

Framework para Mapeamento Objeto- Relacional em Delphi

Acadêmico: Samuel Y. Deschamps

Orientador: Prof. Jacques R. Heckmann



Roteiro

- Introdução
 - Problema a ser resolvido e objetivos
- Fundamentação Teórica
 - ORM, RTTI, Custom Attributes, Nullable
- Desenvolvimento
 - Requisitos, Especificação, Implementação, Resultados
- Conclusão
 - Extensões

Introdução

- Quem usa SGBD relacional?
- Quais os problemas
 - Rotinas de persistência repetitivas
 - Paradigma OO x Paradigma Relacional (Impedância)
 - SQL validado em runtime

Introdução

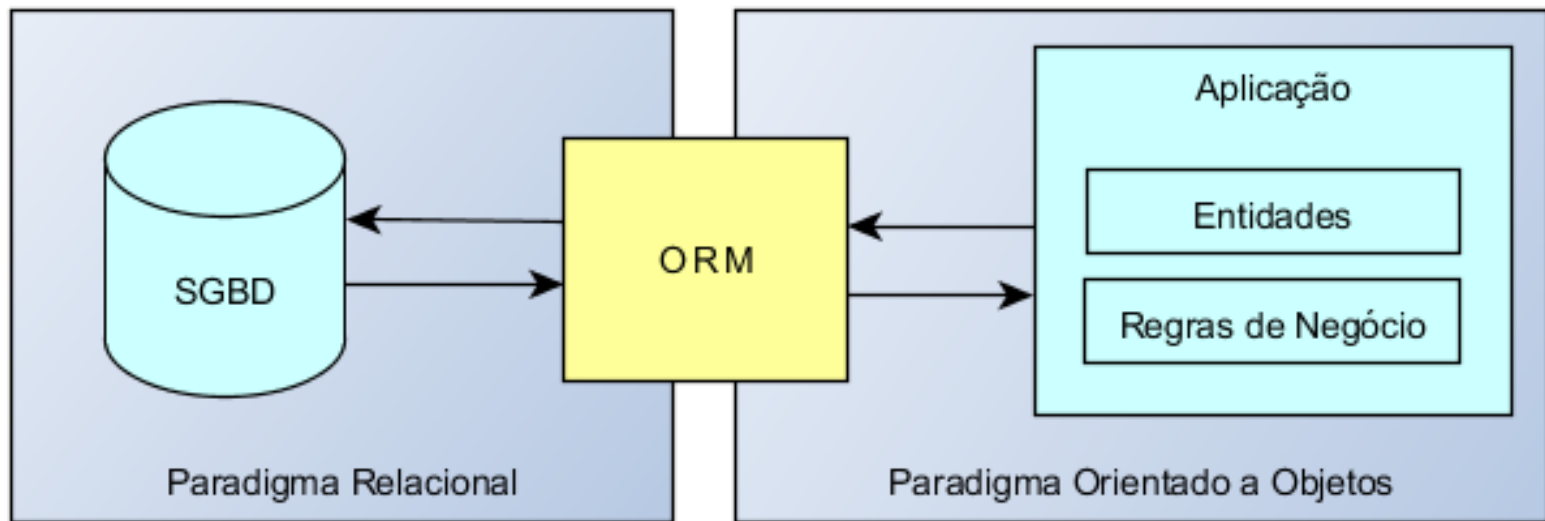
- Organizando melhor o sistema
 - Camadas
- Camada de persistência – *Framework*
 - Ponto único de conexão
 - Geração de código SQL
 - Especificidades de cada SGBD
 - Interface para programação OO

Objetivos do Trabalho

“Desenvolver um Framework de Persistência de Objetos em Delphi que utilize a técnica de Mapeamento Objeto-Relacional para fornecer a camada de persistência a uma aplicação.”

- Modelo de anotação
- Motor de persistência: Comandos SQL
- Gerenciador de objetos: Interface OO
- Suportar Firebird e MySQL

Mapeamento Objeto-Relacional



Mapeamento Objeto-Relacional

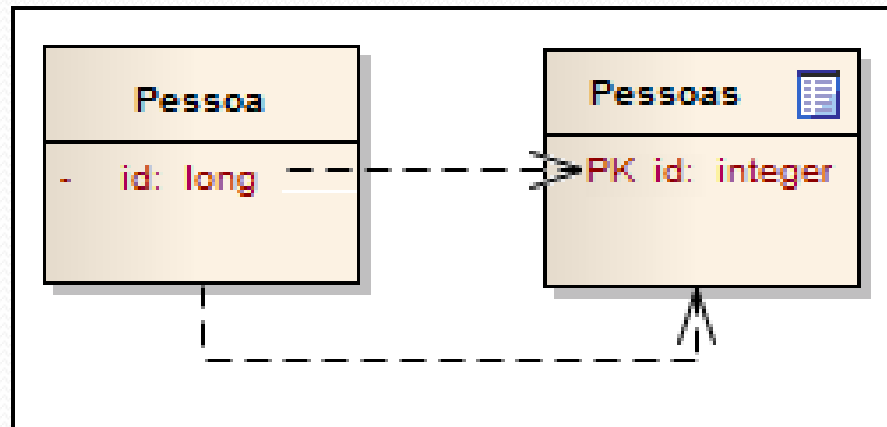
- Vantagens para as aplicações
 - Independência de SGBD
 - Apenas um paradigma: OO
 - Herança
 - Associações
 - Não se preocupa com código SQL

Mapeamento Objeto-Relacional

- Regras básicas de mapeamento
 - Classes para tabelas
 - Atributos para colunas
- Características obrigatórias
 - Object Identifier
 - Atributos simples, compostos ou multivalorados
 - Mapeamento de herança
 - Mapeamento de associações

