

A collection of historical artifacts including a chessboard, medals, a compass, and glasses. The chessboard is in the top left, with several pieces visible. Below it are two medals with ribbons, one red and one blue. A pair of glasses with thin frames and a red-tipped stick are in the center. A compass is in the bottom left corner.

# Protótipo de software para geração de sistemas distribuídos utilizando Design Patterns

Aluno: Fabiano Oss  
fabiano@inf.furb.br

Orientador: Everaldo A Grahl  
egrahl@furb.br



# Agenda

- ◆ Introdução;
- ◆ Objetivos;
- ◆ Sistemas distribuídos;
- ◆ *Design Patterns*;
- ◆ XML;
- ◆ Desenvolvimento do trabalho;
- ◆ Conclusões.

# Introdução



- Já tive este caso antes, ou algo parecido?
- Sistemas distribuídos

# Objetivos

- ◆ Protótipo para geração de sistemas distribuídos;
- ◆ adptar alguns padrões para sistemas distribuídos;
- ◆ unir as tecnologias XML e CORBA





# Sistemas Distribuídos

# Introdução

- ◆ Evolução do desenvolvimento de sistemas;
  - monolíticos (anos 50 até 70);
  - cliente/servidor (anos 80 e 90);
  - sistemas distribuídos (momento atual);

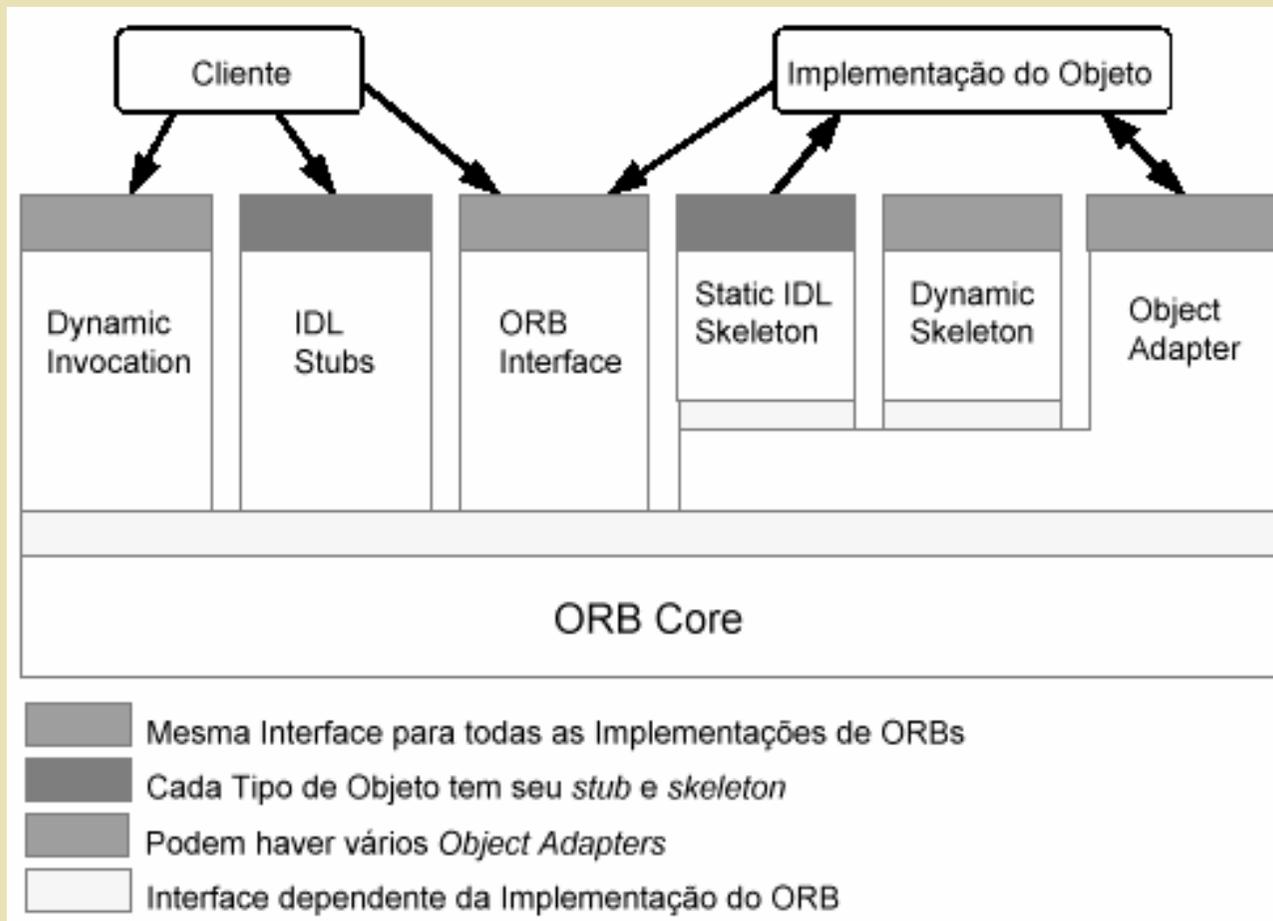




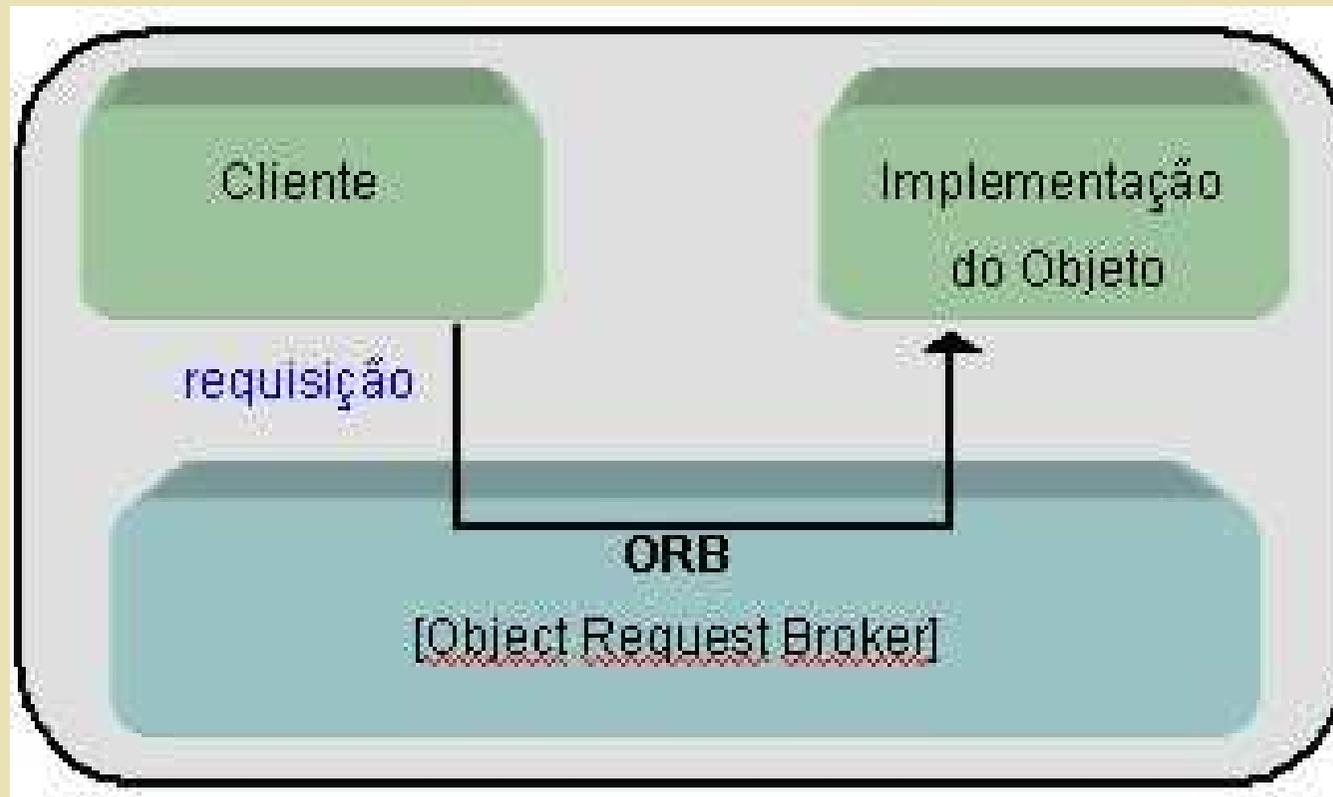
# Arquiteturas

- ◆ *Common Object Request Broker Architecture (CORBA);*
- ◆ *Java Remote Method Invocation and Specification (Java/RMI);*
- ◆ *Distributed Component Object Model (DCOM).*

