores econômicos e organizacionais. Os seguintes custos devem ser considerados:

- *Custo para aquisição de software*: Inclui opções de linguagem, diferentes interfaces, menus, ferramentas, opções de backup.
- *Custo de manutenção*: Ocorre periodicamente ao ser contratada a manutenção padrão, ou para manter a versão do SGBD atualizada.
- *Custo de aquisição de hardware*: Aquisição de memória adicional,

terminais, unidades de disco.

- *Custo de criação e conversão do banco de dados:* Custo para criar o sistema de banco de dados, ou a conversão de um sistema existente no novo software do SGBD.
- *Custo de pessoal:* Frequentemente a primeira aquisição de software de SGBD é acompanhada pela criação, ou reorganização do departamento de processamento de dados.
- *Custo de treinamento:* Pela complexidade do SGBD, frequentemente haverá necessidade de treinamento de pessoal para usar e programar o SGBD.
- *Custo operacional:* Custo que não é normalmente considerado nas alternativas de avaliação porque ele ocorre independentemente do SGBD selecionado.

A aquisição de um SGBD trás imensuráveis benefícios difíceis de medir e quantificar, pois um SGBD possui vantagens intangíveis sobre os sistemas de arquivos tradicionais. Benefícios mais tangíveis incluem o custo reduzido para o desenvolvimento de aplicações, redução de dados e melhor controle e segurança.

#### **1.4. Mapeamento do Modelo de Dados**

O mapeamento do modelo de dados é considerado o Projeto Lógico do Banco de Dados. O mapeamento pode ocorrer em dois estágios:

- *Mapeamento de Sistema Independente:* Não considera nenhuma característica específica ou casos que se aplicam a implementação do modelo de dados do SGBD.
- *Adaptando esquemas a um SGBD específico:* SGBDs diferentes implementam um modelo de dados usando características específicas de modelagem e restrições.

#### **1.5. Projeto Físico do Banco de Dados**

O projeto físico do banco de dados é o processo para escolher quais serão as estruturas de armazenamento específicas e dos caminhos de acesso para os arquivos de banco de dados, de modo a alcançar um bom desempenho nas várias aplicações de banco de dados.

As várias opções oferecidas por um SGBD incluem vários tipos de índices, agrupamento de registros relacionados em blocos de disco. A escolha das estruturas mais apropriadas para os arquivos do banco de dados depende da escolha do SGBD. Para a escolha de um projeto físico de banco de dados devem ser seguidos os seguintes critérios:

- *Tempo de Resposta:* Tempo que decorre após a submissão de uma transação para o banco de dados, sua execução e resposta.
- *Utilização do Espaço:* Todo o espaço usado pelos arquivos do banco de dados e por suas estruturas de caminho de acesso em disco.
- *Taxa de Processamento de transações:* Média das transações processadas por minuto, devem ser medidas em situações de pico do sistema.

Os resultados obtidos na fase de projeto físico do banco de dados é a determinação inicial das estruturas de armazenamento e dos caminhos de acesso para os arquivos do banco de dados.

## **1.6. Implementação e Sintonização do Sistema de Banco de Dados**

A implementação se dá após a finalização dos projetos lógicos e físicos, e fica sob responsabilidade do Administrador do Banco de Dados (DBA). O banco de dados deve ter suas transações implementadas pelos programadores de aplicações, partindo das especificações conceituais das transações. Uma vez pronta as transações com os dados carregados no banco de dados, termina a fase de projeto e implementação e começa a fase operacional do sistema de banco de dados.

Vários sistemas possuem estatísticas de desempenho que incluem número de chamadas de transações para consultas predefinidas, atividades de entrada/saída em arquivos.

Para se obter um melhor desempenho pode-se reescrever algumas consultas ou transações. A sintonização do BD continua, e ao longo de sua existência os requisitos podem mudar eventuais problemas descobertos.

## 2. USO DE DIAGRAMAS UML COMO APOIO PARA A ESPECIFICAÇÃO DE PROJETO DE BANCO DE DADOS

Uma abordagem que vem tendo grande atenção e credibilidade é a UML - Unified Modeling Language (Linguagem Unificada de Modelagem), que proporciona um mecanismo de notação diagramática associado a uma sintaxe de linguagem para cobrir todo o ciclo de vida. Atualmente é usada por desenvolvedores de software, modeladores e projetistas de dados, para exibir a especificação detalhada de uma aplicação.