Object Identifier

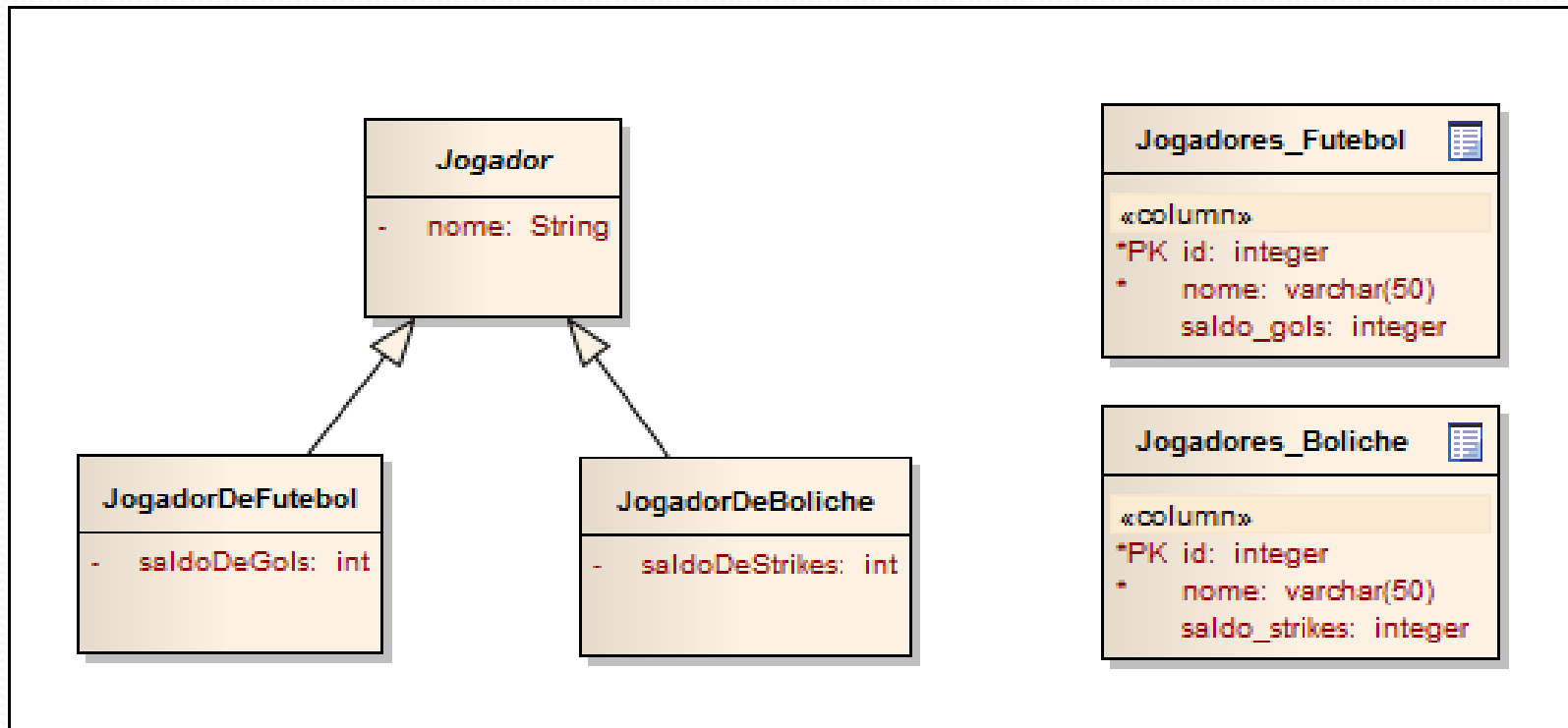
- Objeto.ID = Tabela.PK
- Sem significado para o mundo real
- Valor gerado automaticamente e imutável
- Objetos devem ser únicos
 - Mapa de identidade



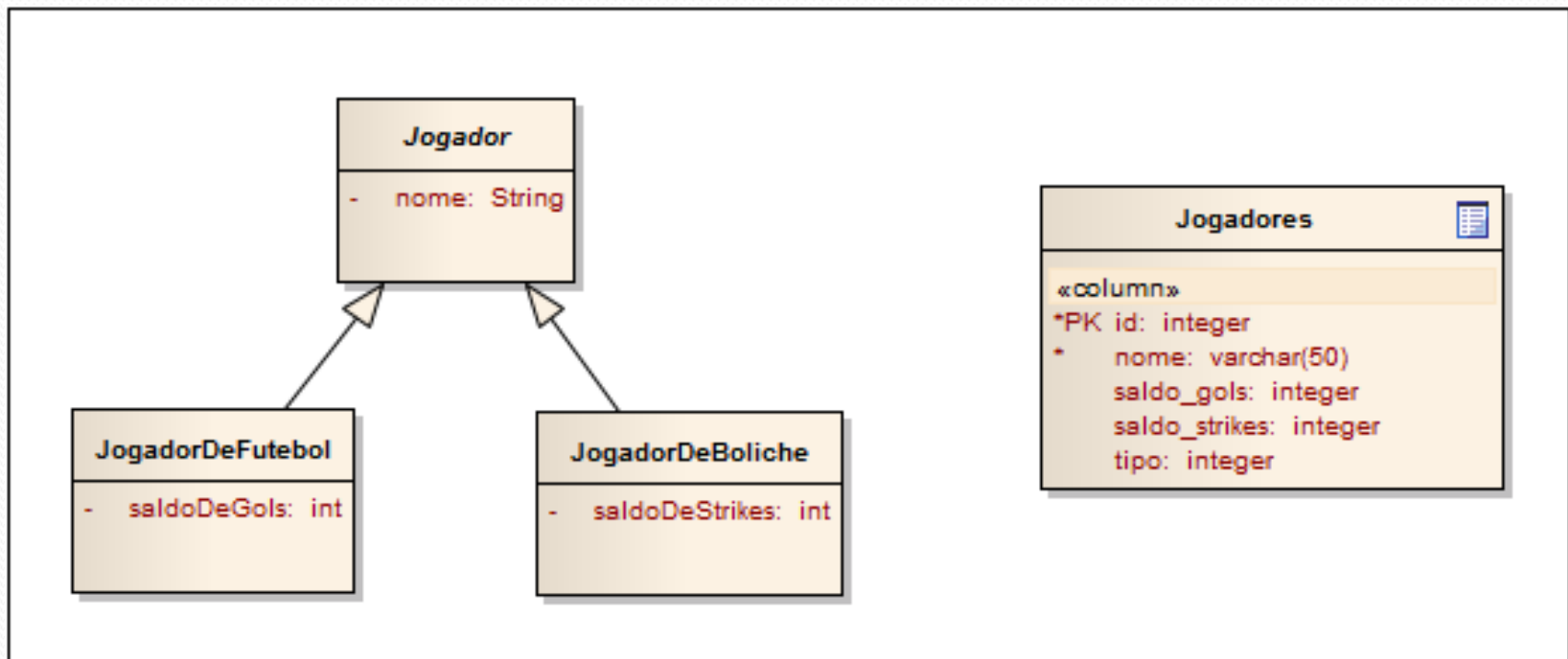
Mapeamento de Herança

- As 3 modalidades
 - Tabela por classe concreta
 - Tabela única
 - Tabela por subclasse