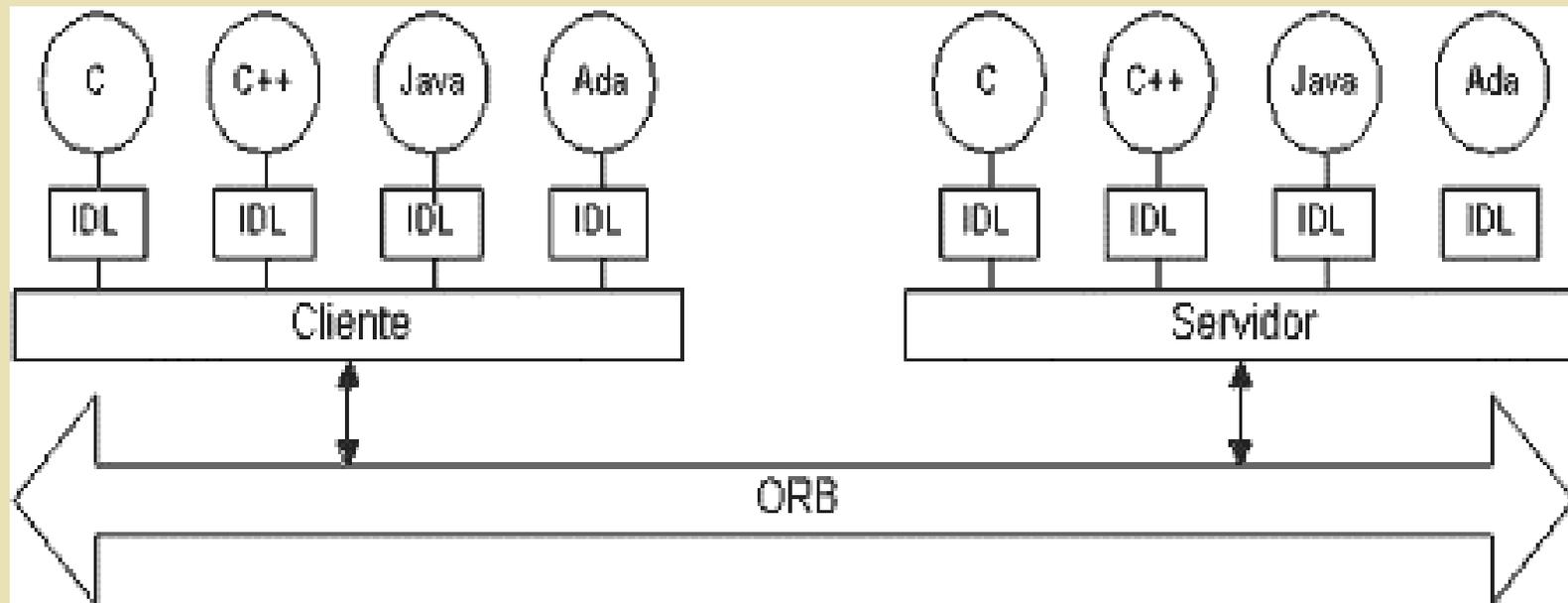
# CORBA



# Requisição



# IDL - Interface Definition Language



# Exemplo

```
interface cliente{
    /* atributos */
    attribute long codigo;
    attribute string nome;
    /*operacoes*/
    long operacao1(in string nome);
};

/* herença */
interface pessoaFisica:cliente{
    void operacao2(in long codigo, inout string nome);
};
```



## Visibroker + Delphi 6

- ◆ Implementação do CORBA 2.3;
- ◆ compilador IDL para Pascal (idl2pas);
- ◆ *units*: Orbpas45.pas e Corba.pas;
- ◆ *runtime*: orbpas40.dll, orb\_br.dll e vport.dll;



# Design Patterns

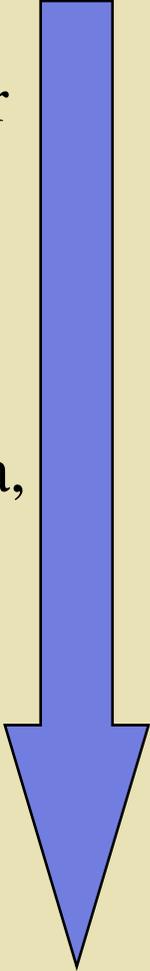


## Conceito

“Um pattern descreve um problema que ocorre várias vezes em nosso ambiente, assim como elabora uma solução reutilizável para ele, de modo que não necessite se preocupar novamente, com a solução do problema nas novas implementações”

(Gamma, 1995)

# Evolução

- 
- ◆ Christopher Alexander
  - ◆ Ward e Kent
  - ◆ Coad, Copplien, Gama,  
...
  - ◆ Atualmente...
- ◆ Conceito
  - ◆ aplicação para software
  - ◆ aplicação para software  
OO.
  - ◆ Conferências, novos  
conceitos, etc.



# Distributed Callback

- ◆ Problema;
- ◆ aplicabilidade;
- ◆ benefícios;
- ◆ consequências



# Como funciona?

```
interface foo_app {  
    string getInfo(in string my_name, inout string  
                  mood, out string joke);  
}
```

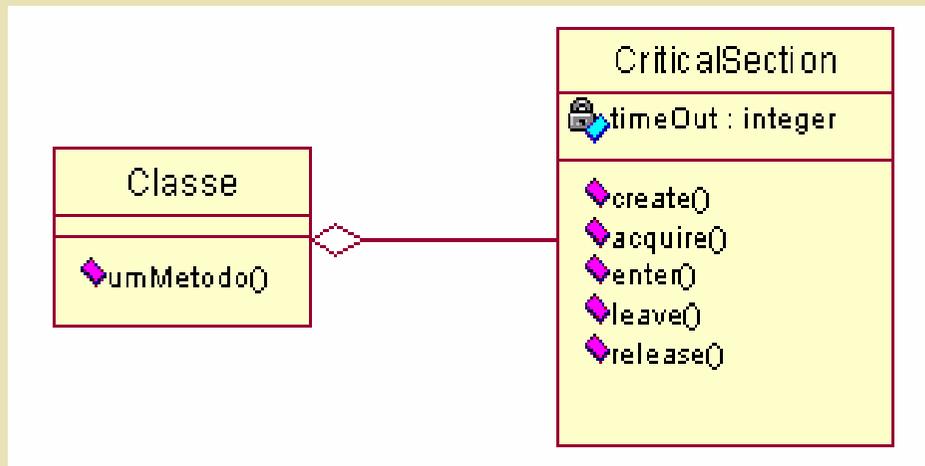




# Resultado

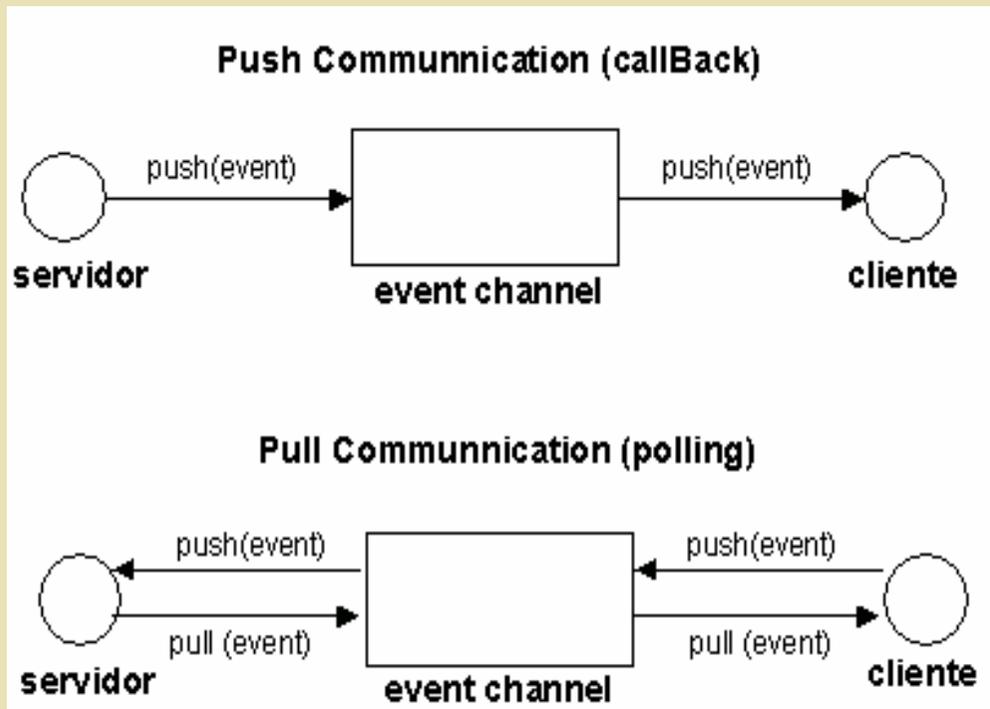
```
interface foo_app_cb;
interface foo_app {
    oneway void get_info(in string my_name,
                        in string mood, in foo_app_cb
                        my_callback);
};
interface foo_app_cb
{
    oneway void callback(in string mood, in string
                        joke,in string return_value,
                        in boolean successful);
};
```

# Lock



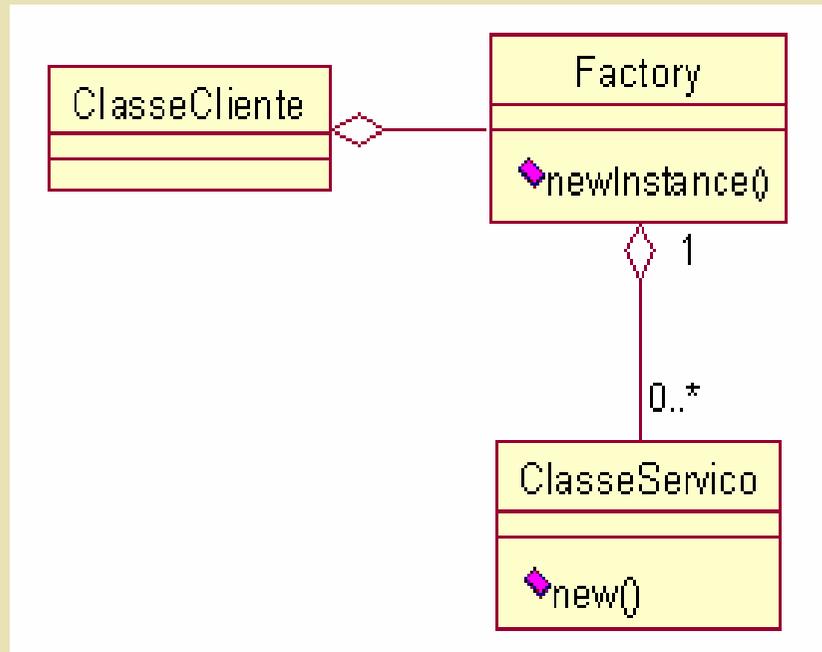
- ◆ Problema;
- ◆ aplicabilidade;
- ◆ resumo;
- ◆ benefícios;
- ◆ conseqüências