Através da combinação de conceitos e métodos de OO e metodologias, a UML é aplicável a qualquer domínio e linguagem e permitem aos desenvolvedores de software criarem modelos claros e inteligíveis para a especificação, visualização, construção e documentação de componentes dos sistemas de software.

### 2.1. Os Diferentes Diagramas UML

São nove tipos de diagramas definidos pela UML, que são divididos em duas categorias:

- *Diagramas Estruturais:* Relações estruturais ou estáticas entre os componentes. Fazem parte dos diagramas estruturais:
  - *Diagramas de Classe:*

Capturam a estrutura estática do sistema e servem de base para outros modelos. Mostram as classes, interfaces, colaborações, dependências, generalizações, associação e outros relacionamentos. São extremamente úteis para a modelagem do esquema conceitual do banco de dados.
  - *Diagramas de Objeto:*

Mostram um conjunto de objetos e seus relacionamentos. Podem ser chamados de Diagramas de Instâncias. Nos dão uma visão estática do sistema, e geralmente são usados para testar os diagramas de classe.

- *Diagramas de Componentes:*

Ilustram as organizações e as dependências entre os componentes do software.
- *Diagramas de Desenvolvimento:*

Representam a distribuição dos componentes. Responsáveis pela descrição dos recursos físicos de um sistema, ainda são usados para mostrar a configuração dos elementos em tempo de execução.
- Diagramas de comportamento: Descrevem relações de comportamento entre os componentes. Fazem parte dos diagramas de comportamento:
  - *Diagramas de Caso de Uso:*

São usados para modelar as iterações funcionais entre os usuários e o sistema. É uma sucessão de passos que descrevem iterações entre usuário e sistema. Um caso de uso é um conjunto de cenários que tem uma meta comum. Os diagramas de caso de uso mostram ainda atores interagindo e pode ser facilmente entendido. Como exemplo usamos a figura 1 que ilustra um ambiente universitário no qual os casos de uso ‘matricula em curso’ e ‘informa grade’, tem como atores envolvidos ‘aluno’ e ‘professor’.
  - *Diagramas de Sequência:*

Descrevem as interações entre os vários objetos ao longo do tempo. Através desses diagramas temos uma visão dinâmica do sistema que mostra basicamente o fluxo de mensagens entre os objetos. Em um diagrama de seqüência, um objeto ou um ator é mostrado com uma caixa no topo de uma linha vertical que é a Lina de vida do projeto. Num banco de dados, esse objeto é alguma coisa física, como um livro em uma livraria. A linha de vida representa a existência do objeto ao longo do tempo. A ordem da leitura das mensagens é de cima para baixo.
  - *Diagramas de Colaboração:*

Representam interações entre objetos como séries de mensagens seqüenciadas. Nesses diagramas se dá ênfase na organização estrutural dos objetos que enviam e recebem mensagens. No diagrama de sequências se dá ênfase na ordenação temporal das mensagens. O uso de diagramas de seqüência e colaboração é opcional.

○ *Diagramas de Estado:*

Descrevem as mudanças de estado de um objeto em relação a eventos externos. Basicamente é um diagrama criado para mostrar todos os possíveis estados que um objeto pode assumir durante sua existência. Os elementos de um diagrama de estados são:

- Estados: caixas com cantos arredondados, representam situações na vida de um objeto.

- Transições: setas sólidas entre os estados, são os caminhos entre os estados diferentes de um objeto.

- Estado Inicial: círculo sólido, com uma seta de partida para um estado.

- Estado final: círculo cheio com linha dupla, com uma seta vinda de um estado.

