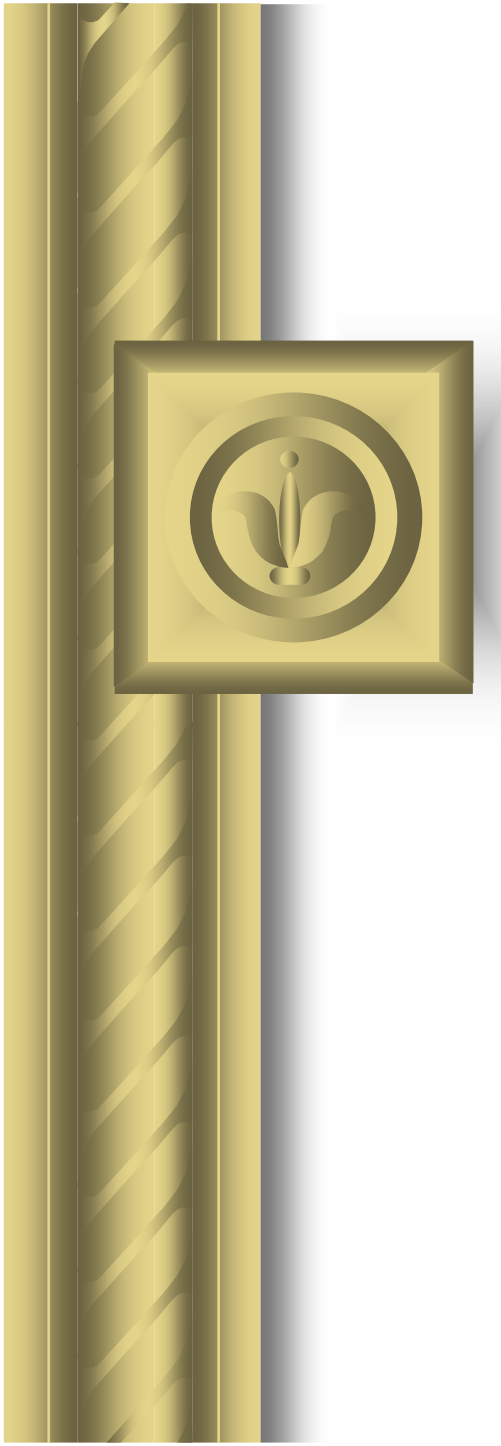


Estudo do SGBD “Caché” com uma aplicação na reserva de vagas em eventos acadêmicos via web

Acadêmico: Aloisio Arbegaus

Orientador (Prof.): Marcelo José Ferrari

Roteiro

- Introdução
  - Objetivos
- Fundamentação Teórica
  - Evolução dos B.D.
    - Modelo Hierárquico
    - Modelo de Rede
    - Modelo Relacional
  - Banco de Dados Orientado a Objetos
    - Conceitos de OO para SGBDOO
    - Conceitos de BD para SGBDOO
  - Banco de Dados na Web
  - O Banco de dados Caché
    - O modelo multidimensional
    - As Formas de Acesso aos dados do Caché
    - O Caché Objects
- Desenvolvimento do Protótipo
  - Especificação
    - Técnicas e Ferramentas Utilizadas
    - Apresentação da Especificação
  - Implementação
    - Técnicas e Ferramentas Utilizadas
    - Integração Web via JDBC com o caché
    - Operacionalidade da Implementação
- Conclusão
  - Sugestões para trabalhos futuros

1. Introdução

Segundo Pinheiro (2000), o modelo de dados orientado a objetos possui algumas diferenças com relação ao modelo relacional tradicional. As principais são:

a) maneira de organizar a informação;

b) representação do modelo comportamental, que na verdade são os métodos dos objetos para manipular os dados armazenados.

Dessa forma, o modelo de dados orientado a objetos permite o projeto estrutural e comportamental da base de dados, através da definição dos objetos com seus atributos e métodos.

1.1 Objetivos

O objetivo é mostrar como o *Caché* manipula e administra seus objetos, bem como, permite a uma aplicação WEB fazer consultas SQL no banco via JDBC.

Os objetivos específicos do trabalho são:

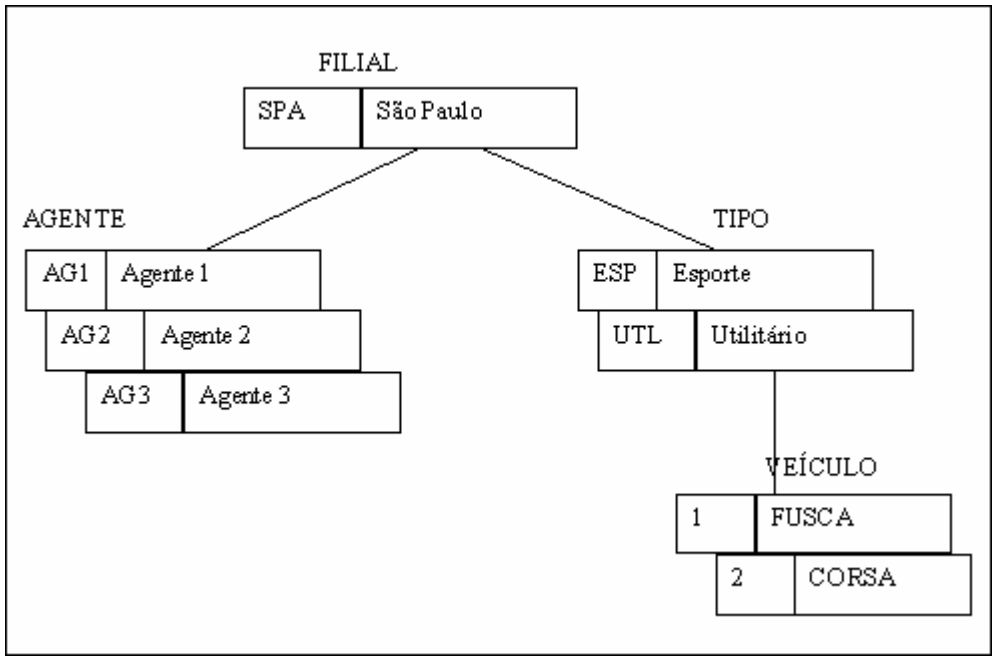
a) mostrar o uso da orientação a objetos desde o desenvolvimento de uma aplicação WEB até o armazenamento dos objetos no banco de dados *Caché*;

b) especificar, modelar e implementar um sistema básico para reservas com interface WEB.

2. Evolução dos Banco de dados

2.1 O modelo hierárquico

Cada registro, em um banco de dados hierárquico, pode ter quantos descendentes quiser, mas somente um ascendente (exceto a raiz, que não possui ascendentes). Dessa forma, as relações entre pais (nós ascendentes) e filhos (nós descendentes) são de um nó pai para zero ou muitos nós filhos (figura 2.1).