# Notificação (Event Service)



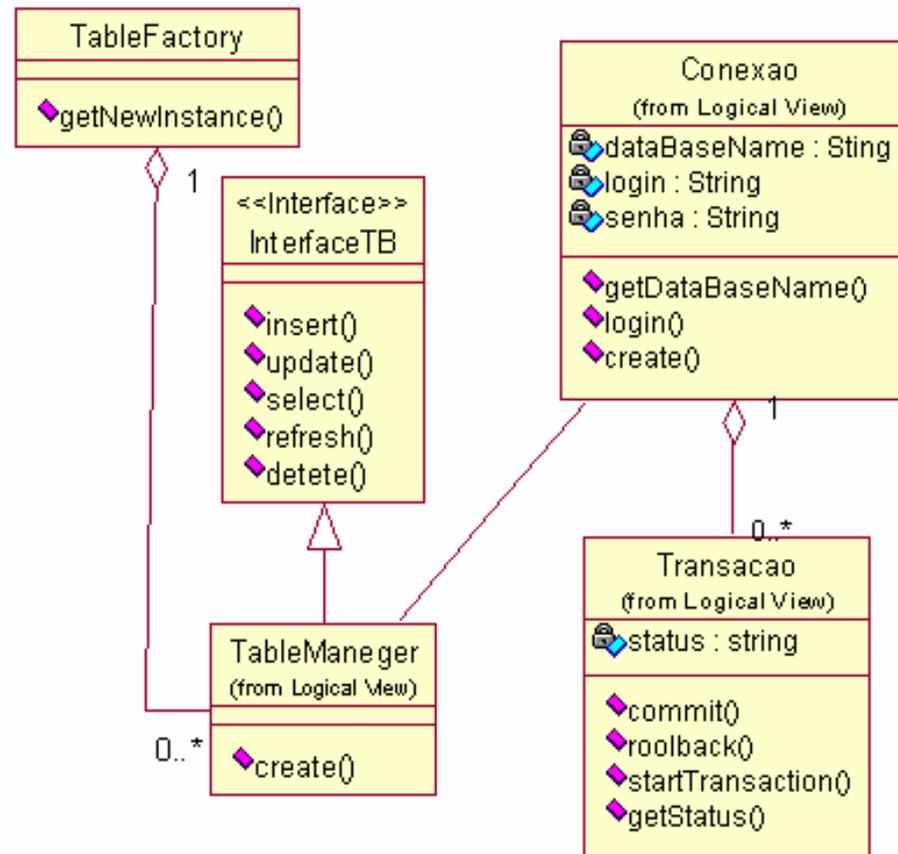
- ◆ Problema;
- ◆ aplicabilidade;
- ◆ resumo;
- ◆ benefícios;
- ◆ conseqüências

# Factory



- ◆ Problema;
- ◆ aplicabilidade;
- ◆ resumo;
- ◆ benefícios

# Persistent Layer



- ◆ Problema;
- ◆ aplicabilidade;
- ◆ resumo;
- ◆ benefícios;

# IDL Persistent Layer

```
typedef sequence<string> strArray;
exception ERR_CONEXAO {
    string descricaoErro;
    long codigoErro;
};

exception ERR_DB {
    string descricaoErro;
    long codigoErro;
};

interface conexao {
    attribute string dataBaseName;
    boolean login(in string driver, in string login, in string senha) raises
(ERR_CONEXAO);
    boolean desconecta();
    Object getConexao();
};
interface tableManager{
    void setConexao(in conexao con);
    string select(in string stm);
    string execStorageProcedure(in string nome, in strArray parametros) raises
(ERR_DB);
    long insert(in strArray colName, in strArray colValue, in string tableName)
raises (ERR_DB);
    long insertDB(in string stm) raises (ERR_DB);
    long update(in strArray colName, in strArray colValue, in string tableName, in
strArray where) raises (ERR_DB);
    long updateDB(in string stm) raises (ERR_DB);
    long delete(in strArray where, in string tableName) raises (ERR_DB);
    long deleteDB(in string stm) raises (ERR_DB);
    string refresh(in string stm) raises (ERR_DB);
};
```

# Template

```
object #classeForm: T#classeForm
OldCreateOrder = False
PageProducer = AdapterPageProducer
  Left = 297
  Top = 188
  Height = 150
  Width = 215
```

- ◆ Problema;
- ◆ aplicabilidade;
- ◆ resumo;
- ◆ benefícios;





# Extensive Markup Language (XML)



# XML

- ◆ inteligência;
- ◆ adaptação;
- ◆ manutenção;
- ◆ troca de informações;
- ◆ auto-descritivos;
- ◆ portabilidade, etc.

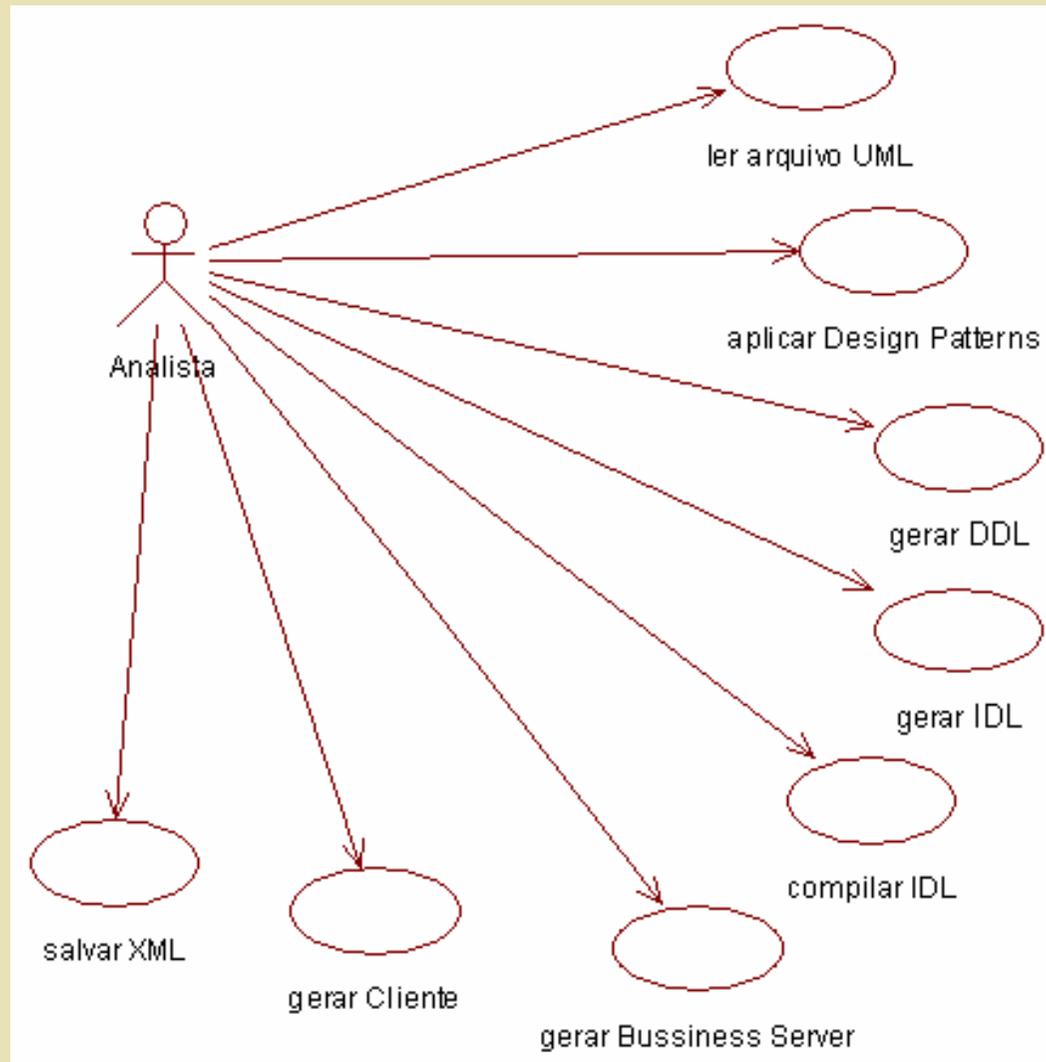
# Exemplo...

```
<?xml version="1.0" ?>
<!DOCTYPE dic.dtd (View Source for full doctype...)>
- <ListaDicionarios>
- <Dicionario Nome="Formatura.mdl">
+ <Classe Nome="Pessoa">
+ <Classe Nome="Financeiro">
- <Classe Nome="Contas">
  <Documentacao />
  <Descricao />
  <Visibilidade>public</Visibilidade>
- <Metodo Nome="getTotalConta">
  <Visibilidade>Public</Visibilidade>
  <Result>float</Result>
- <ListaParametros>
- <Atributo Nome="conta">
  <Tipo>integer</Tipo>
  </Atributo>
  </ListaParametros>
  </Metodo>
- <Atributo Nome="codigo">
  <Visibilidade>private</Visibilidade>
  <Tipo>integer</Tipo>
  </Atributo>
- <Atributo Nome="nome">
  <Visibilidade>private</Visibilidade>
  <Tipo>string</Tipo>
  </Atributo>
  </Classe>
+ <Classe Nome="Aluno">
+ <Classe Nome="Fornecedor">
+ <Classe Nome="Cidade">
+ <Classe Nome="MathFinanceiro">
+ <Associacao Nome="movimentacao">
+ <Associacao Nome="pertence">
+ <Associacao Nome="referente">
+ <Associacao Nome="reside">
+ <Associacao Nome="calcula">
  <Descricao>obs</Descricao>
  </Dicionario>
</ListaDicionarios>
```