○ *Diagramas de Atividades:*

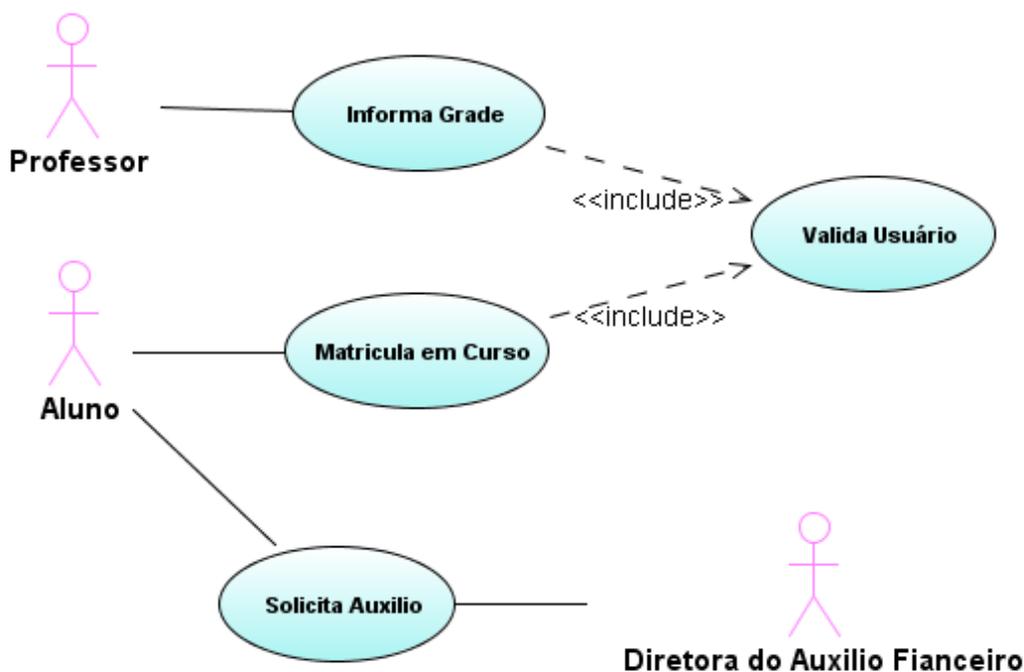
Representam uma visão dinâmica do sistema, modelando o fluxo de controle de uma atividade para outra. Uma atividade pode ser definida como um estado de fazer alguma coisa. Normalmente esses diagramas são utilizados para modelar o fluxo de trabalho.

### 3. MODELAGEM E PROJETO DE UM BANCO DE DADOS UNIVERSITÁRIO.

Será apresentado o projeto de um banco de dados para uma universidade com base em diagramas UML. Explanaremos de maneira genérica o uso dos diagramas, que conduziram ao projeto conceitual e dos componentes de programa.