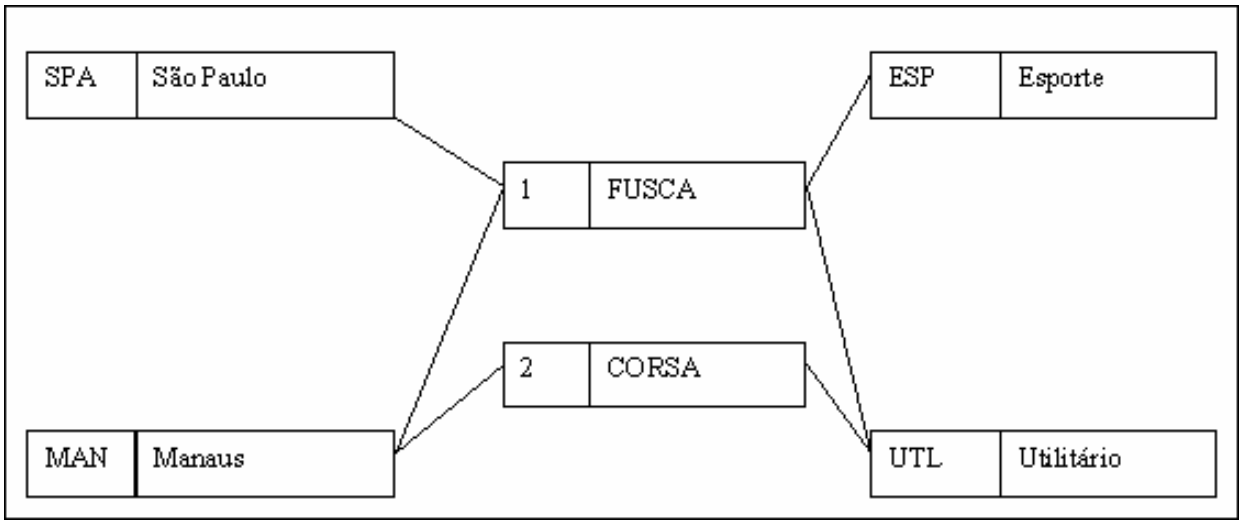


Fonte: adaptado de Kern (1994)

Figura 2.1 – Representação Hierárquica de Registros

2.2 O modelo de rede

Segundo Kern (1994), um banco de dados de rede consiste em uma visão de grão ou malha de ligações um para muitos entre os registros. Um tipo de registro pode estar envolvido em vários relacionamentos e pode ter diversos ascendentes e descendentes.



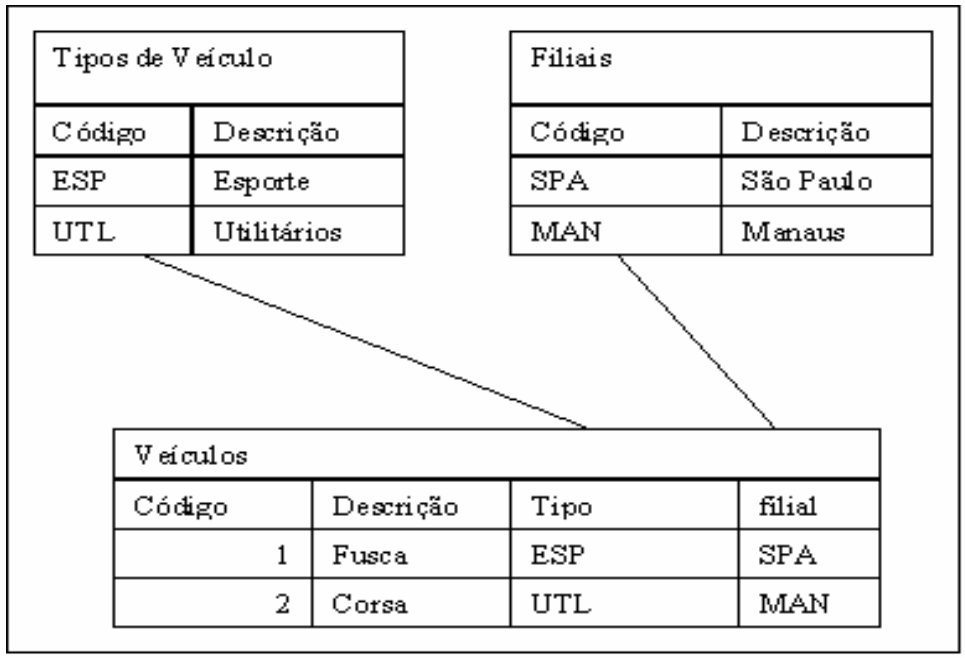
Fonte: adaptado de Kern (1994)

Figura 2.2 – Representação em Rede de Registros (Adaptado de Kern 1994).

2.3 O modelo relacional

O modelo de dados relacional foi criado por E.F. Codd e baseia-se em um único conceito: a tabela (figura 2.3).

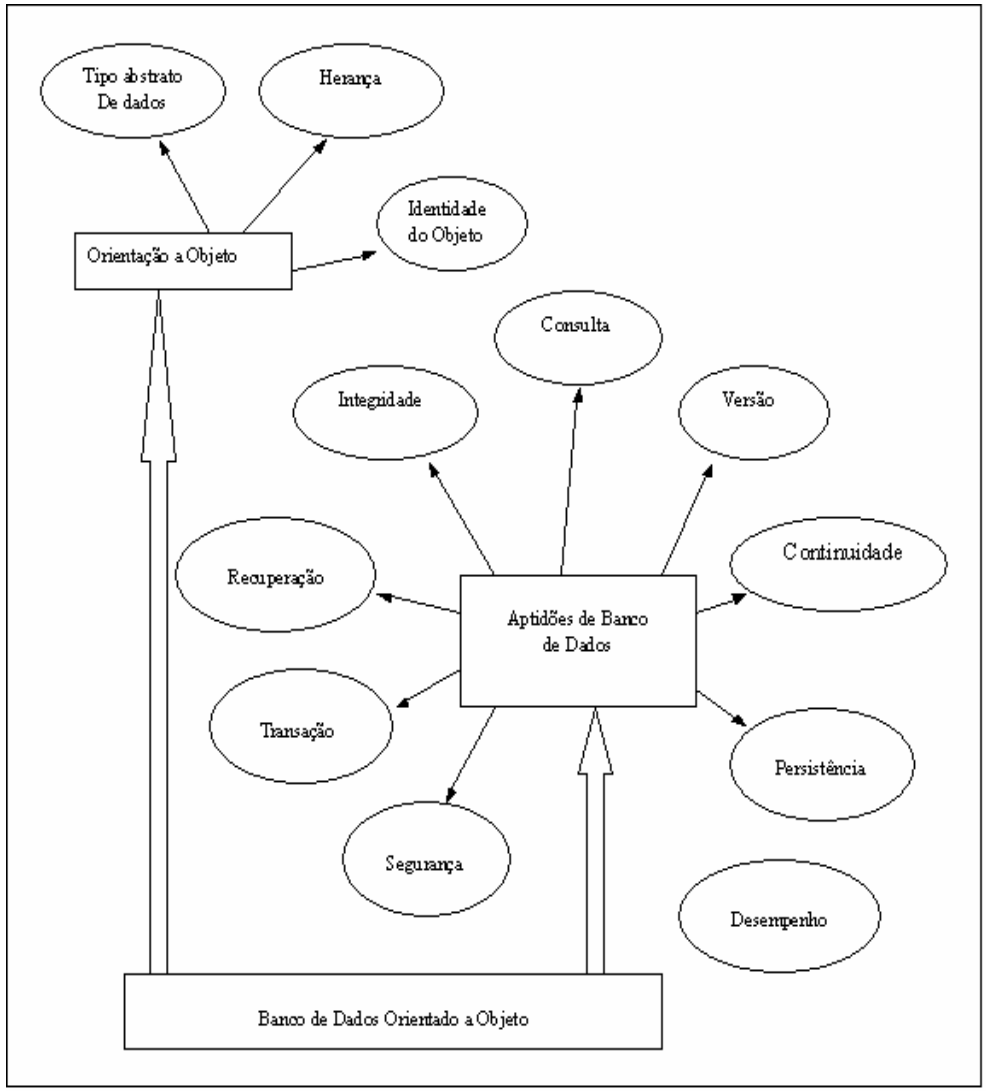
Para Kern (1994) o modelo relacional é definido como aquele no qual os dados são percebidos pelos usuários como tabelas e as operações aplicáveis ao sistema geram tabelas a partir das primeiras.