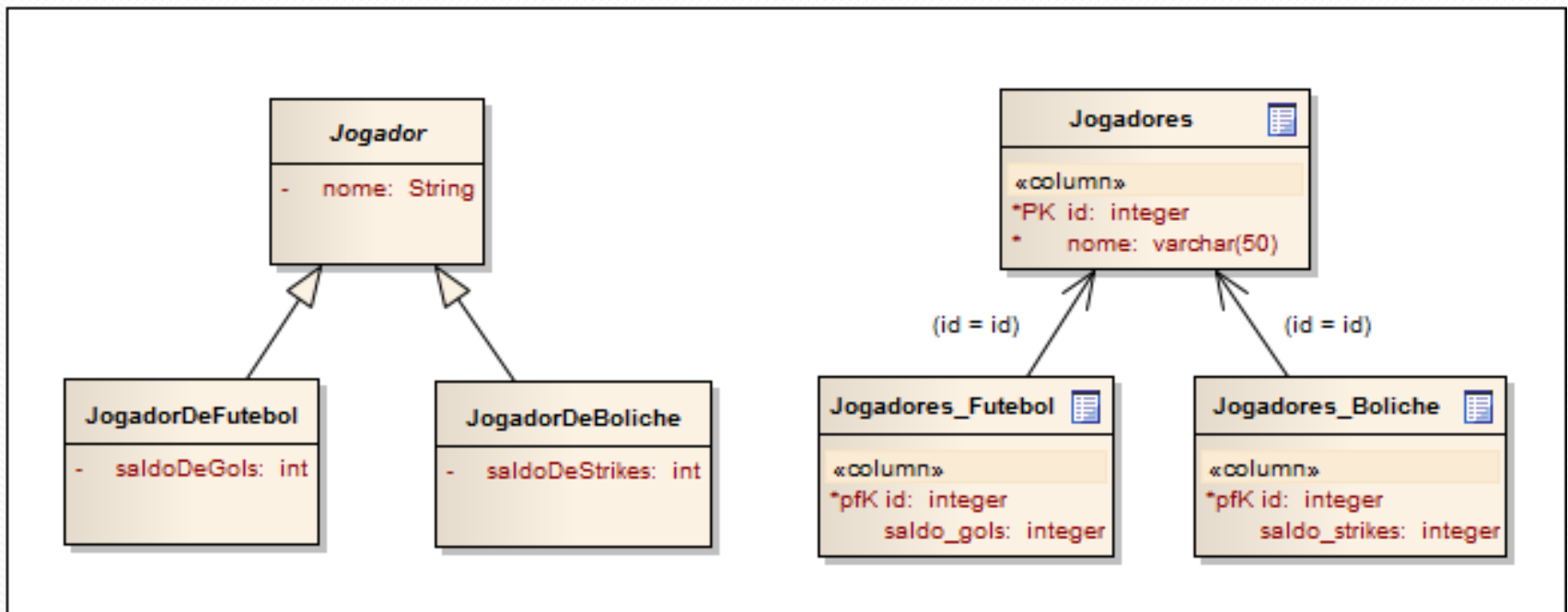
Herança de Tabela por Classe Concreta



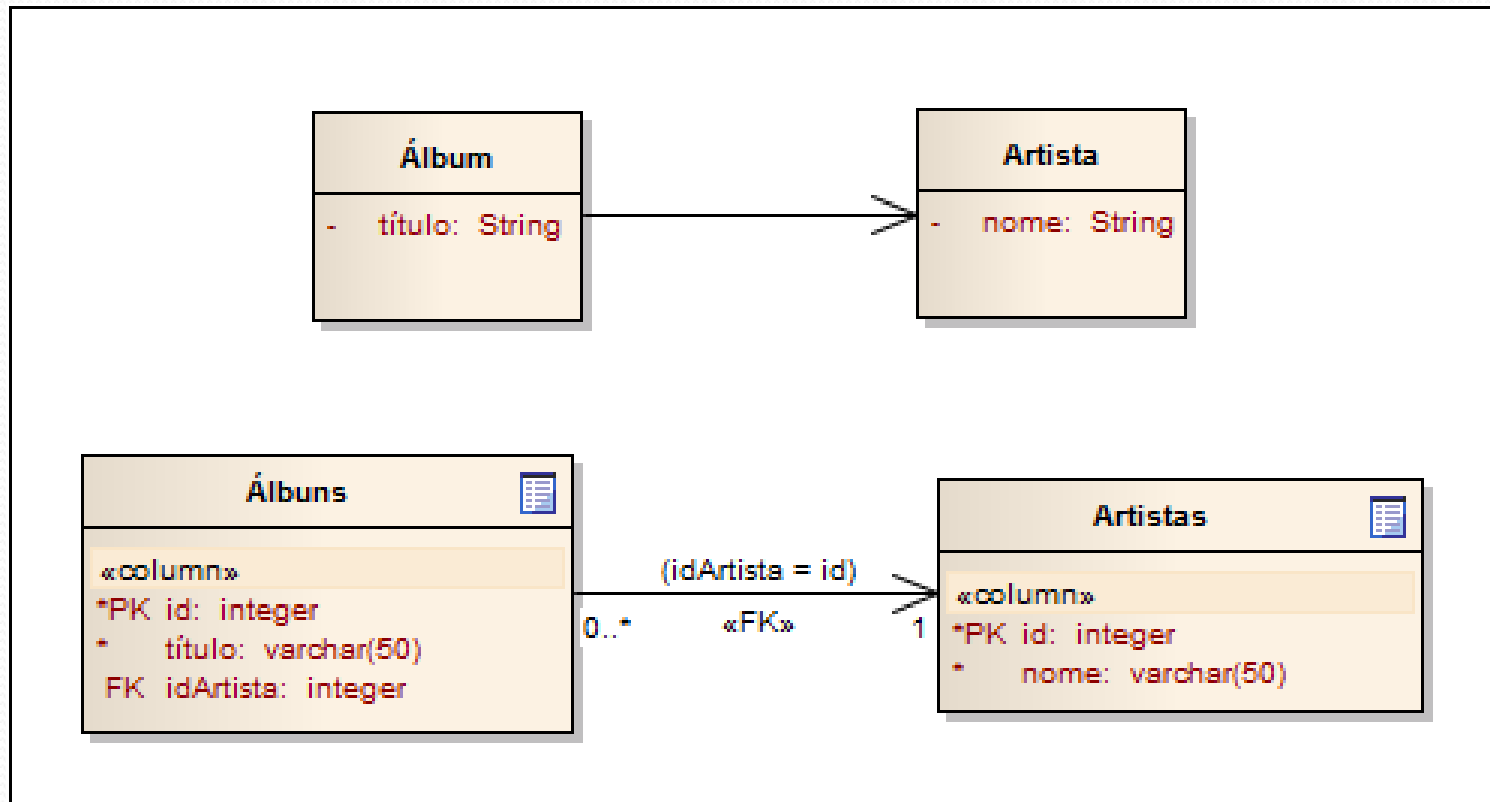
Herança de Tabela Única



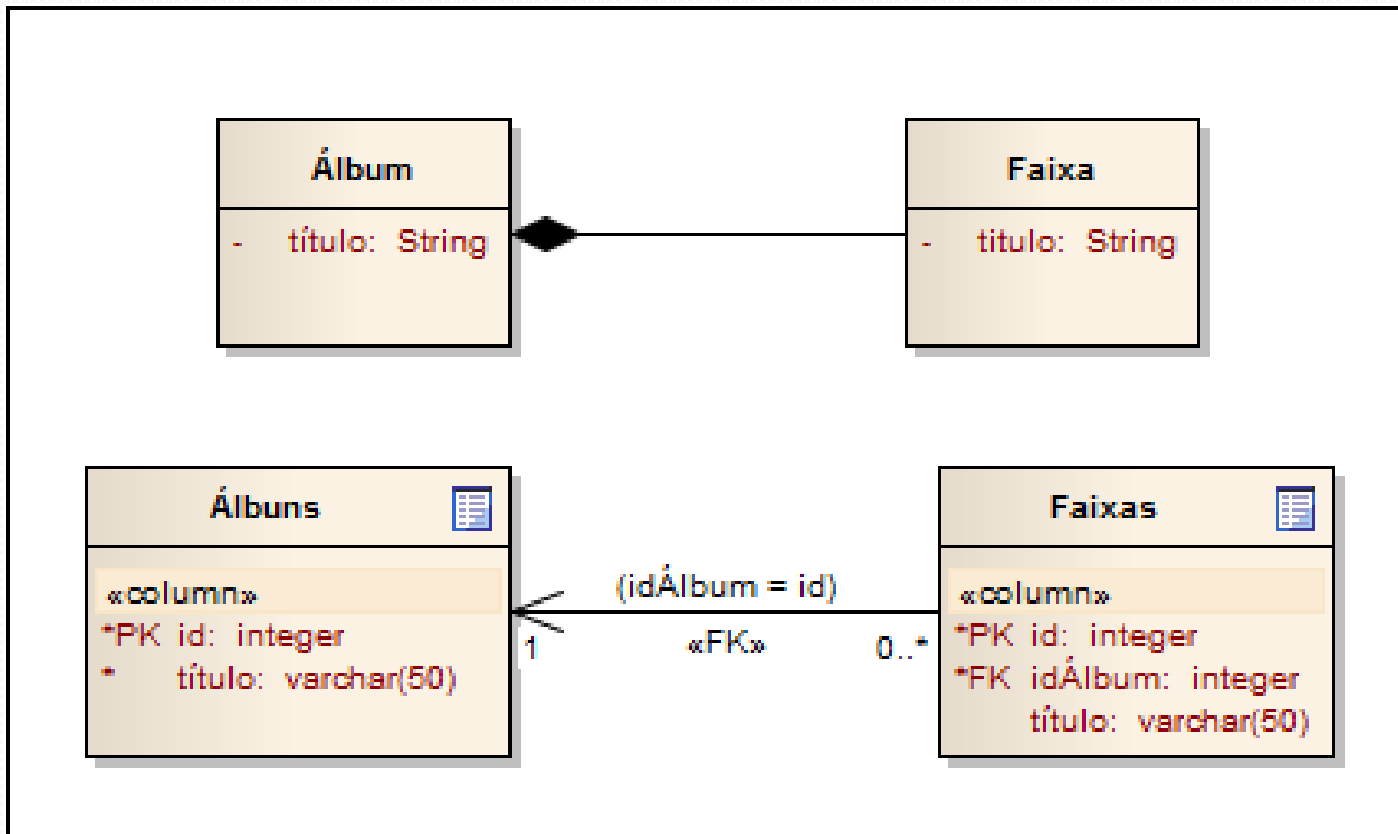
Herança de Tabela por Subclasse



Mapeamento de Associações Simples (N para 1)



Mapeamento de Associações Multivaloradas (1 para N)



RunTime Type Information

- Obter metadados em Runtime
 - Classes, records, atributos, *properties*, métodos e Interfaces implementadas
 - Obter e alterar valores dos objetos
- Delphi 2010:
 - Membros de qualquer visibilidade
 - Nova API, mais prática e completa

Custom Attributes

- Anotação de informações em tipos e membros
- Úteis para construções de *frameworks*
 - Ex: Definições de comportamento
- Criados a partir da TCustomAttribute
- Lidados via RTTI

```
TA algumaClasse = class
private
  [AtributoCustomizado('Texto com informações.')]
  procedure FazAlgumaCoisa;
end;
```

Nullable

- Atributos de tipo primitivo não aceitam NULL
- Colunas no banco de dados aceitam
- Solução: Estrutura genérica Nullable<T>
 - Ex: Nullable<Integer>, Nullable<string>
- Inspirada no .Net Framework