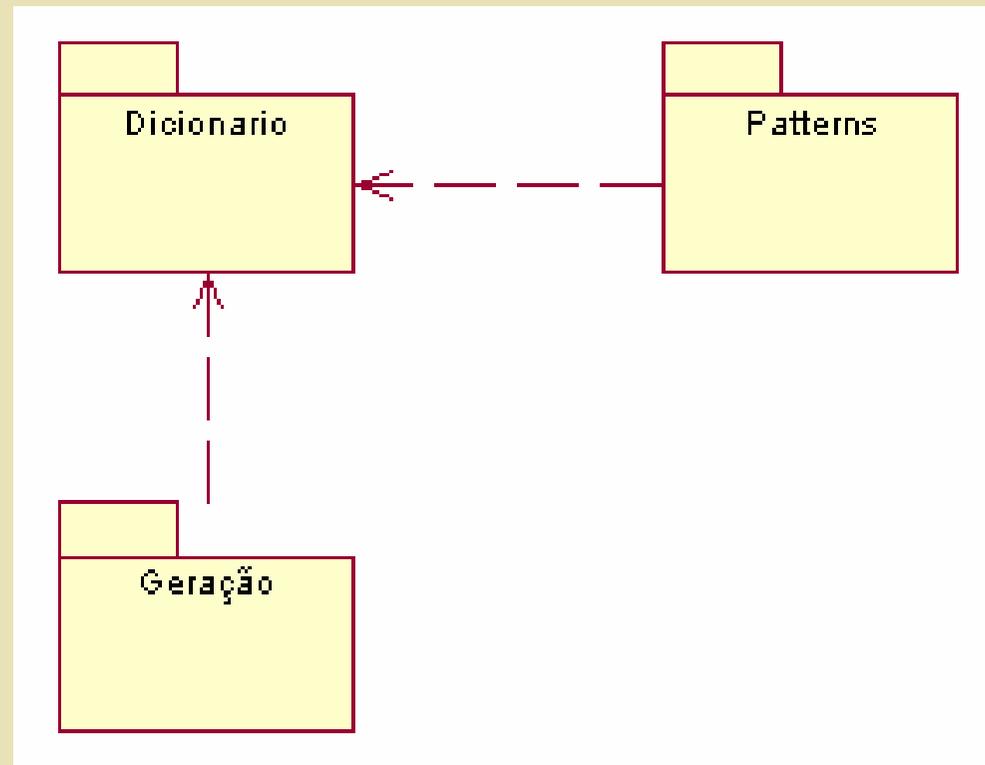


# Desenvolvimento

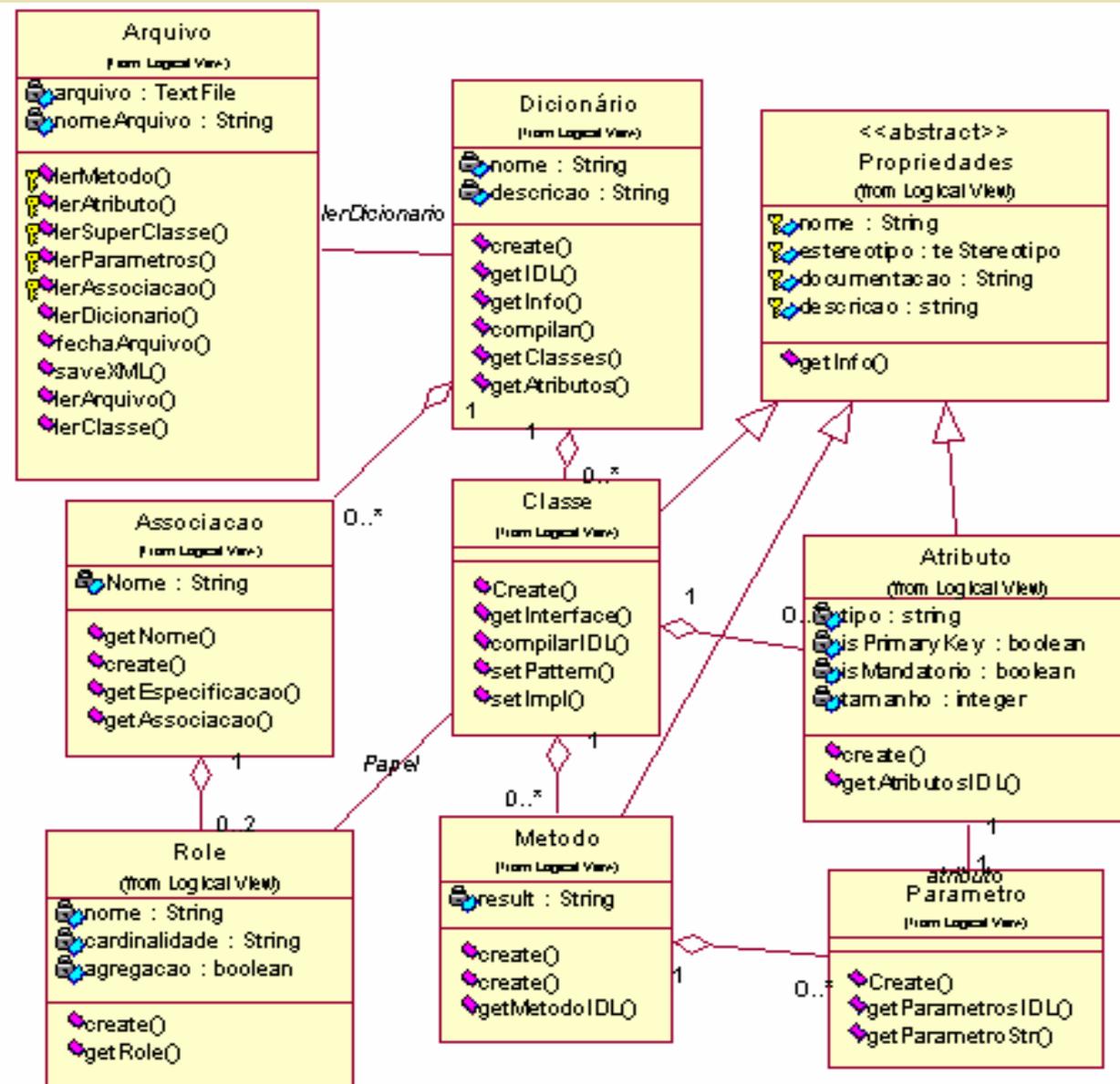
# Use Case



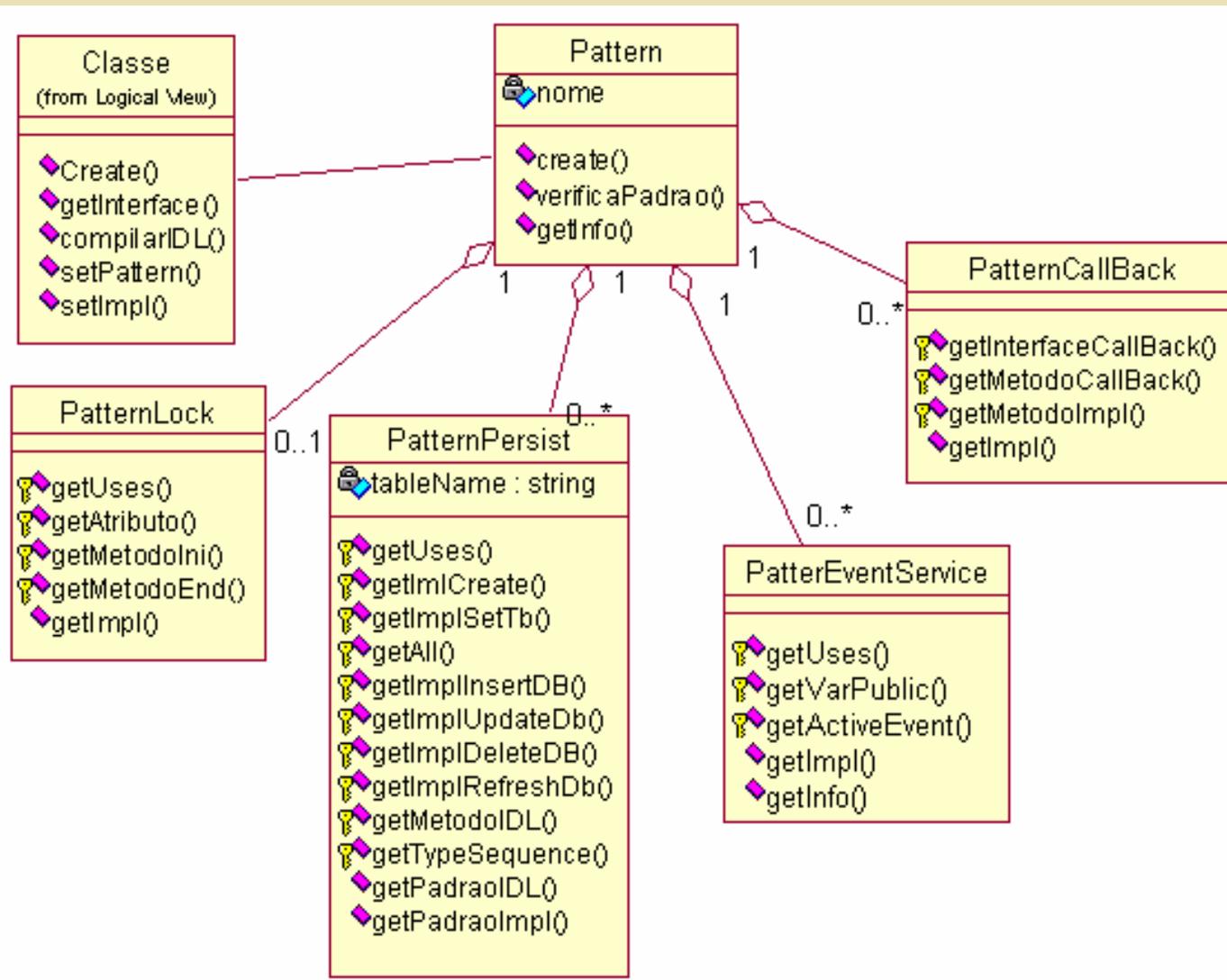
# Package



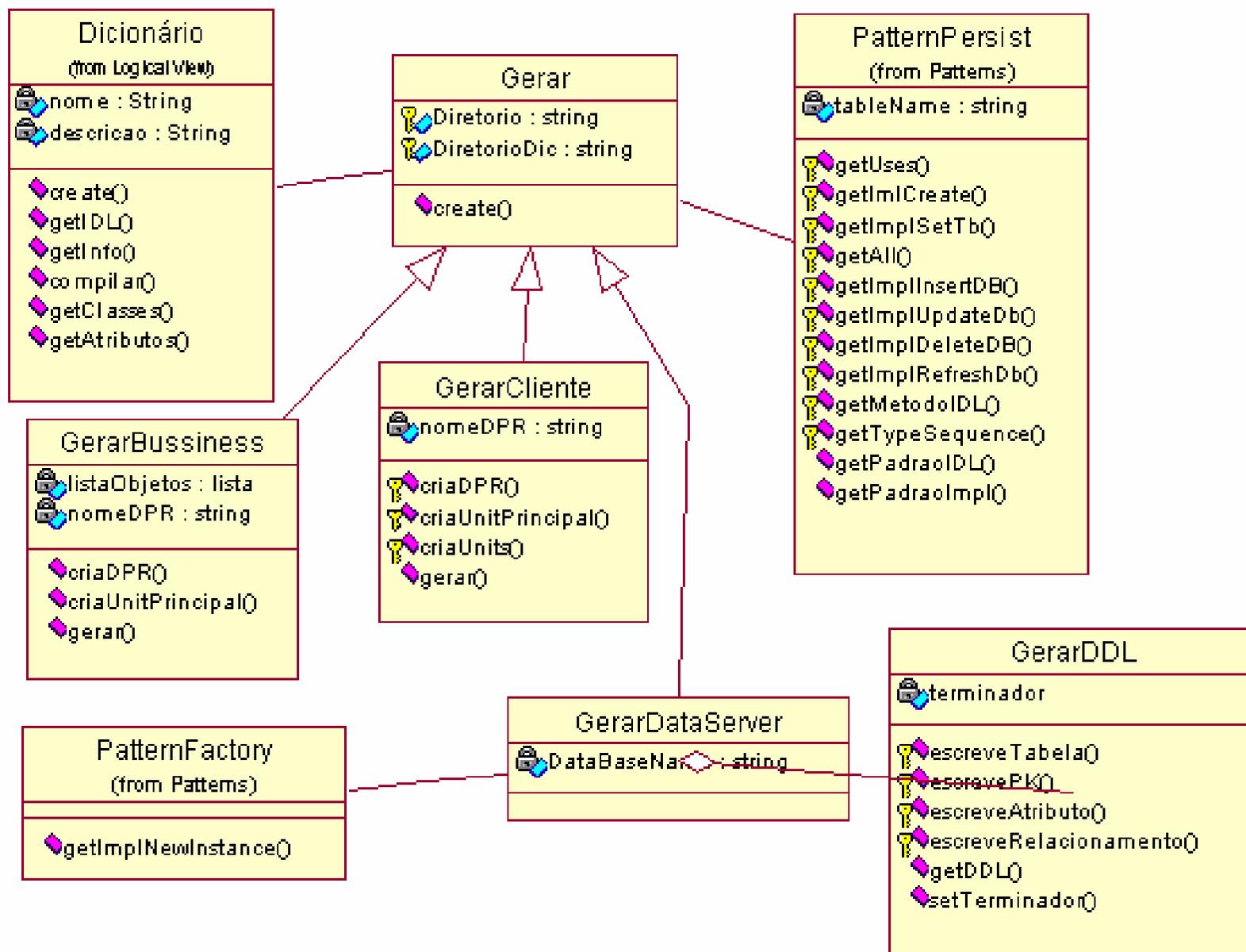
# Package Dicionário



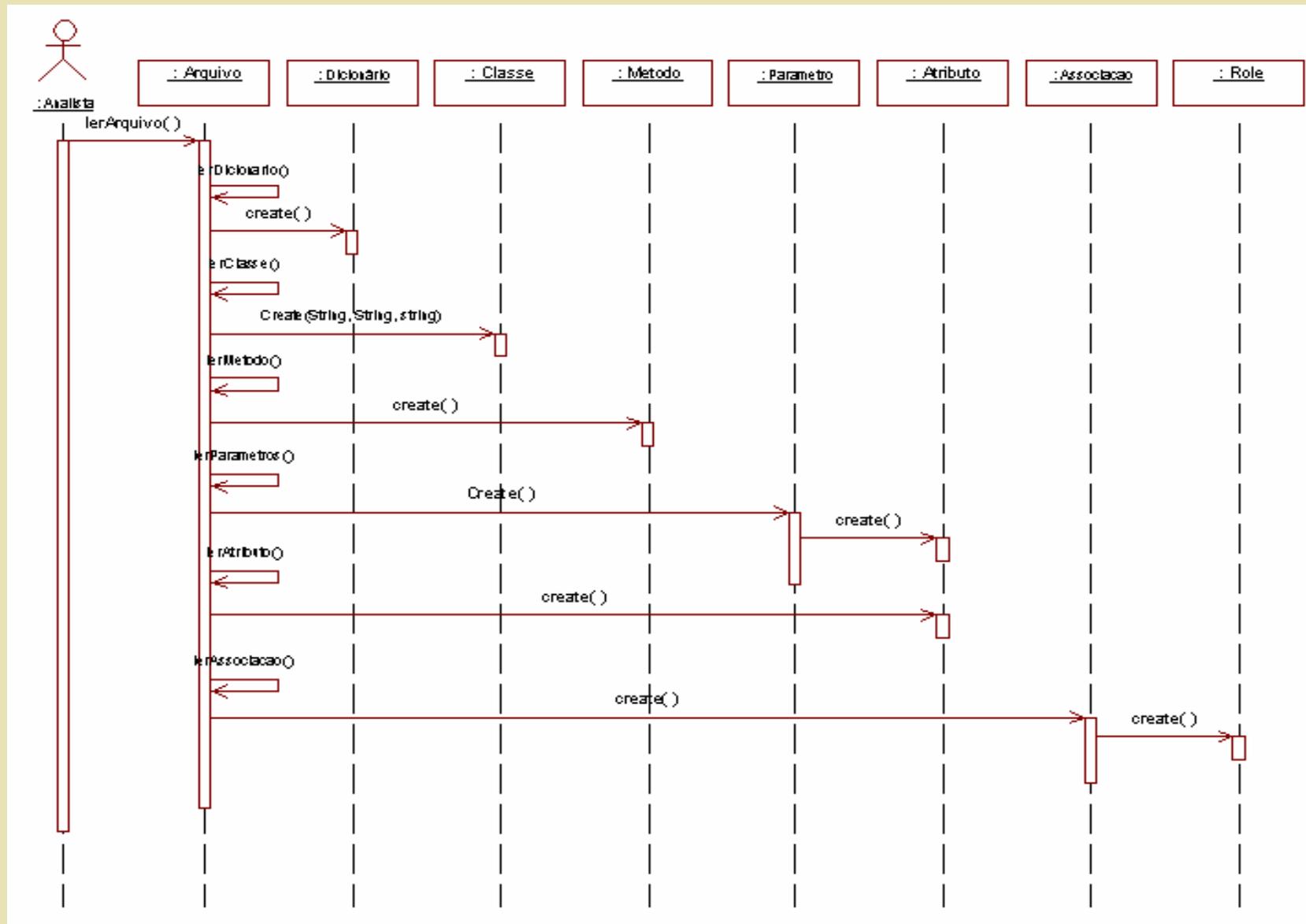
# Package Patterns



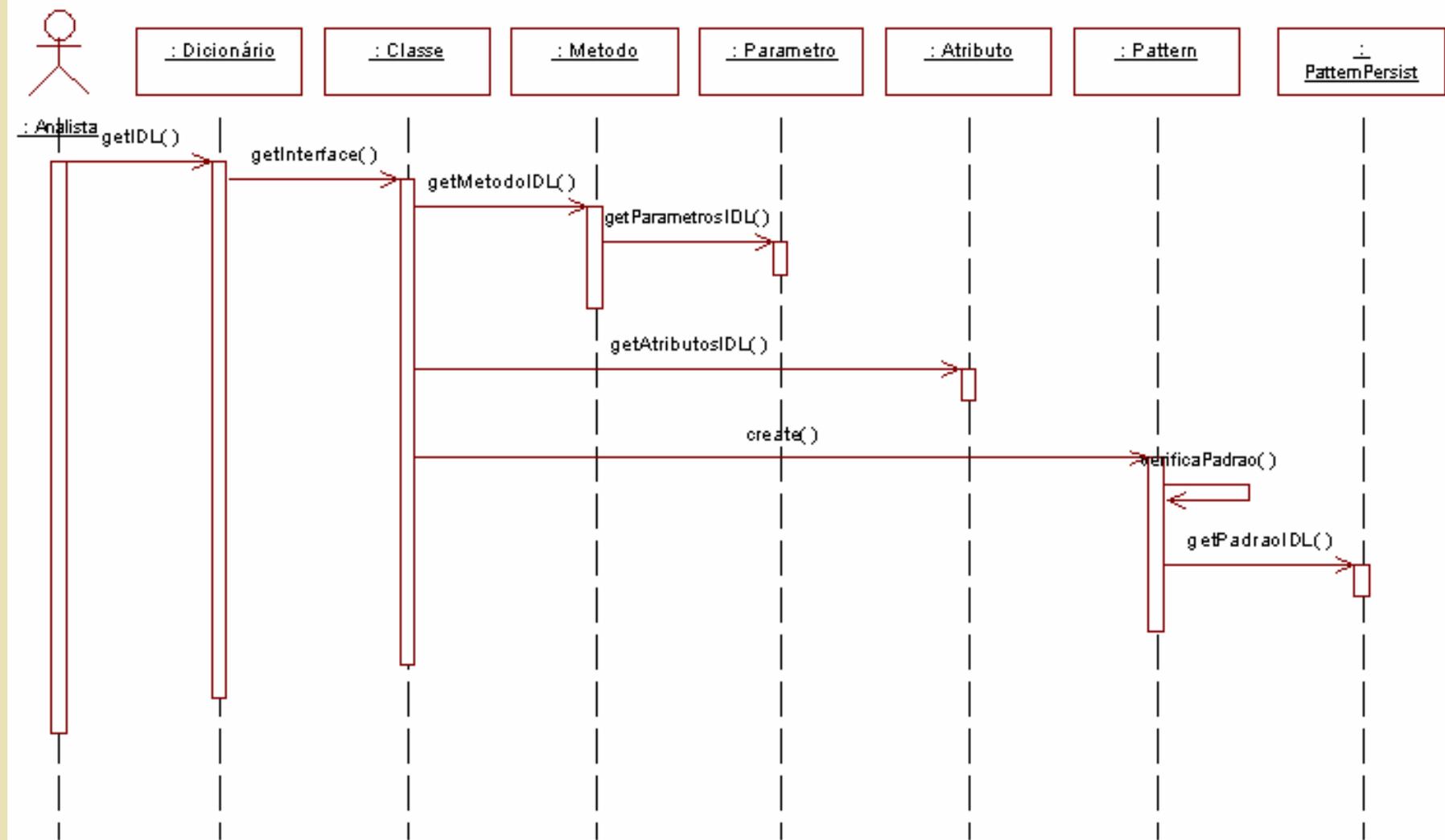
# Package Geração



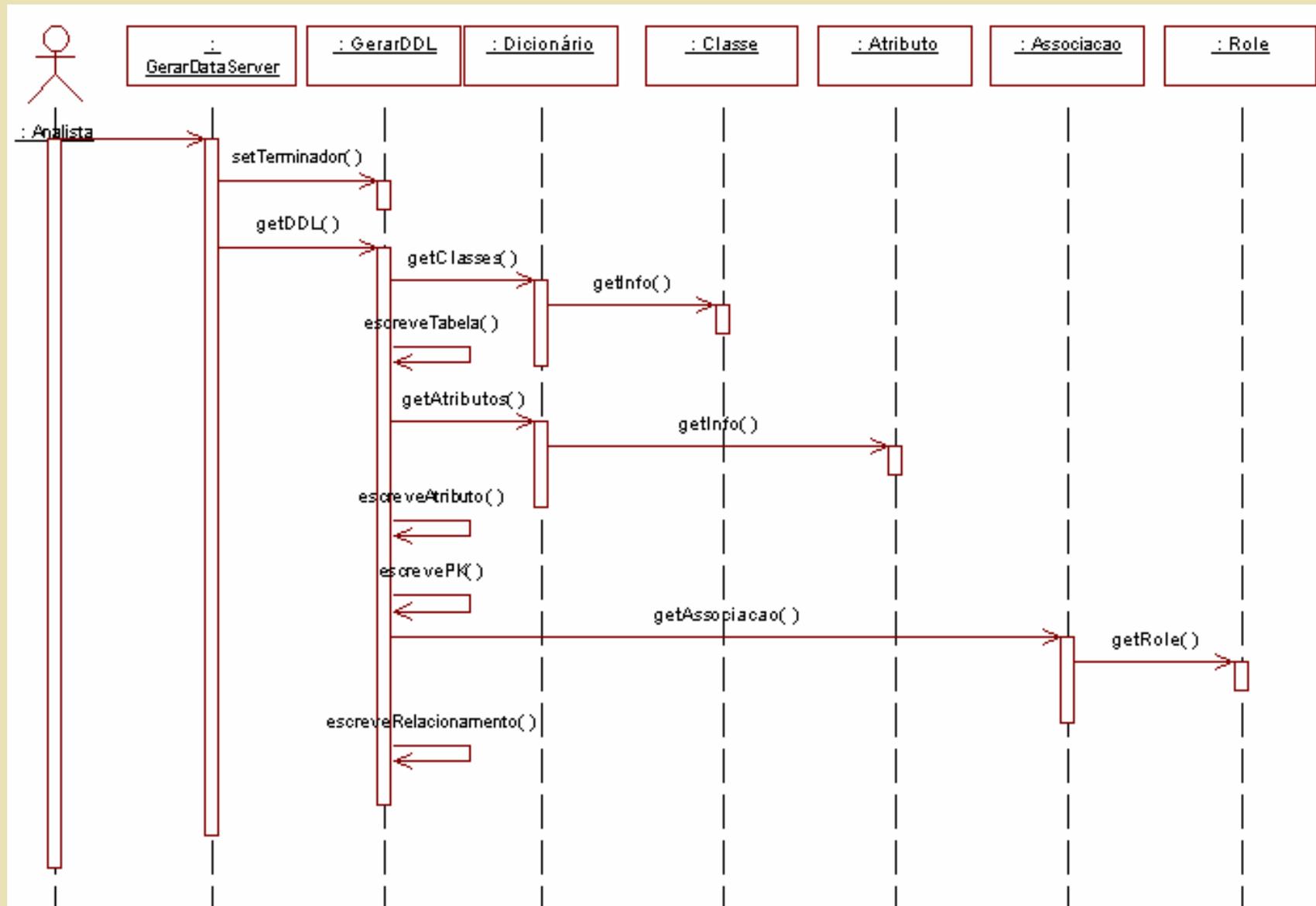
# Diagrama Seqüência Ler Arquivo UML



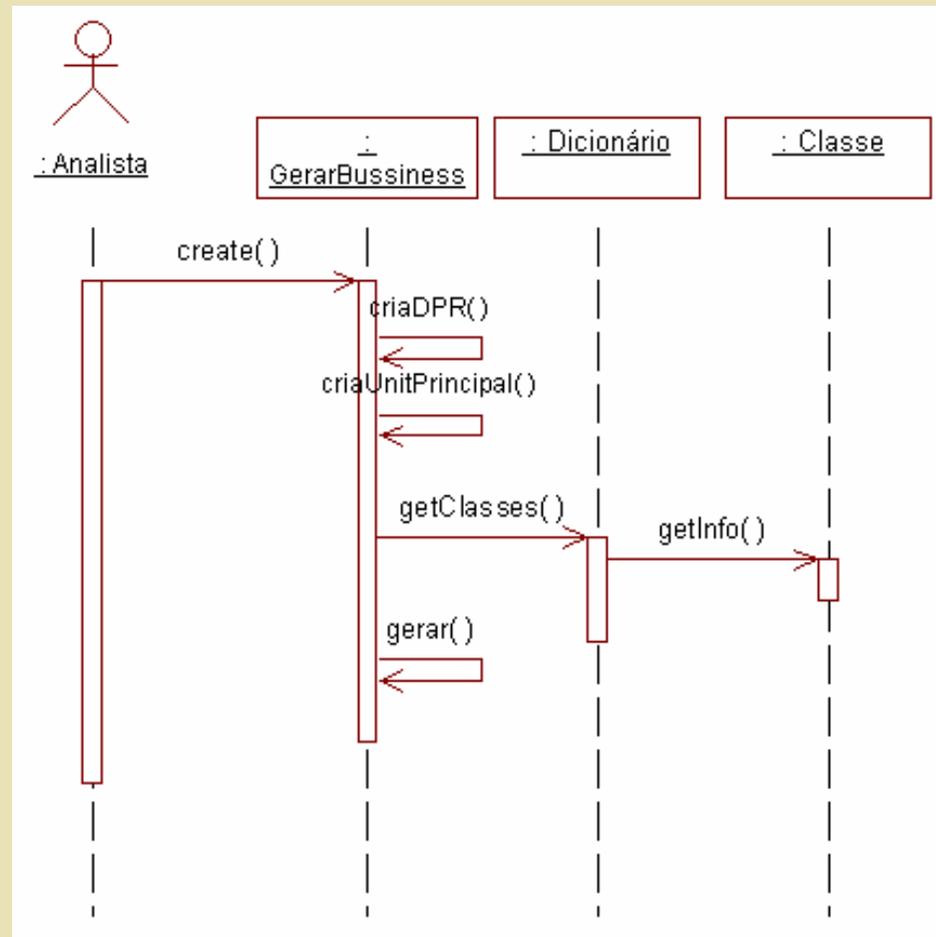