Fonte: adaptado de Baehr Jr. (1999)

Figura 2.3 – Um exemplo de Banco de Dados Relacional

3. Banco de dados orientado a objetos



Fonte: Khoshafian (1994)

Figura 3.1 – Banco de Dados Orientados a Objeto

3.1 Conceitos de O.O para SGBDOO

Objetos para banco de dados:

a) Identidade de objetos;

b) Objetos complexos

i. Objetos embutidos

ii. Objetos referenciados.

Hierarquia de classes e herança para banco de dados

3.2 Conceitos de B.D para SGBDOO

a) Transações;

b) Concorrência;

c) Recuperação;

d) Versionamento.

4. Banco de dados na web

4.1 Integrando web e banco de dados

De modo geral pode-se enumerar as seguintes vantagens da integração Web e Banco de Dados:

a) mecanismo eficientes;

b) disponibilidade de informações;

c) expansão de novas tecnologias.

4.2 Problemas de integração web e banco de dados

a) sistemas legados;

b) transacionais;

c) segurança e acesso ao banco de dados;

d) controle de restrições de integridade.

5. O banco de dados caché

Produto da Intersystems Corporation, os aplicativos desenvolvidos em *Caché* oferecem, Intersystems (2002):

a) alta disponibilidade e segurança de dados;

b) eliminação de qualquer armazenamento desnecessário de informações;

c) modificação dos modelos de dados de maneira eficiente;

d) superioridade de recursos em relação ao modelo relacional;

e) um fácil e rápido modelo de aplicação utilizando objetos;

f) uma nova LPOO: o *Caché Object Script*;

g) alta performance, através dos objetos;

h) rápido desenvolvimento de aplicações.

5.1 O modelo de dados multidimensional

No modelo de dados multidimensional a mesma estrutura física pode tomar quantas dimensões forem necessárias para modelá-la de forma a atender as reais necessidades da aplicação.

O modelo multidimensional do *Caché* mapeia os dados em estruturas do tipo árvore, onde cada índice pode indicar o início de uma sub-árvore. Qualquer ramo abaixo de um nó está automaticamente relacionado com ele.

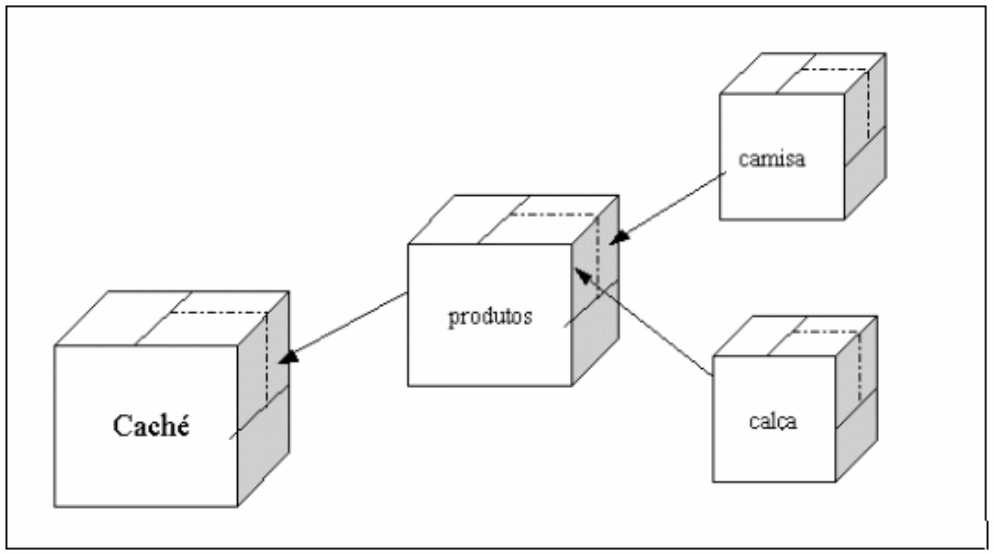
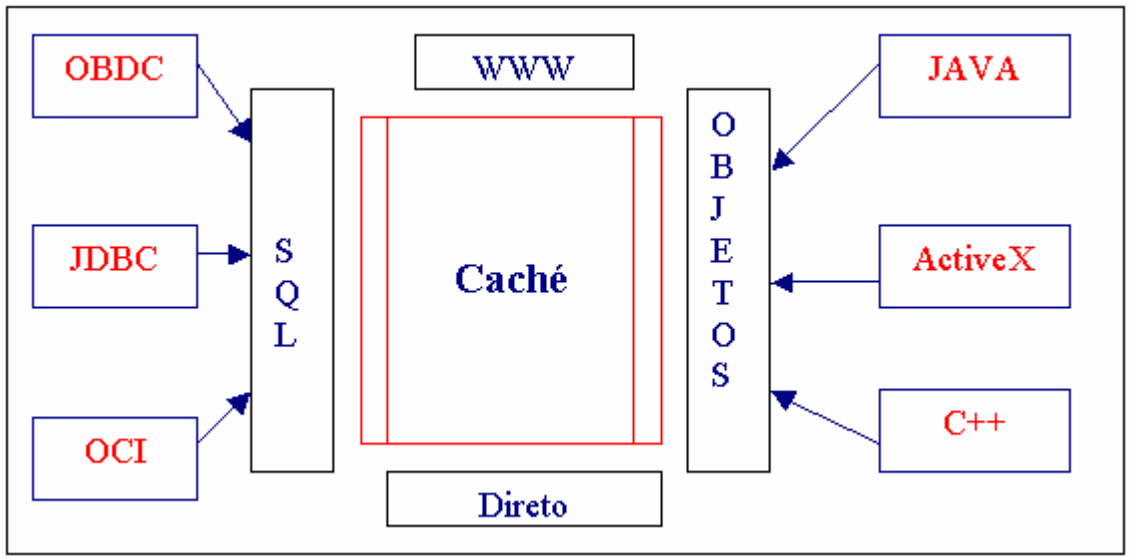


