A

-

Universidade Federal de Pernambuco

Centro de Informática



Banco de Dados Avançado

Banco de Dados Multimídia

Alunos:  
Antônio Carlos da Silva Júnior  
Bruno Florêncio Pinheiro  
Danilo Torres Ferreira  
Rafael Barbosa Gonçalves  
Vinicius Monteiro

Recife, 30 de novembro de 2010

Sumário

[1. Introdução 3](#_Toc278852072)

[2. MiniMundo 3](#_Toc278852073)

[3. Definição das Entidades 3](#_Toc278852074)

[4. Modelo Conceitual 4](#_Toc278852075)

[5. Esquema Relacional 5](#_Toc278852076)

[6. Scripts de Inserção e Recuperação 6](#_Toc278852077)

[7. Consultas Baseadas no Contexto 9](#_Toc278852078)

[Anexos 15](#_Toc278852079)

# Introdução

Este documento apresenta o modelo conceitual, esquema relacional e scripts do projeto de Banco de Dados Multimídia.

Nos próximos capítulos será mostrado a descrição do MiniMundo, a definição das entidades, a modelagem cenceitual e o esquema relacional do banco de dados , os scripts de inserção e recuperação dos dados multimídia do BD e os scripts de criação das tabelas.

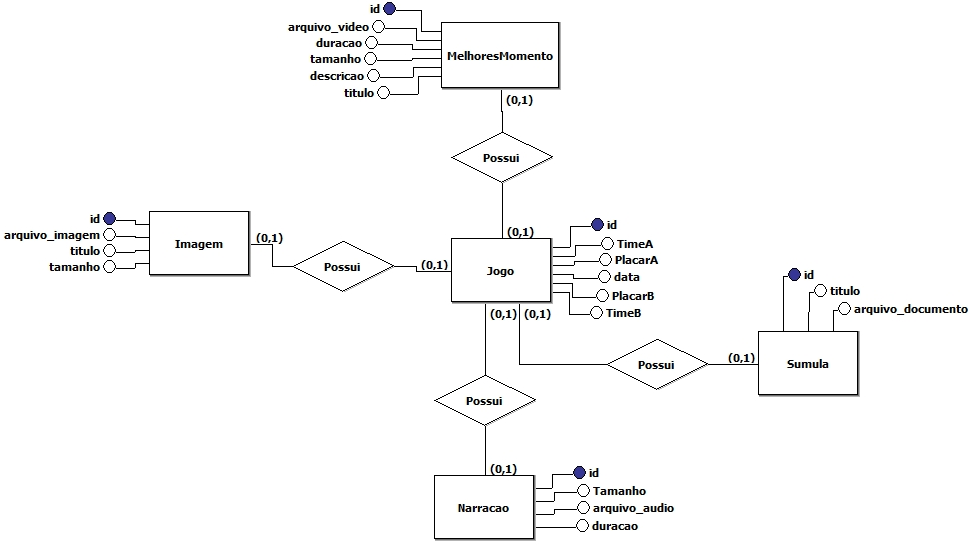
# MiniMundo

A aplicação desenvolvida consiste em um sistema que gerencia jogos para a Copa do Mundo de 2014. A Copa possui vários jogos associados e cada jogo armazena os times, a data de realização do jogo, a quantidade de gols marcados por cada equipe. Cada jogo possui diferentes tipos de mídia envolvidas nas partidas, como arquivos de vídeo com os melhores momentos da partida, de áudio com a narração do jogo, imagens da partida e ducumentos, como a súmula do árbitro por exemplo.

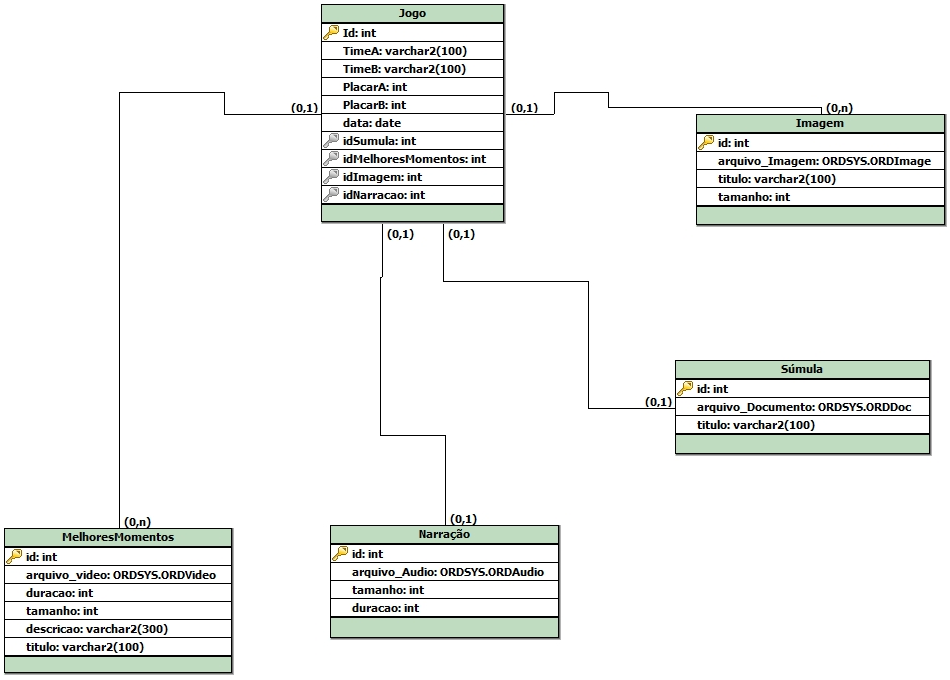
# Definição das Entidades

* *Jogo:* Possui o nome dos times, a quantidade de gols marcado por cada equipe e a data da partida.
* *MelhoresMomentos:* Possui um arquivo de vídeo, o título do vídeo, sua descrição, a duração, e o tamanho do vídeo;
* *Narracao:* Possui um arquivo de áudio, a duração, e o tamanho do áudio;
* *Imagem:* Possui um arquivo de imagem, o título da imagem e seu tamanho;
* *Sumula:* Possui o arquivo do documento e o título do documento;