# Diagrama Seqüência Gerar IDL



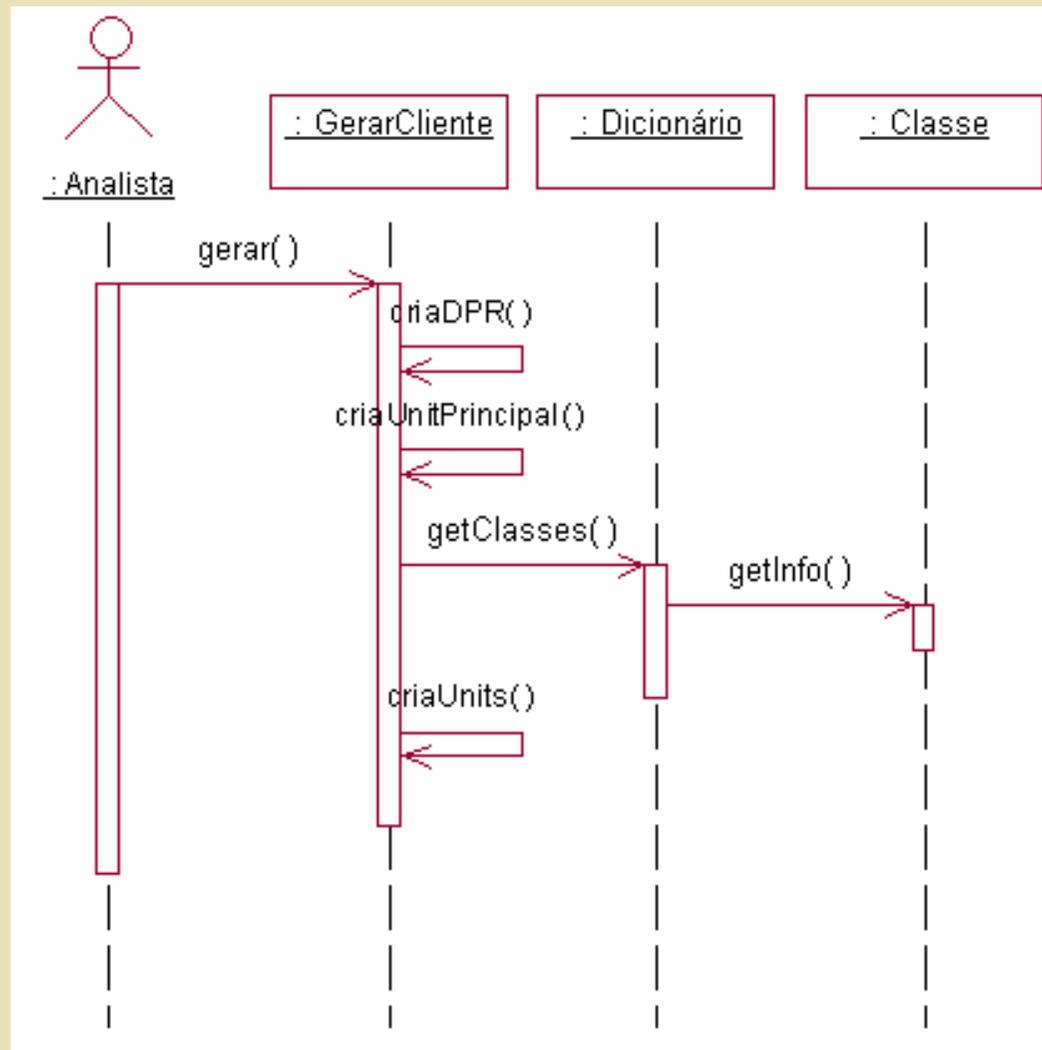
# Diagrama Seqüência Gerar DDL



# Diagrama Seqüência Gerar Business Server



# Diagrama Seqüência Gerar Cliente



# Protótipo

The screenshot shows a software application window titled "Dicionario" with a menu bar (Arquivo, Configuracoes, Ajuda) and a toolbar. The main interface is divided into two panes: "Dicionário" on the left and "Classe" on the right.

The "Dicionário" pane has tabs for "Cliente", "Negocio", and "Dados". Under "Negocio", a tree view shows a