Figura 5.1 Armazenamento em cubo (matrizes multidimensionais)

5.2 As formas de acesso aos dados do caché



Fonte: Baher Jr.(1999)

Figura 5.2 o Caché e suas diversas formas de acesso

5.2 As formas de acesso aos dados do caché

a) O acesso SQL:
   i. ODBC (*Open DataBase Connectivity*);

   ii. JDBC (Java DataBase Connectivity):

   iii. OCI (Oracle Calling interface).

b) O caché weblink;

c) O acesso ao objeto;
   i. suporte a segurança, inclusive herança múltipla;

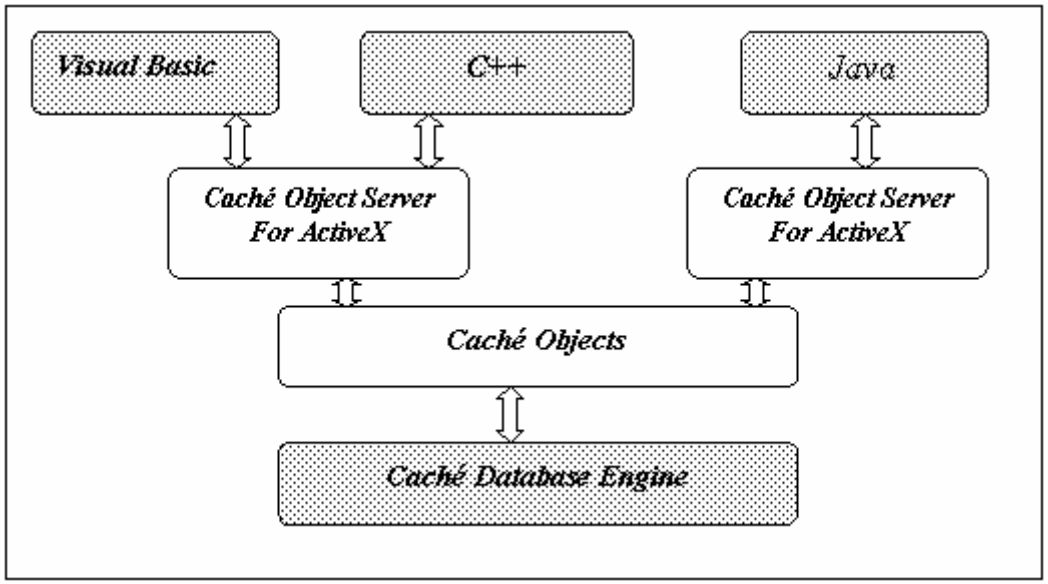
   ii. *advanced data types* (Tipos de Dados Avançados);

   iii. Um modelo completo de OO, incluindo OID, referências entre objetos e objetos “embutidos”.

d) O acesso direto.

5.3 O caché objects

O *Caché Objects* é o componente (figura 5.3) da estrutura do banco de dados *Caché* responsável por disponibilizar, simultaneamente, o desempenho e o poder de modelagem da estrutura multidimensional de dados do *Caché* associada às características da tecnologia OO.



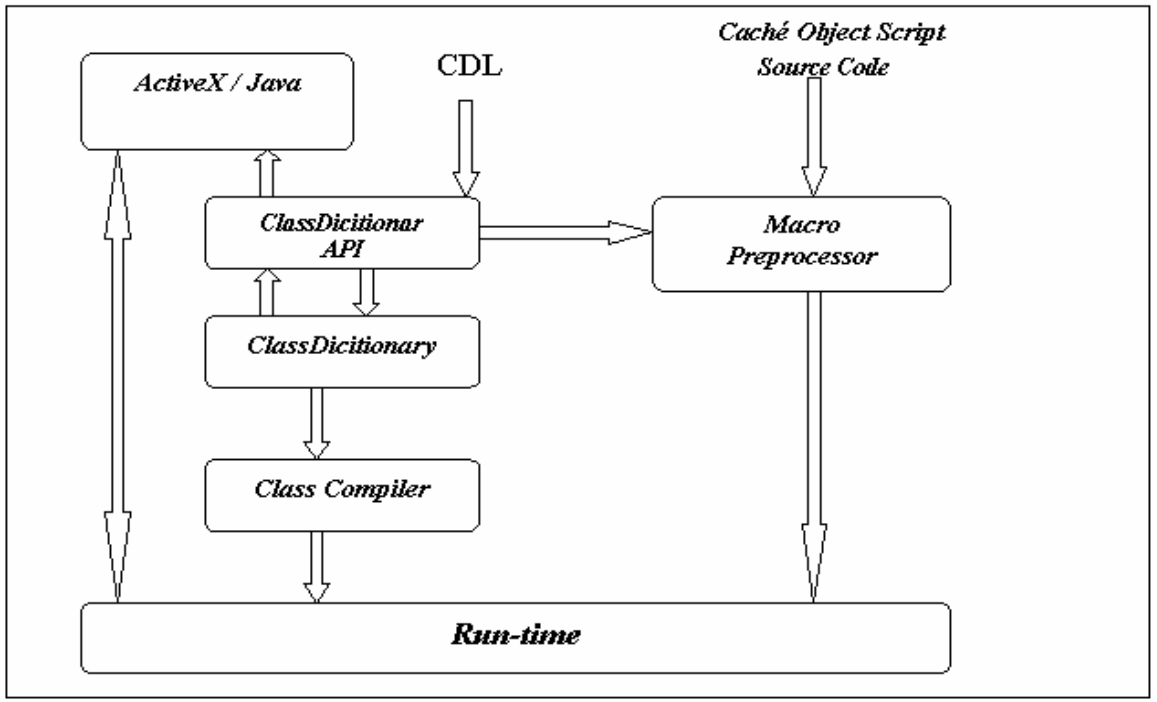
Fonte: Intersystems (1998)

Figura 5.3: Visualização do Caché Objects

5.3.1 A estrutura do caché objects

O *Caché Objects* é um sistema (figura 5.4) formado pelos seguintes sub-sistemas:

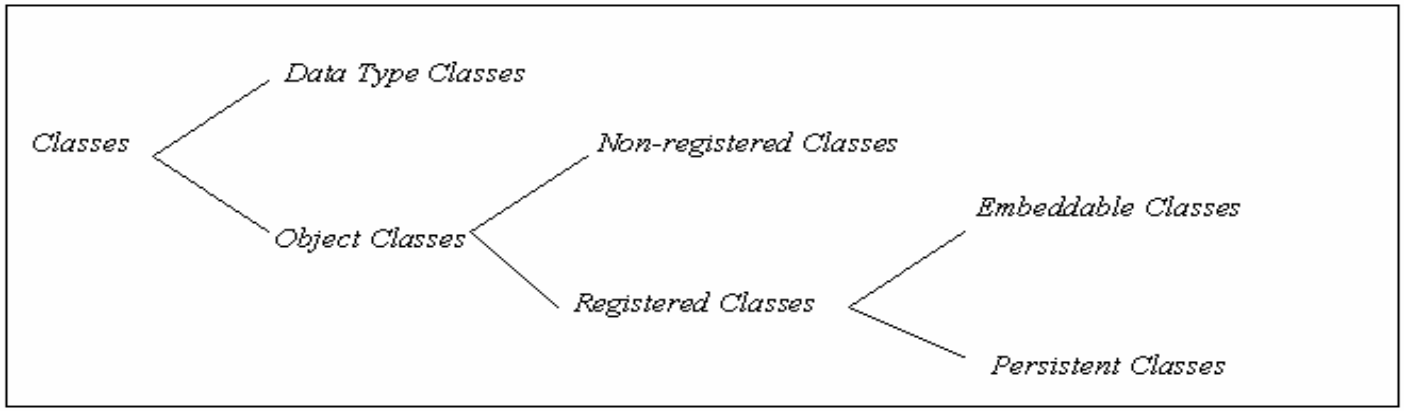
a) ClassDictionary;
b) ClassDictionary Application Program Interface (ClassDictionary API);
c) ClassCompiler;
d) Caché Object Server for ActiveX;
e) Caché Object Server for Java;
f) Macro Preprocessor.



Fonte: Intersystems (1998)
Figura 5.4: Os Subsistemas do *Caché Objects*.

5.3.2 O modelo de objetos do caché objects

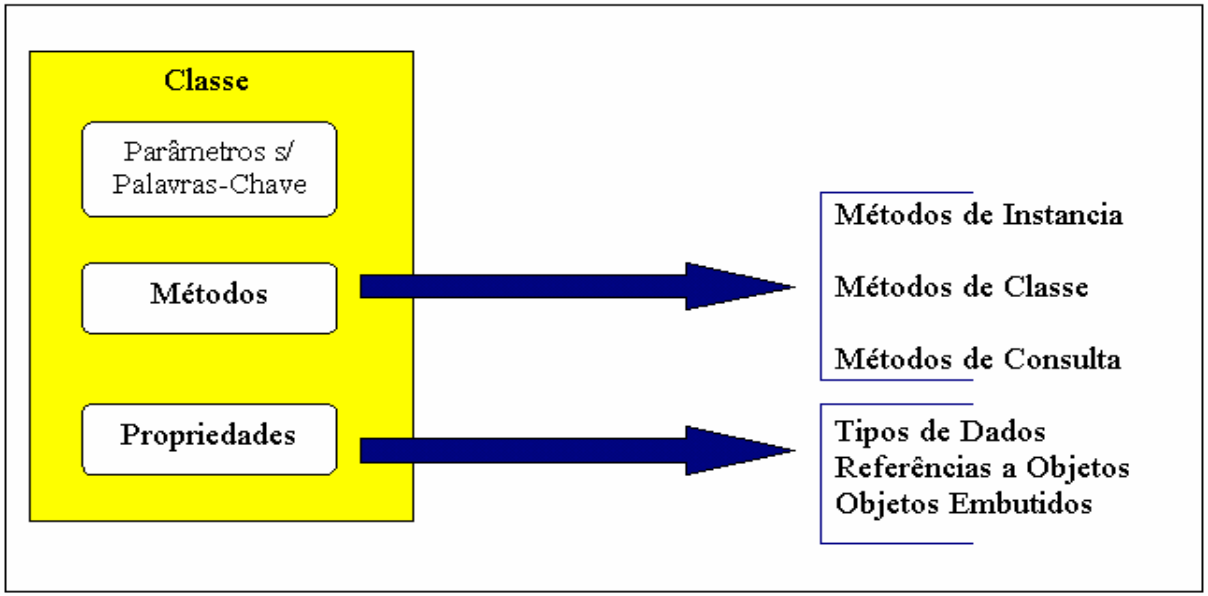
Dentro do *Caché Objects* as classes estão divididas em vários tipos (figura 5.5). A divisão básica existente é a de Classes de Objetos (*Object Classes*) e Classes de Tipos de Dados (*Data Type Classes).*



Fonte: Intersystens (1998)
Figura 5.5: Os Tipos de Classes do Caché Objects

5.3.2 Classes no caché objects

As classes do *Caché Objects* são formadas por um conjunto de parâmetros/palavras-chave, métodos e propriedades (figura 5.6).



Fonte: Baehr Jr. (1999)

Figura 5.6: Conteúdo de uma Classe do *Caché Objects*

6. Desenvolvimento do Protótipo

6.1 Especificação

Para o desenvolvimento do protótipo, especificação e apresentação de um controle de reserva de vagas, utilizou-se a *Unified Modeling Language* (UML) composto de diversos diagramas, dos quais:

a) diagrama de caso de uso;
b) diagrama de classe;
c) diagrama de seqüência.

6.1.1 Diagrama de caso de uso

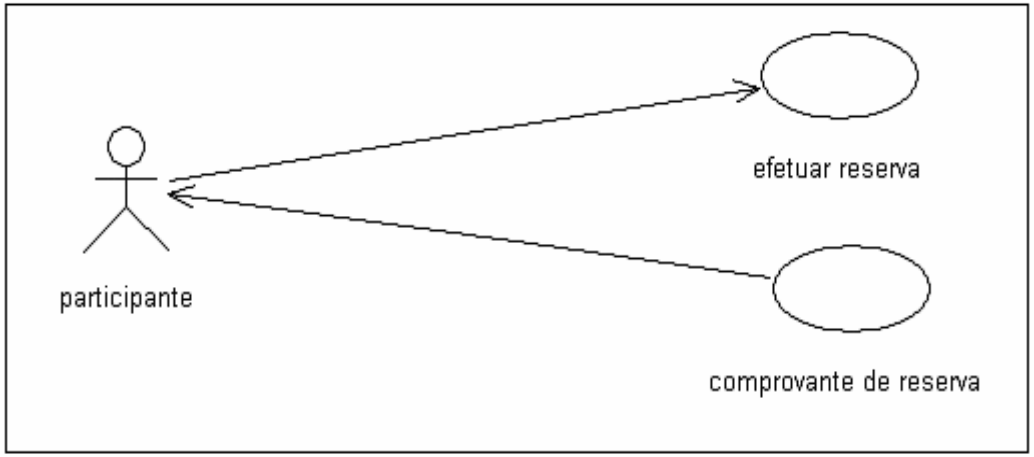


Figura 6.1 – Diagrama de caso de uso para o protótipo

6.1.2 Diagrama de classe

Na figura 6.2 está representado o diagrama de classe. Para representação deste diagrama foi utilizada a ferramenta *Rational Rose*.