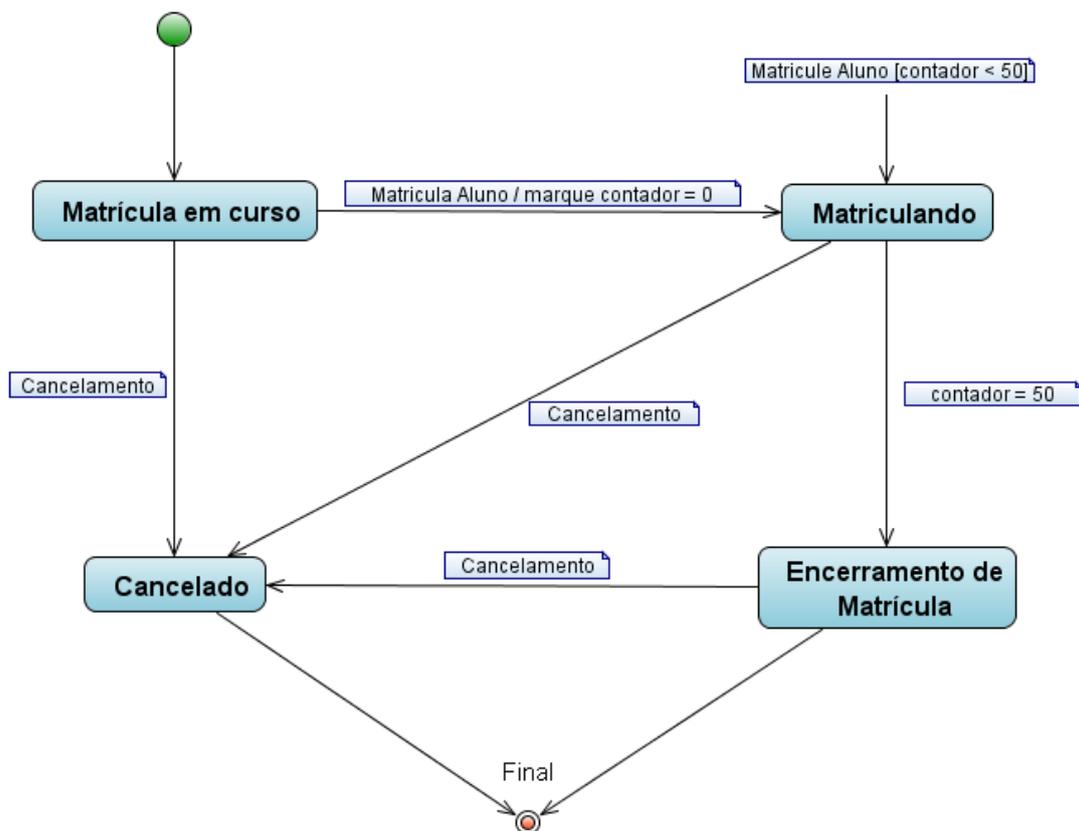
Imagine o seguinte exemplo: um cenário em que alunos se matriculem em cursos oferecidos pelos professores. A encarregada pela programação dos cursos e manutenção de catálogos sobre eles será a secretária. Ela poderá adicionar e cancelar cursos. Os cursos terão limites de vagas. A diretoria de ajuda financeira processa aplicações de apoio aos alunos. O banco de dados terá que manter dados sobre alunos, professores, cursos etc. Deverá possuir funções para inscrição no curso, manutenção de catálogos de curso. Esses requisitos citados acima serão descritos em diagramas UML.

Imagine que um dos requisitos do banco de dados seja a permissão para que os professores entrem com os cursos que irão ministrar e que os alunos possam se matricular e solicitar ajuda financeira. A *figura 1* pode ilustrar de maneira clara essa situação.



**Figura 1:** Diagrama de caso de uso para um banco de dados universitário.

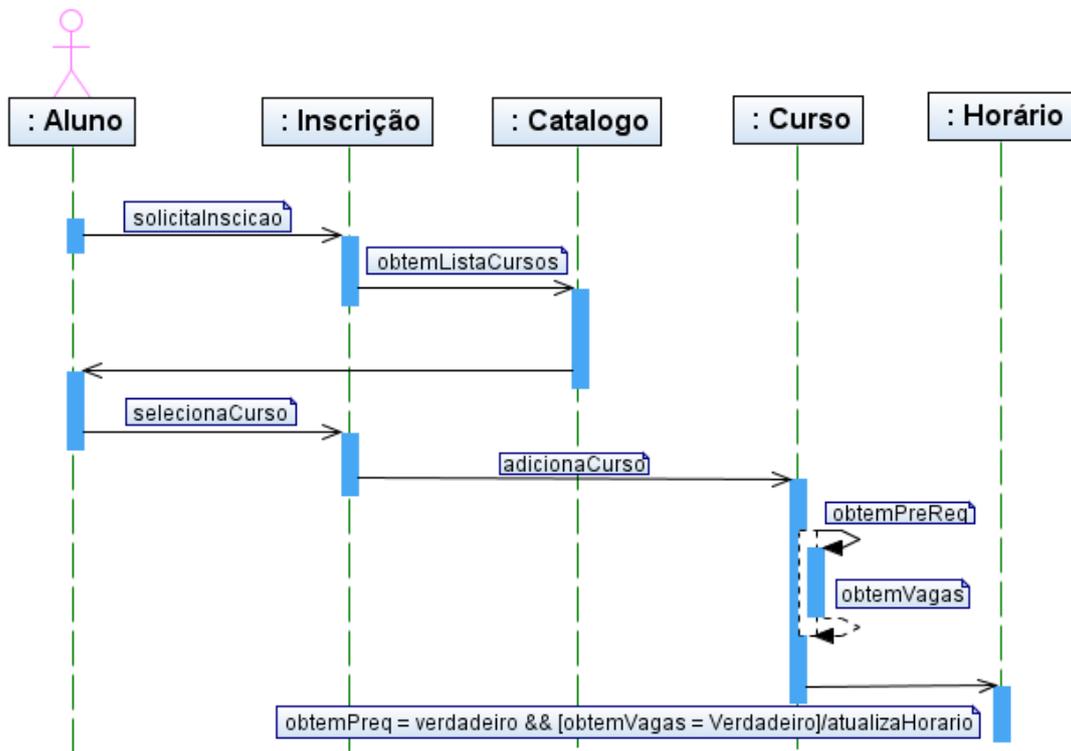
É importante também ilustrar os estados que o sistema poderá vir a assumir, pois nos ajudará a visualizar os diversos estados durante o curso da aplicação. Nesse exemplo que estamos abordando suponha a abertura para inscrições de um curso com 50 vagas. Durante o estado de matrícula, a transição 'Matricula Aluno' continua enquanto o numero de alunos matriculados for menor que 50. Essa representação poderá ser vista na *figura 2*. Perceba que são mostrados os estados do curso enquanto as matrículas estão abertas.