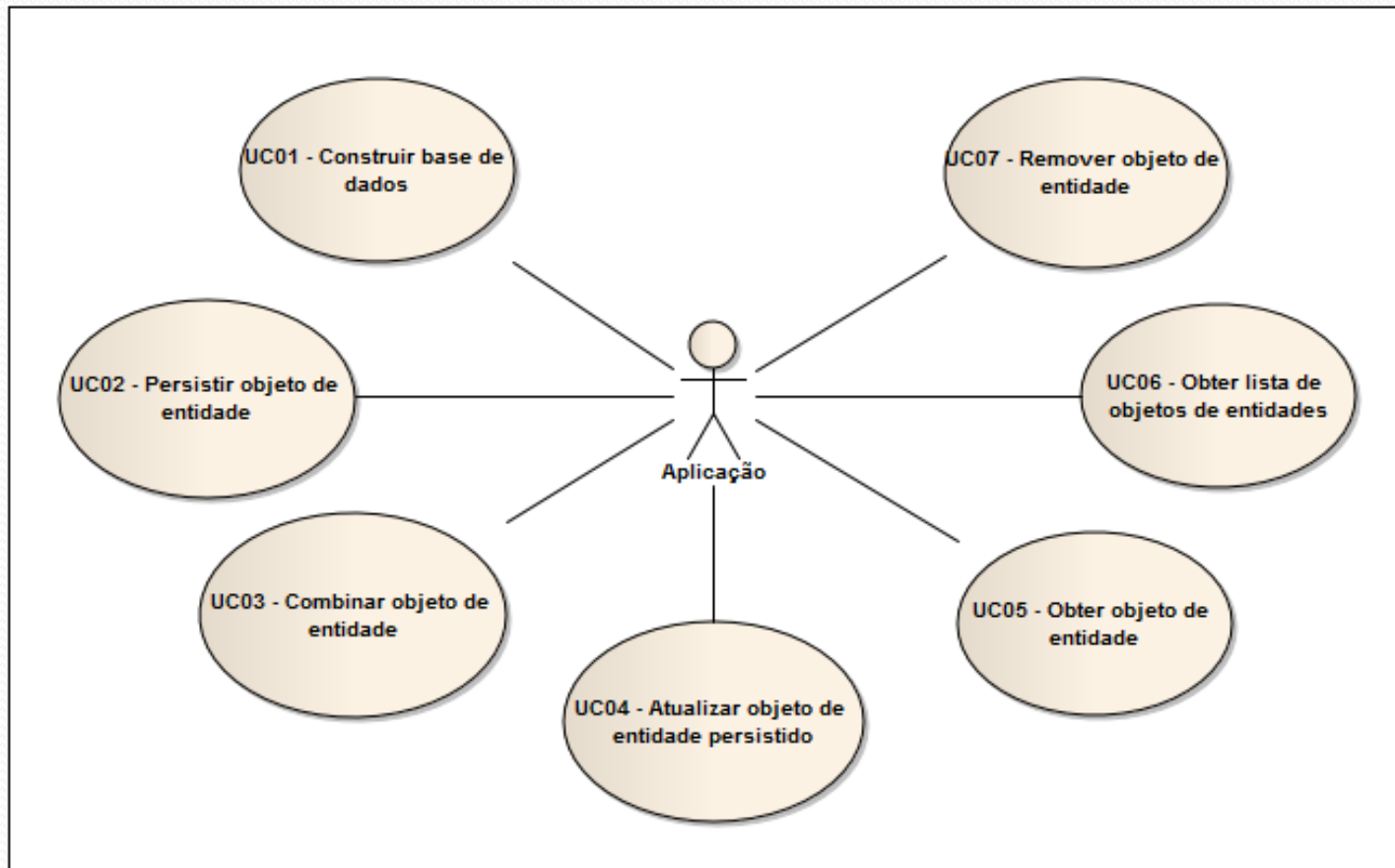
Trabalhos Correlatos

- Hibernate
 - Java, ORM, JPA (Java Persistence API)
- Instant Objects
 - Delphi, OPF, Datasets
- TCC Furb
 - Ferramenta para Aplicação do Padrão DAO em Sistemas Desenvolvidos em Delphi
 - Acadêmico: Marcelo Sardagna
 - Orientador: Adilson Vahldick