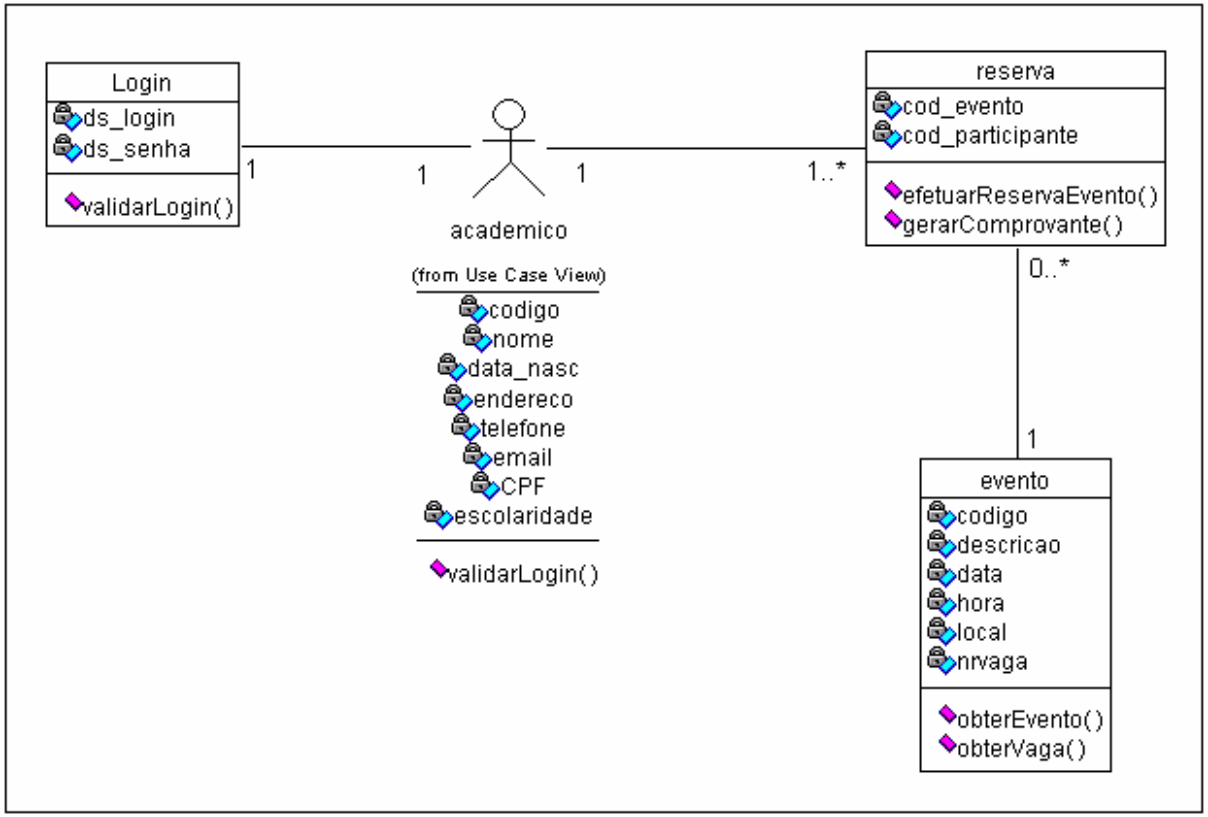


Figura 6.2 – Diagrama de classe para o protótipo

6.1.2 Diagrama de sequência

Através do diagrama de seqüência é representado os objetos e a troca de "mensagens", conforme Figura 6.3, que demonstra o caso de uso "Efetuar Reserva".

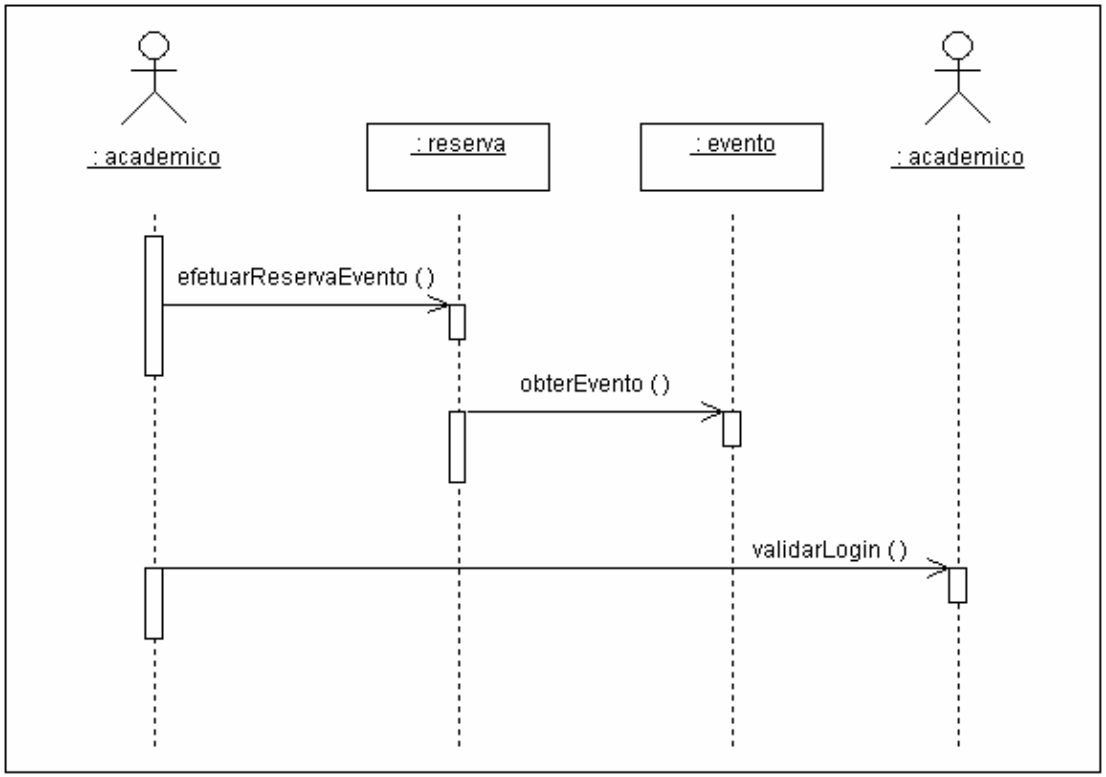


Figura 6.3 – Diagrama de seqüência – Efetuar reserva

O caso de uso “Emitir Comprovante” está demonstrado na Figura 6.4.