**Figura 2:** Exemplo de diagrama de estados para o banco de dados universitário.

Finalizados o diagrama de casos de uso e o diagrama de estados, podemos fazer o diagrama de sequência, que irá visualizar a execução dos casos de uso. O diagrama de sequência mostrado na figura 3 é referente ao caso de uso onde o aluno faz inscrição e seleciona um curso em particular para se matricular. No diagrama de sequência são conferidos os pré-requisitos e a capacidade do curso. Então o curso é acrescentado ao horário do aluno se os

pré-requisitos forem atendidos.



**Figura 3:** Diagrama de sequência para o banco de dados universitário.

Os diagramas mostrados anteriormente não são a especificação completa do banco de dados. Haverá diversos outros casos. As metodologias de projeto estão sujeitas a diversos tipos de julgamento, preferências pessoais etc. O que podemos estar certos é que os diagramas de classe responderá pelas especificações que foram determinadas pelos casos de uso, diagramas de estados e sequência.

## **4. RATIONAL ROSE**

A Rational Rose é uma ferramenta para projeto de banco de dados baseada em UML. É uma das ferramentas mais importantes usada na indústria para o desenvolvimento de sistemas de informação. A Rational Rose nos dá a possibilidade da especificação inicial em UML que nos conduzirá ao desenvolvimento do banco de dados. Atualmente a ferramenta dá suporte para projetos e modelagem conceitual, lógica e física de banco de dados.

### **4.1. Rational Rose Data Modeler**

É uma ferramenta de modelagem visual para banco de dados. Seu diferencial é que ela é baseada em UML. Ela possibilita o trabalho em conjunto de projetistas de banco de dados, desenvolvedores e analistas. Outra importante vantagem do Rose é sua capacidade de processo de modelagem que permite modelar o comportamento do banco de dados. Essa ferramenta proporciona ainda capacidade de engenharia avante do banco de dados, possibilitando atualizações constantes dos requisitos, e de engenharia reversa para o projeto conceitual de um banco de dados já implementado.

### **4.2. Usando o Modelador de Dados da Rational Rose**

O modelador de dados da rational rose permite criar um modelo de dados com base na estrutura do banco de dados ou criar um banco de dados com base no modelo de dados.

- Projeto Conceitual em Notação UML: Como já foi dito uma das vantagens da Rational Rose é a modelagem do banco de dados utilizando notação UML. Os diagramas de ER podem facilmente ser construídos utilizando os diagramas de classe da Rational Rose.

## **CONCLUSÃO**

Manipular manualmente toda a complexidade de relacionamento e restrições das aplicações existentes em um banco de dados realmente não é uma tarefa fácil. Devido a isso o numero de opções de projetos para modelar essas informações aumenta rapidamente.

Foi devido a esses fatores que se originaram as ferramentas para projeto de banco de dados. A utilização de diagramas e outros modelos UML vem só para adicionar mais facilidades no processo.

## REFERÊNCIAS

Elmasri, Ramez E., Navathe, Shamkant B. (2004), Fundamentals of Database Systems, Addison Wesley, 4<sup>th</sup> edition.