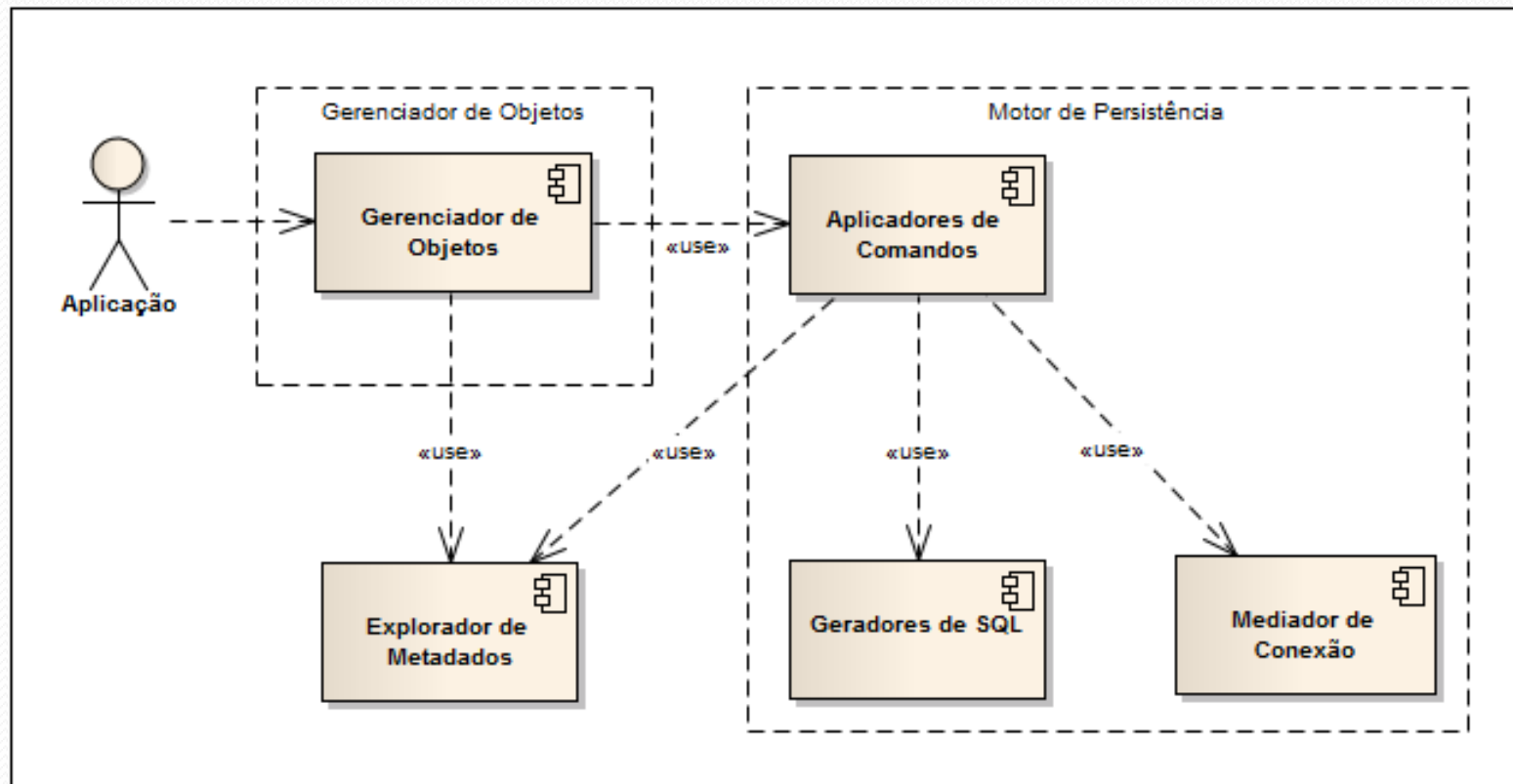
Requisitos do Trabalho

- Disponibilizar modelo de anotação para ORM
- Gerar comandos SQL
 - Select, Insert, Update e Delete
- Gerar as tabelas no banco de dados
- Dispor camada de persistência transparente
 - Criar, obter, salvar e remover objetos
- Gerenciador de entidades