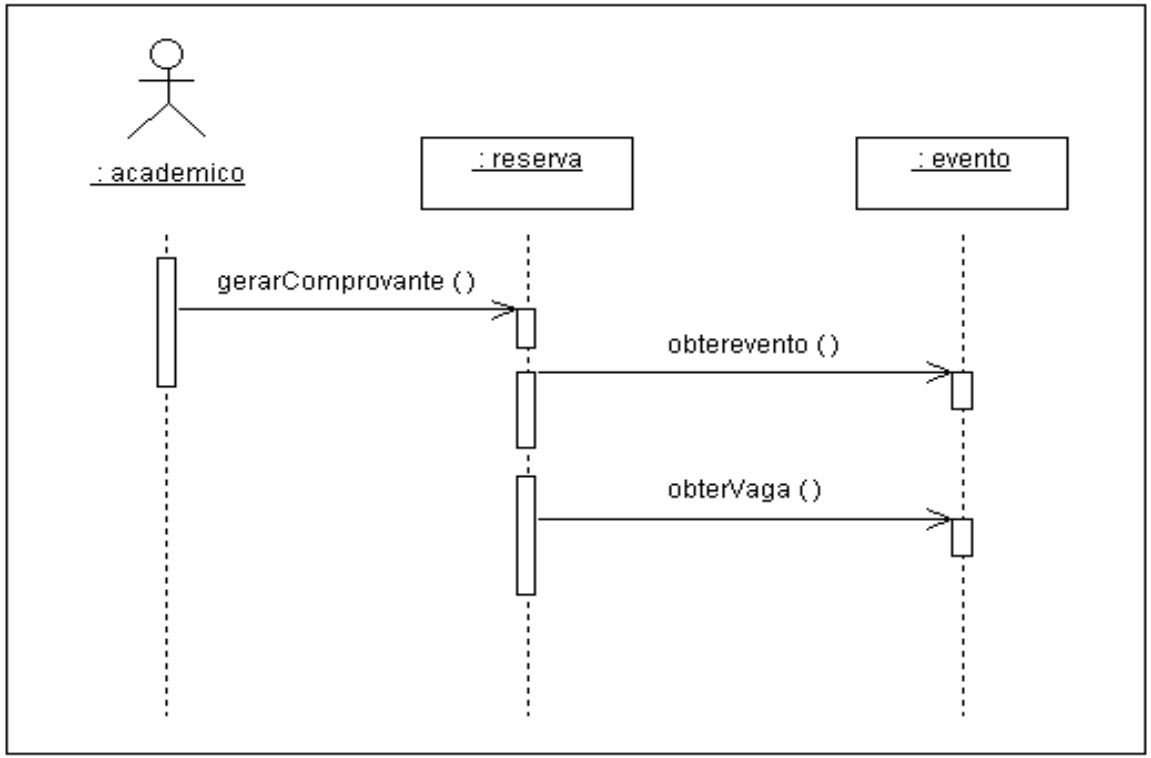


Figura 6.4 – Diagrama de seqüência – Emitir comprovante

6.2 Técnicas e ferramentas utilizadas

Este trabalho foi implementado para reproduzir uma das formas de acesso ao banco de dados *Caché*. A forma escolhida foi o acesso SQL via JDBC.

As ferramentas utilizadas para a programação do protótipo foram o Jbuilder 9.0 e o banco de dados *Caché*.

6.2.1 Borland Jbuilder 9

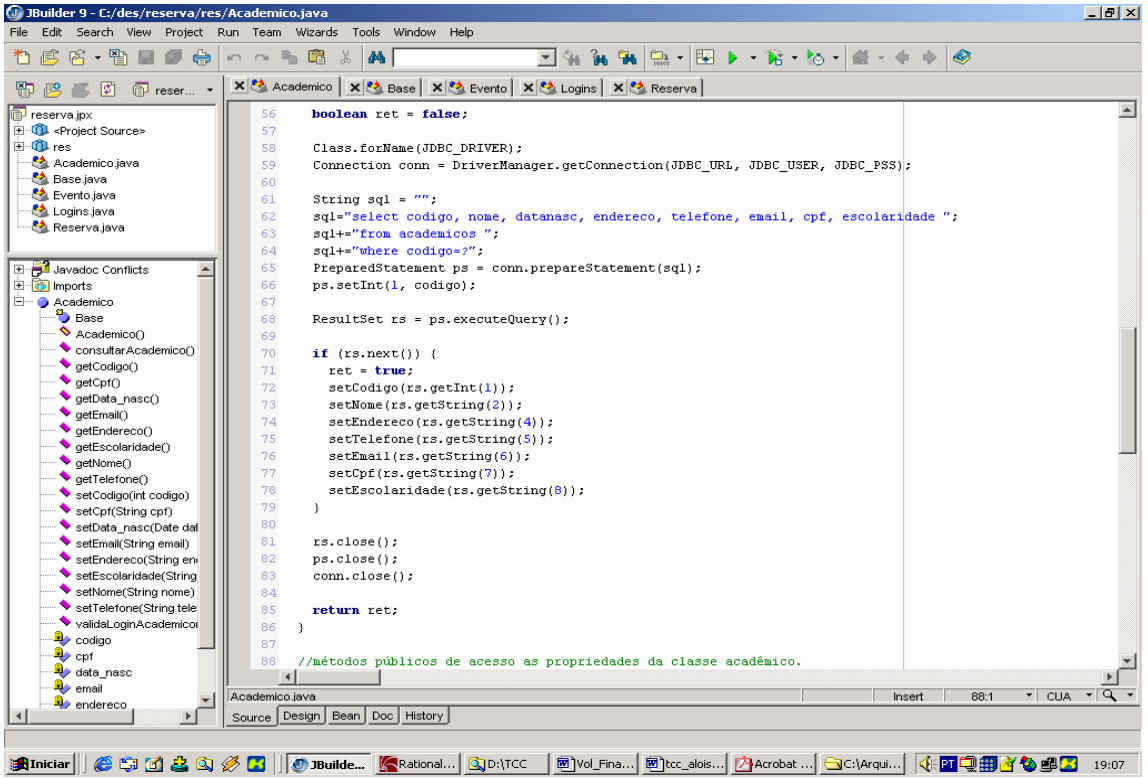


Figura 6.4 – Ambiente de desenvolvimento Jbuilder 9

6.2.2 Caché studio

O *Caché Studio* (figura 6.5) é um ambiente com interface GUI que permite a manipulação completa das classes do *Caché Objects.*