folder "Formatura.mdl" containing several classes: Pessoa, Financeiro, Contas, Aluno, Fornecedor, Cidade, MathFinanceiro, movimentacao, pertence, referente, reside, and calculo.

The "Classe" pane has tabs for "Descricao", "Metodo", "Atributos", "IDL", and "Implementacao". The "Metodo" tab is active, showing the "Especificacao" for a method:

- Nome:
- Parametros:
- Retorno:
- Visibilidade:
- Pattern:

A dropdown menu is open for the "Pattern" field, showing the following options:

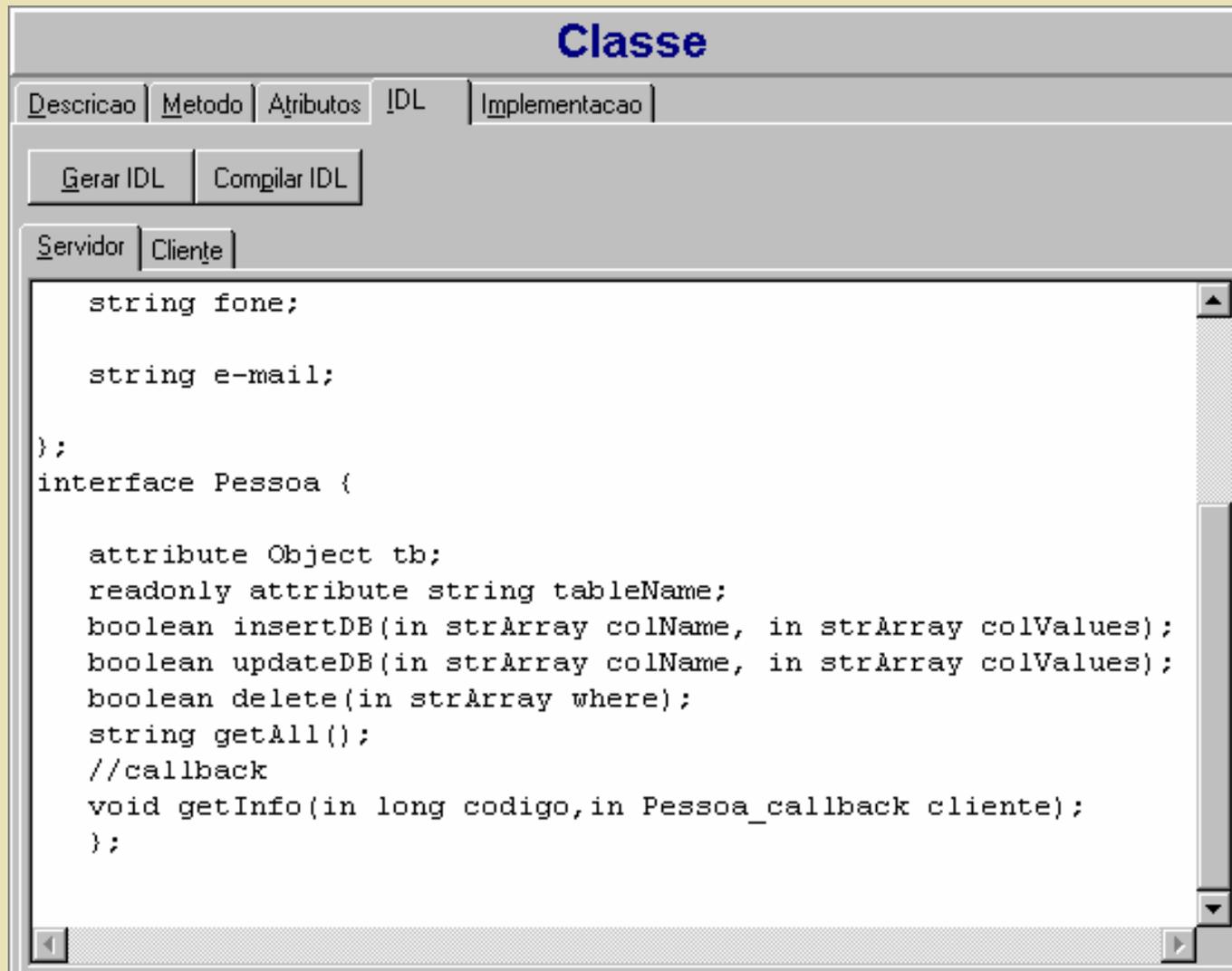