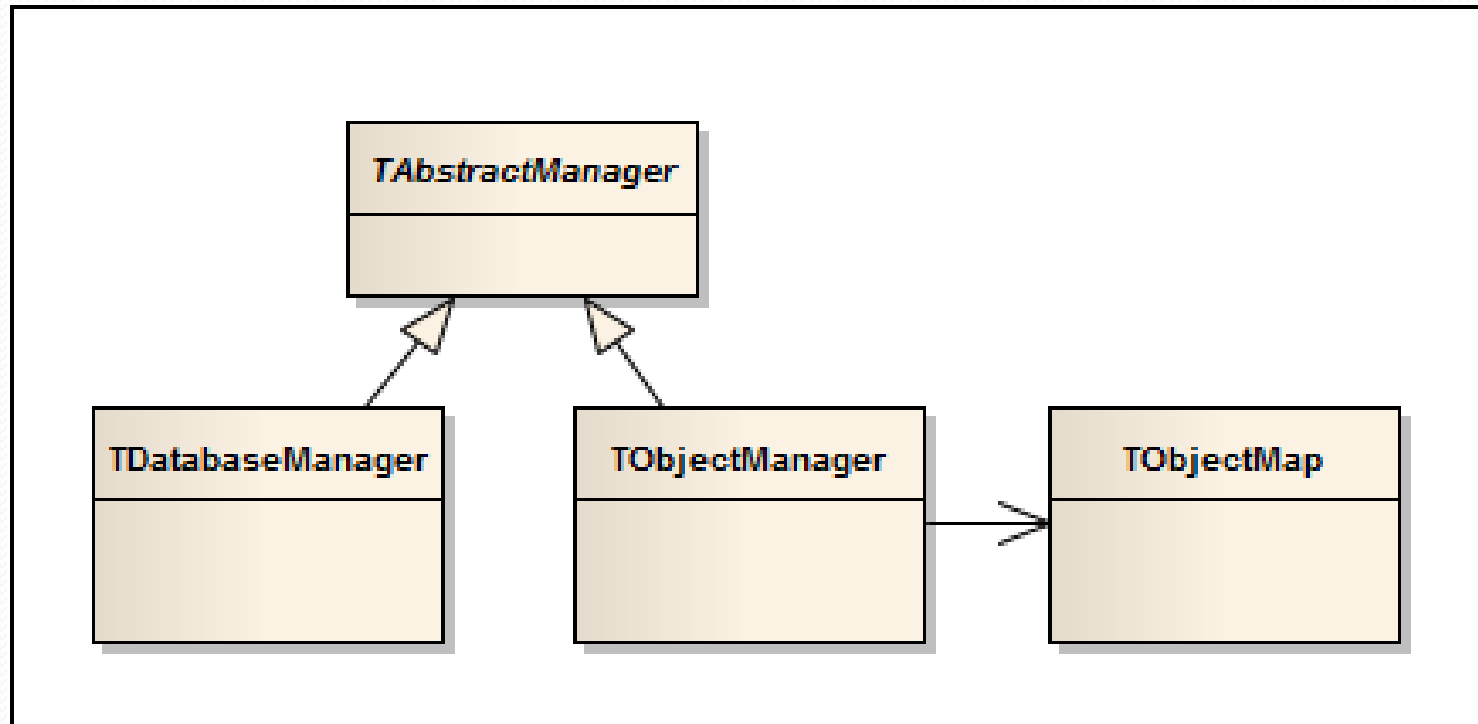
Casos de uso



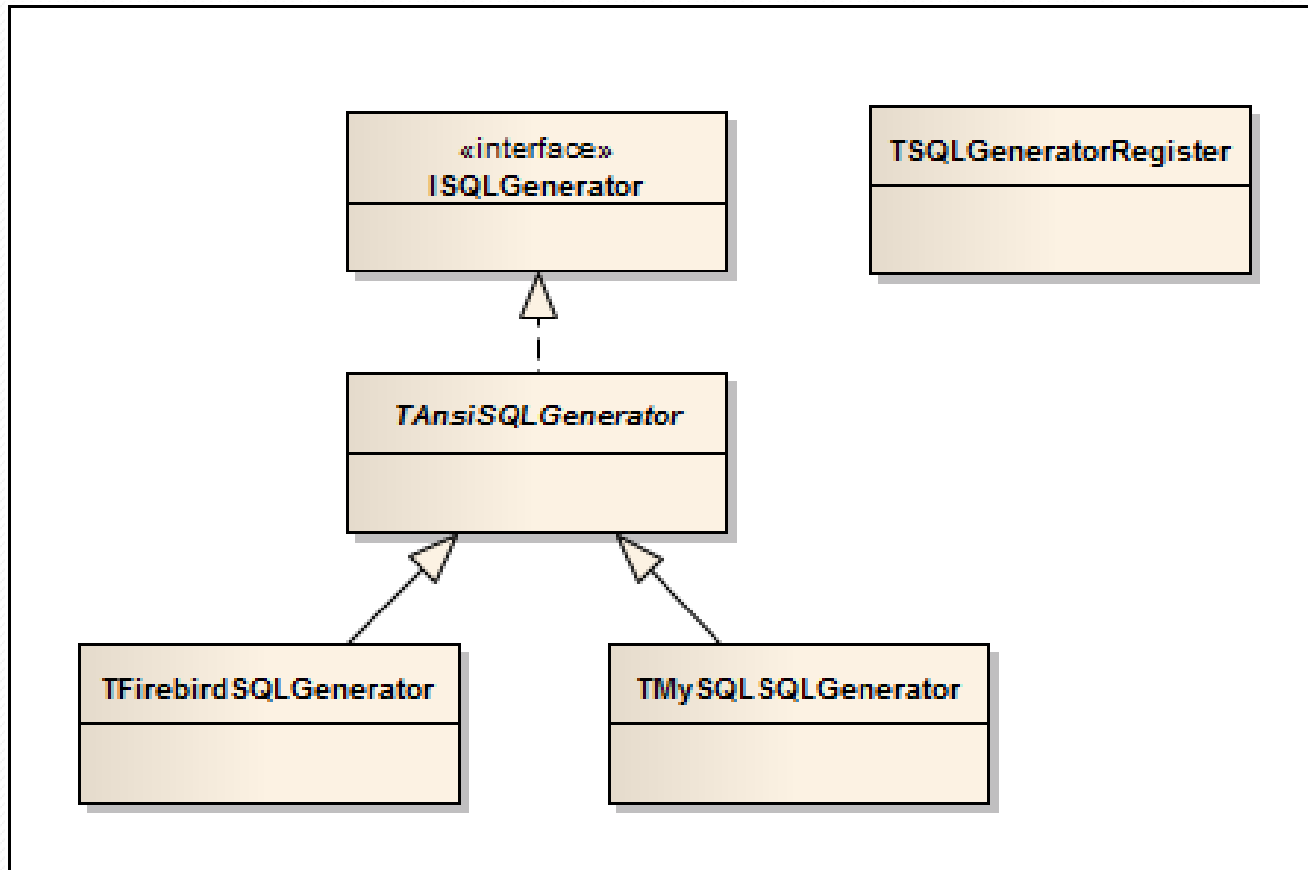
Visão geral - Componentes



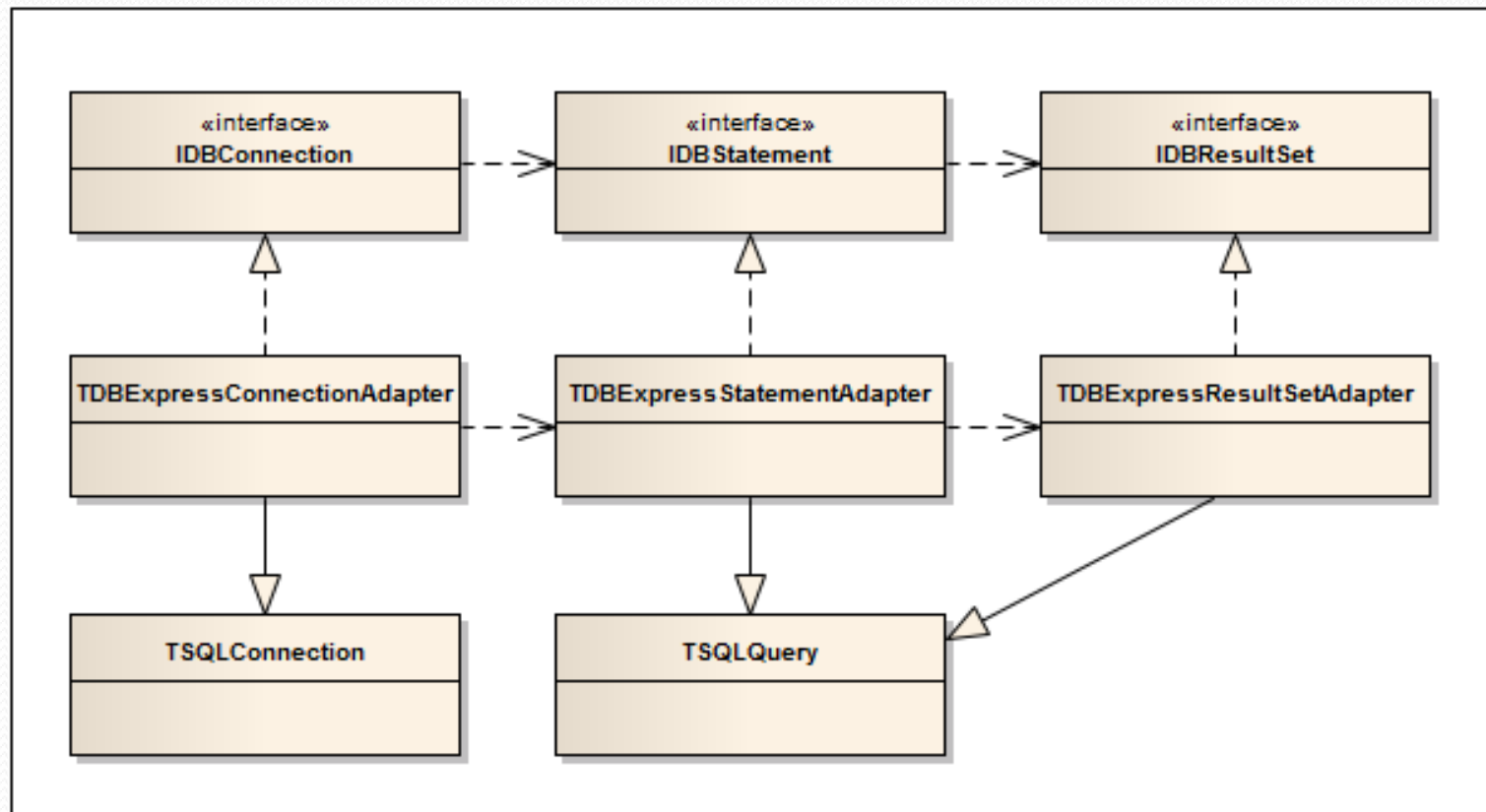
Gerenciador de objetos



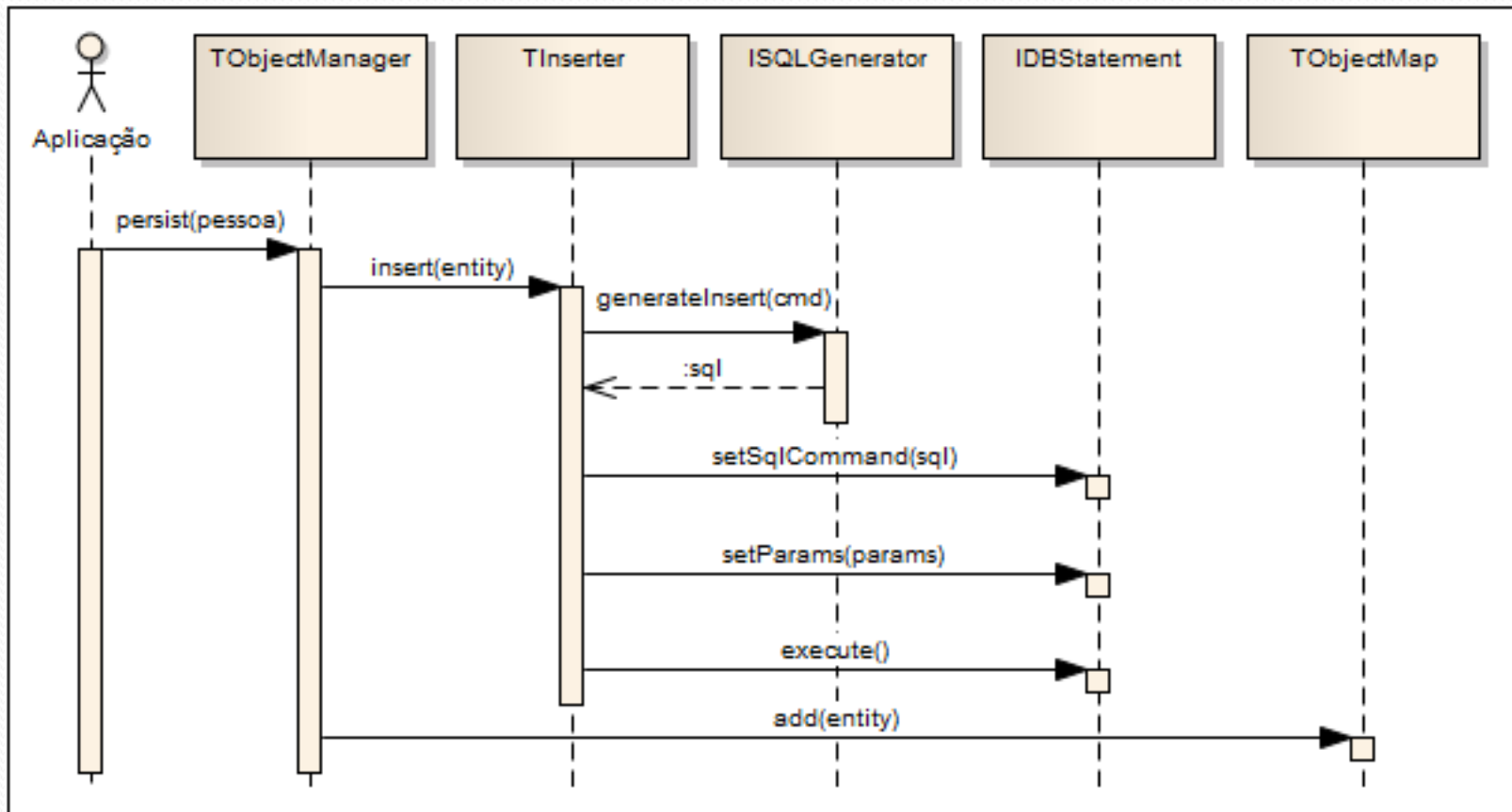
Geradores de SQL



Mediador de conexão



Interação – método persist()



Modelo de anotação

```
type
  [Entity]
  [Table('NOTAS_FISCAIS')]
  [Sequence('SEQ_NOTAS_FISCAIS')]
  TNotaFiscal = class
  private
    FId: LongInt;
    FNumero: Integer;
    FEmissao: TDateTime;
    FCliente: TCliente;
  public
    [Id]
    [Column('ID', [cpUnique, cpRequired])]
    property Id: LongInt read FId write FId;

    [Column('NUMERO', [cpUnique, cpRequired])]
    property Numero: Integer read FNumero write FNumero;

    [Column('DATA_EMISSAO', [cpRequired])]
    property Emissao: TDateTime read FEmissao write FEmissao;

    [Association(ftEager, orRequired, ctCascadeAll)]
    [JoinColumn('ID_CLIENTE')]
    property Cliente: TCliente read FCliente write FCliente;
  end;
```

Ferramentas utilizadas

- Delphi 2010 para Windows
 - VCL, RTL, DBExpress, RTTI, Custom Attributes
 - Nenhuma biblioteca de terceiros
- Controle de versão
 - Subversion + TortoiseSVN
- SGBDs
 - Firebird e MySQL

Padrões de projeto utilizados

- Martin Fowler
 - Identity Field
 - Identity Map
 - Foreign Key Mapping
 - Lazy Load
- Gamma (GoF)
 - Singleton
 - Adapter
 - Strategy
- Core J2EE Patterns