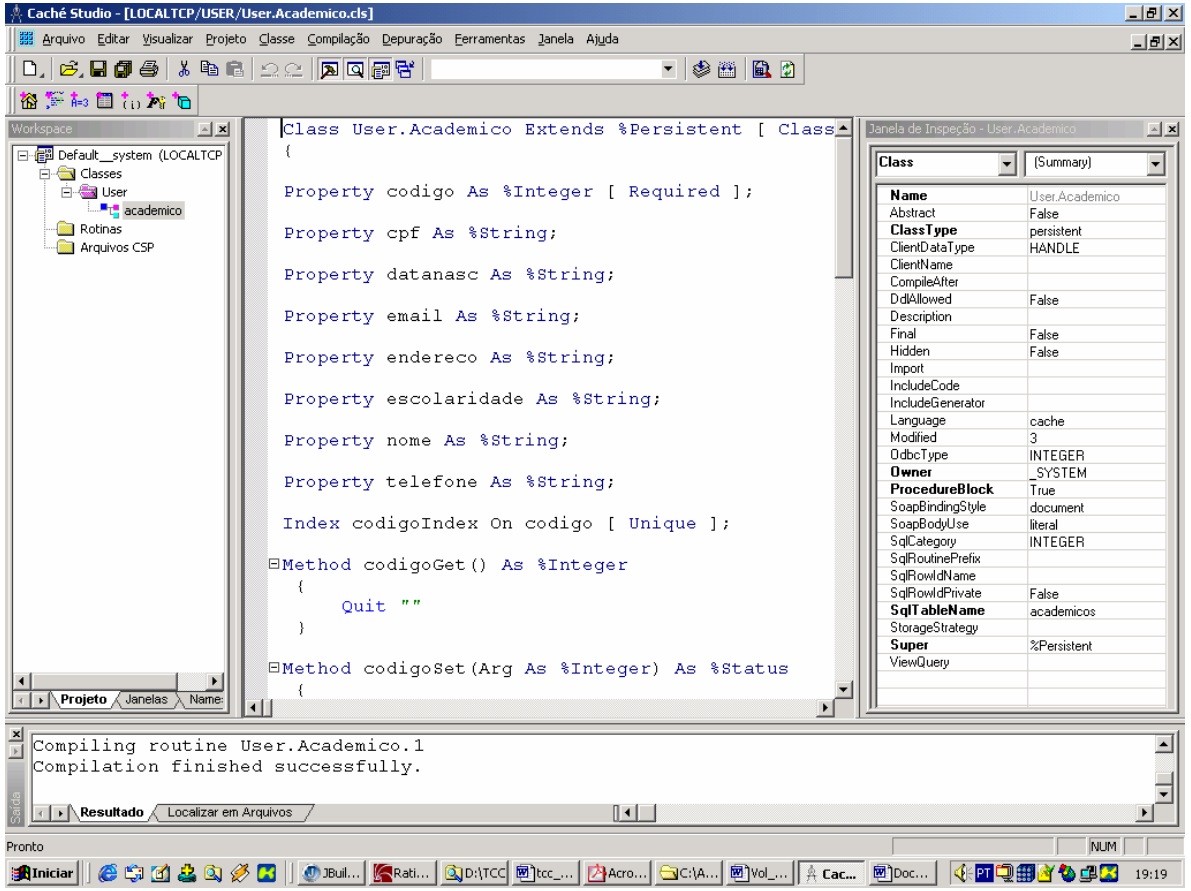


Figura 6.5 – O Caché Studio

6.3 Implementação

6.3.1 Integração web via JDBC com caché

O driver JDBC do servidor *Caché* está no arquivo CacheJDBC.jar. Este arquivo pode ser encontrado no sub-diretório /Java, dentro do diretório de instalação do *Caché*. Este arquivo deve estar na variável de ambiente CLASSPATH antes de compilar ou executar as aplicações.

```
set CLASSPATH=c:\cachesys\java\CacheJDBC.jar;%CLASSPATH%
```

A primeira coisa a fazer é carregar o driver JDBC a ser utilizado. O nome da classe a ser carregada é com.intersys.jdbc.CacheDriver. Caso contrário o *ClassLoader* vai disparar uma ClassNotFoundException quando o programa for executado, Intersystems (2003b).

```
//java code
try {
  Class.forName("com.intersys.jdbc.CacheDriver");
}
catch (ClassNotFoundException ex) {
  System.out.println(ex.getMessage());
}
```

Desenvolvimento do Protótipo

O segundo passo é estabelecer uma conexão com o servidor *Caché*. A URL abaixo endereça um servidor *Caché* na máquina local, respondendo pela porta *default*, 1972; e ainda aciona o *NameSpace* “User”, Intersystems (2003b).

```
"jdbc:Cache://127.0.0.1:1972/User"
```

O segmento de código abaixo mostra como estabelecer uma conexão com o servidor *Caché*:

```
//java code
String url = "jdbc:Cache://127.0.0.1:1972/User";
String user = "_SYSTEM";
String password= "SYS";

Connection conn = DriverManager.getConnection(url, user, password);
```

6.3.1 Operacionalidade do protótipo

O protótipo permitirá ao acadêmico:

a) Efetuar a reserva de vaga: o participante deverá efetuar sua reserva no evento escolhido, seja ele, seminário, feira, congresso, etc., escolhendo as opções no sistema;

b) Emitir comprovante de inscrição no evento: após confirmada a inscrição no evento o sistema emitirá um comprovante, que será utilizado pelo participante para garantir sua entrada no evento.

7. Conclusão

De forma geral, conseguiu-se atingir os objetivos propostos, através da implementação do protótipo e de estudos sobre o banco de dados utilizado, bem como realizando o teste na forma de acesso que o banco de dados permite.

Uma grande vantagem do modelo de objetos do *Caché* consiste que, ao se definir um dicionário de Classes, o sistema *Caché* gera implicitamente um dicionário de dados relacional na forma de tabelas. Dessa forma, de um único processo de definição de dados obtêm-se ambos os modelos de dados, relacional e orientado a objeto. As mesmas estruturas de dados podem ser acessadas por ambos os modelos, concorrentemente, sem problemas.

Uma limitação do protótipo é funcionar somente com banco de dados cujos fabricantes disponibilizem o driver JDBC, uma vez que a forma de acesso é via JDBC. Outra limitação do sistema é o fato de somente acadêmicos poderem participar dos eventos, pois deverão estar devidamente cadastrados no sistema.

Um outro problema encontrado foi com a versão do banco de dados *Caché* 4.1.6, que não funcionou com o acesso via JDBC, mesmo seguindo as instruções do tutorial. Sendo assim, a versão utilizada para o protótipo foi a 5.0.4, que atendeu perfeitamente ao objetivo deste trabalho.

7.1 Sugestões para trabalhos futuros

a) desenvolvimento de sistemas complexos para averiguar a performance do banco de dados, já que o protótipo em si não conseguiu comprovar esta característica;

b) utilizar as outras formas de conexões com Java, seja projetando as classes do *Caché* como classes Java ou como *Enterprise JavaBeans* (EJB);

c) a implementação do mesmo protótipo em outro tipo de acesso que o banco de dados permita, como por exemplo, utilizando a ferramenta *WebLink* do *Caché*;

d) a implemantação de outros conceitos (reserva de vagas em outra áreas que não a acadêmica).

Referências Bibliográficas

BAEHR Jr., Ivo. **Protótipo de sistema para gerenciamento de ordens de serviço acessando um banco de dados orientado a objetos e um banco de dados relacional**. Blumenau, 1999. 109f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, 1999.

DATE, C.J. **Introdução a sistemas de banco de dados**. Rio de Janeiro: Campus, 1991.

INTERSYSTEMS CORPORATION INC. **User´s Guide**. Cambridge: InterSystems Corporation, 1998.

INTERSYSTEMS CORPORATION INC. **Uma nova era na tecnologia dos bancos de dados**. São Paulo: InterSystems Corporation, 2002. Disponível em: < http://www.intersystems.com.br/downloads/ WhitepaperUmanovaeradeBD.pdf>. Acesso em: 18 jan. 2002.

KERN, Vinícius. **Banco de dados relacionais**. São Paulo: Érica, 1994.

KHOSHAFIAN, Setrag. **Banco de dados orientado a objeto**. Rio de Janeiro: Infobook, 1994.

KORTH, Henry F., SILBERSCHATZ, Abraham. **Sistema de banco de dados**, 2ªed. São Paulo: Makron Books, 1997.

PINHEIRO, Carlos André Reis. Fundamentos de bancos de dados orientados a objetos. **Developers cioMagazine**, São Paulo, v. 1, n. 44, p. 18-20, abril 2000.