- lock
- distributed callback
- lock
- event service

Below the dropdown is a table titled "Lista" with the following columns: Nome, Visibilidade, Parametros, Retorno, and Pattern.

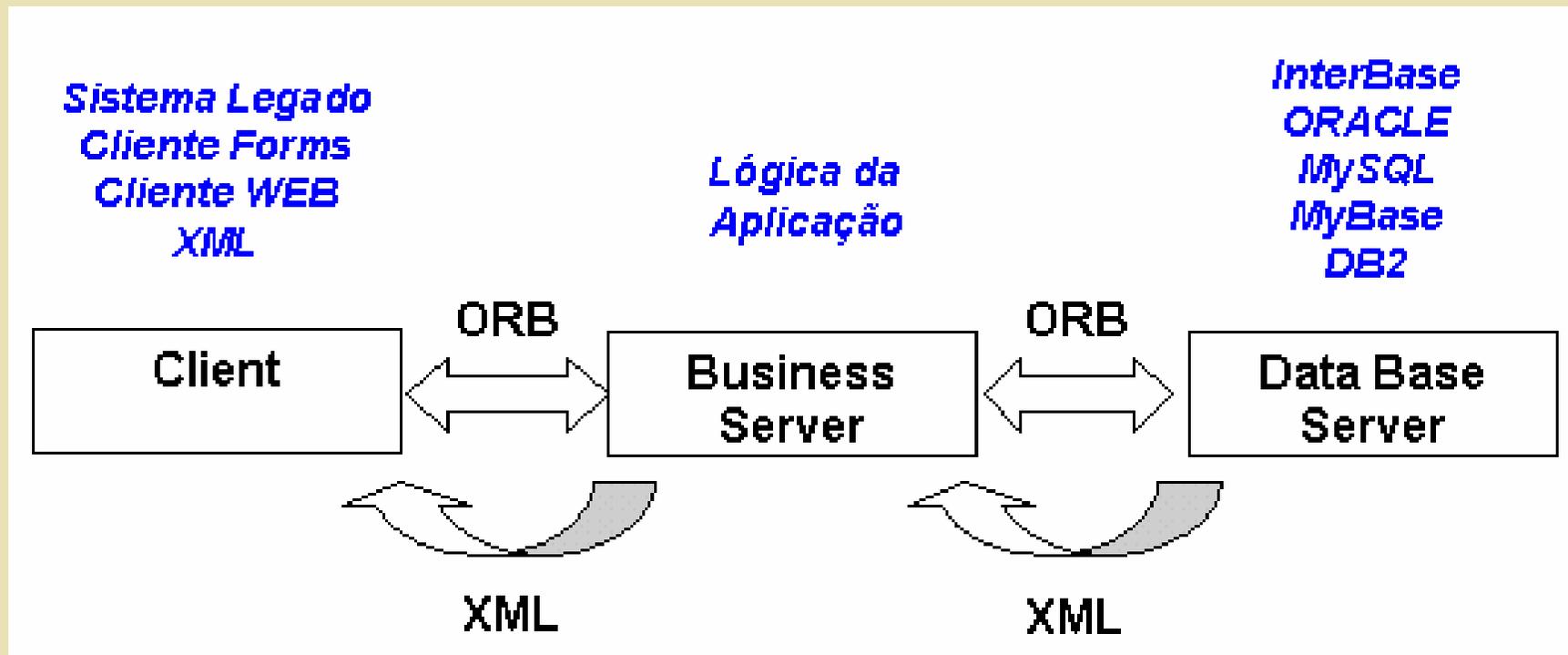
Nome	Visibilidade	Parametros	Retorno	Pattern
getValorCorrigido	Public	(juro:float,...	float	
getDebito	Public	()	any	
getAVencer	Public	(data:sttrin...	string	

A callout box with the text "Escolhendo o Design Pattern" has a line pointing to the "lock" option in the dropdown menu.