 - Data Access Object (DAO)

Comparações

Característica	Este trabalho	Hibernate	Instant Objects
Estratégias de mapeamento de herança	3	3	0
Suporta mapeamento de associações	X	X	X
Formas de mapeamento	Anotações	Anotações, XML	XML, Comentários
Permite consultas polimórficas	X	X	
Implementa Carga Tardia	X	X	X
Implementa Carga Ansiosa com Joins	X	X	X
Suporta tipos anuláveis	X	X	X
Integração nativa c/ controles de tela			X
Possui editor gráfico de classes nativo			X
Possui linguagem específica de consulta		X	X
Permite mapear usando tipos comuns da linguagem	X	X	
Constrói a base de dados	X	X	X

Conclusões

- Requisitos e objetivos atendidos com sucesso
- Ferramentas e padrões de projeto foram adequados
- Novo RTTI e Custom Attributes são produtivos
- Uso de Generics possibilitou estruturas como Nullable
- Suporte a vários tipos de aplicações
- Fácil de estender
- Inovação em persistência em Delphi Win32

Problemas enfrentados

- Controles de concorrência
 - Alta complexidade de implementação
 - Não foi implementado
- Lazy Load
 - Desafio: Disparar carregamento de forma transparente
 - Hibernate implementa com OAP no Java
 - Foi implementado com tipo genérico Proxy<T>

Extensões

- Consultas customizáveis (Ex: HQL, IQL)
- Associações N-N (com ou sem classe associativa)
- Ids compostos
- Suporte a mais SGBDs
- Gerência de configurações
- Controle avançado de transações
- Integração com outros frameworks

Obrigado.