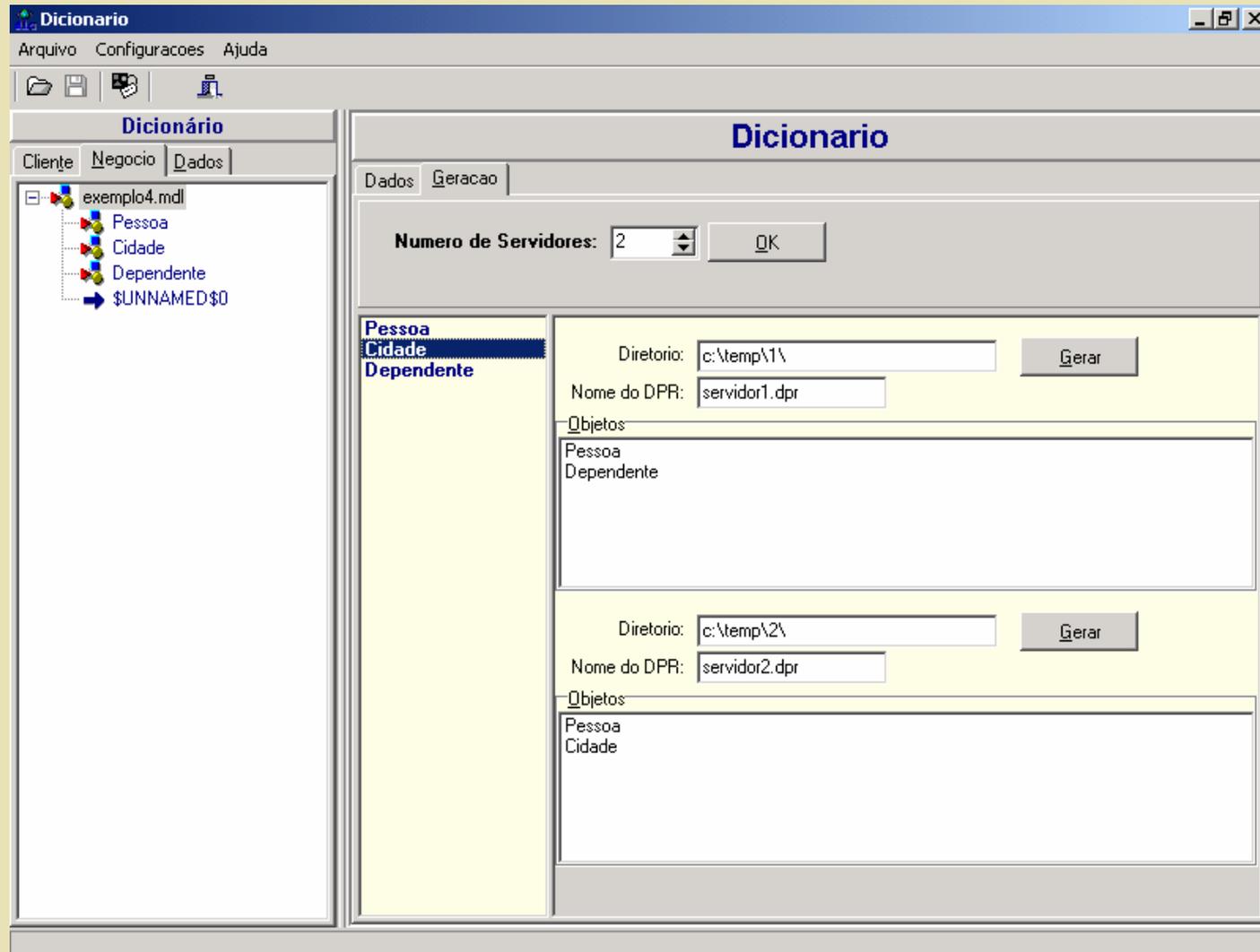
# Gerar IDL



# Arquitetura



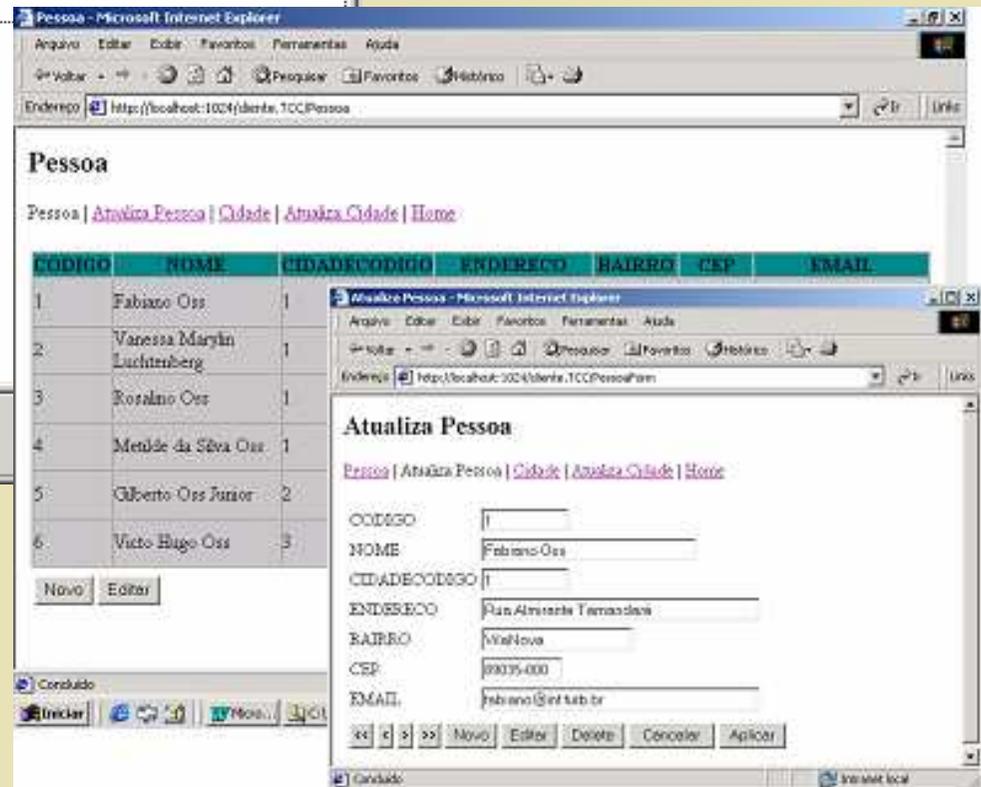
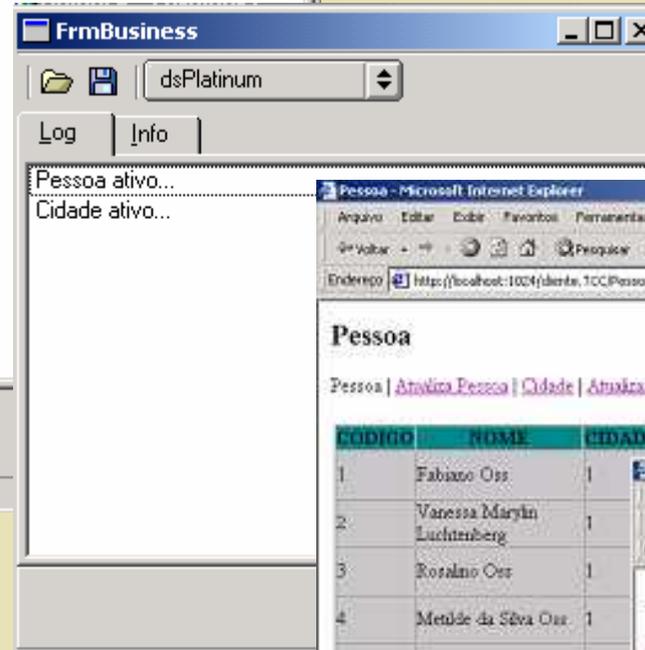
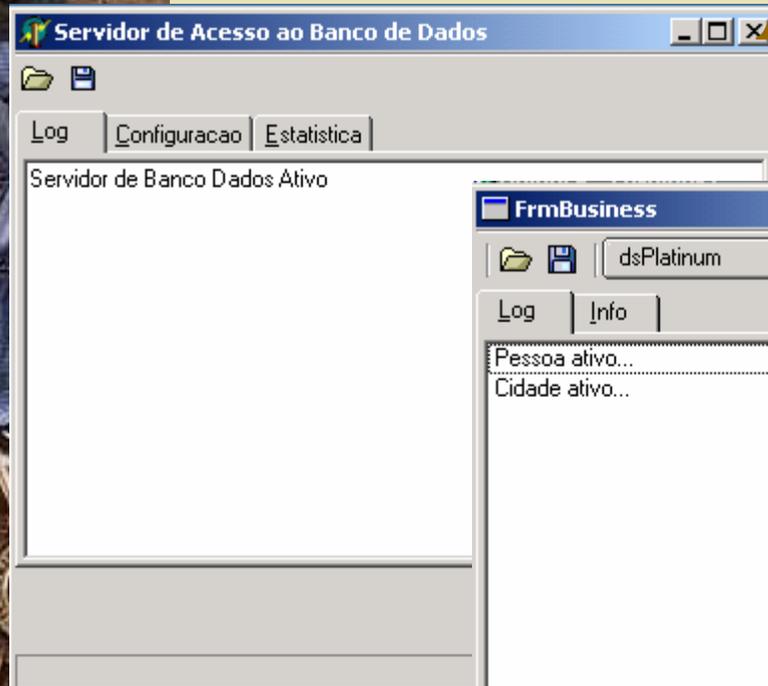
# Gerar Business Server



# Gerar Cliente



# Aplicação



# Conclusão

- ◆ Sistemas distribuídos;
- ◆ *Design Patterns*;
- ◆ *Padrões utilizados*;
- ◆ *Arquitetura*;
- ◆ *Ferramentas utilizadas*





# Demonstração



Perguntas ???



*Fim!*