# Modelo Conceitual



# Esquema Relacional



# Scripts de Inserção e Recuperação

* 1. **Inserção**
* **InsereAudio**

|  |
| --- |
| create or replace PROCEDURE insereAudio ( NOMEARQUIVO VARCHAR2, TAMANHO NUMBER, DURACAO NUMBER) IS    narracaoObj ORDSYS.ORDAudio;  ctx\_narracao RAW (2000) := NULL;  idAudio number;    BEGIN  INSERT INTO Narracao(ID, ARQUIVO\_AUDIO, DURACAO, TAMANHO)  VALUES(SEQ\_Audio.nextVal, ORDSYS.ORDAudio.init(), DURACAO, TAMANHO);  --pega o som inserido para colocar seu conteudo  Select SEQ\_AUDIO.CURRVAL into idAudio from dual;  SELECT n.ARQUIVO\_AUDIO INTO narracaoObj FROM Narracao n WHERE n.id=idAudio FOR UPDATE;  -- seta as propriedades do objeto do tipo audio  narracaoObj.setSource ('FILE', 'G102IF695\_EQ05', nomeArquivo) ;  narracaoObj.import (ctx\_narracao);  narracaoObj.setProperties (ctx\_narracao);    -- atualiza a tupla  UPDATE Narracao n SET n.ARQUIVO\_AUDIO = narracaoObj WHERE n.id = idAudio;  COMMIT ;  end insereAudio; |

* **InsereDocumento**

|  |
| --- |
| create or replace  PROCEDURE insereDocumento (  NOMEARQUIVO VARCHAR2, TITULO VARCHAR2) IS  sumulaObj ORDSYS.ORDDoc;  ctx\_sumula RAW (2000) := NULL ;  idDocumento number;  BEGIN  INSERT INTO Sumula(ID, ARQUIVO\_Documento, TITULO)  VALUES(SEQ\_Documento.nextVal, ORDSYS.ORDDoc.init(), TITULO);  Select SEQ\_Documento.CURRVAL into idDocumento from dual;  SELECT s.ARQUIVO\_Documento INTO sumulaObj FROM Sumula s WHERE s.id=idDocumento FOR UPDATE;  sumulaObj.setSource ('FILE', 'G102IF695\_EQ05', nomeArquivo) ;  sumulaObj.import (ctx\_sumula, false);  sumulaObj.setMimeType ('application / pdf') ;    UPDATE Sumula s SET s.ARQUIVO\_Documento = sumulaObj WHERE s.id = idDocumento;  COMMIT ;  end insereDocumento; |

* **InsereJogo**

|  |
| --- |
| create or replace PROCEDURE insereJogo (  TIMEA VARCHAR2,  TIMEB VARCHAR2,  PLACARA NUMBER,  PLACARB NUMBER,  DATA DATE,  IDDOCUMENTO NUMBER,  IDAUDIO NUMBER,  IDIMAGEM NUMBER,  IDVIDEO NUMBER) IS  BEGIN  INSERT INTO JOGO(ID, TIMEA, TIMEB, PLACARA, PLACARB, DATA, IDSUMULA, IDNARRACAO, IDIMAGEM, IDMelhoresMomentos)  VALUES(SEQ\_JOGO.nextVal, TIMEA, TIMEB, PLACARA, PLACARB, DATA, IDDOCUMENTO, IDAUDIO, IDIMAGEM, IDVIDEO);  END insereJogo; |

* **InsereImagem**

|  |
| --- |
| create or replace  PROCEDURE insereImagem (  NOMEARQUIVO VARCHAR2, TITULO VARCHAR2, TAMANHO NUMBER) IS  fotoObj ORDSYS.ORDImage;  fotoObj\_sig ORDSYS.ORDImageSignature;  ctx\_foto RAW (2000) := NULL;  idImagem number;  BEGIN  INSERT INTO Imagem(ID, ARQUIVO\_IMAGEM, Arquivo\_Imagem\_sig, TITULO, TAMANHO)  VALUES(SEQ\_IMAGEM.nextVal, ORDSYS.ORDImage.init('FILE', 'G102IF695\_EQ05', NOMEARQUIVO), ORDSYS.ORDImageSignature.init(), TITULO, TAMANHO);  Select SEQ\_Imagem.CURRVAL into idImagem from dual;  SELECT i.arquivo\_imagem, i.arquivo\_imagem\_sig INTO fotoObj , fotoObj\_sig  FROM imagem i WHERE i.id = idImagem FOR UPDATE ;  fotoObj.setProperties ;  fotoObj.import (ctx\_foto) ;  fotoObj\_sig.generateSignature (fotoObj);    UPDATE imagem SET arquivo\_imagem = fotoObj , arquivo\_imagem\_sig = fotoObj\_sig WHERE id = idImagem;  COMMIT;    end insereImagem; |

* **InsereVideo**

|  |
| --- |
| create or replace  PROCEDURE insereVideo (  NOMEARQUIVO VARCHAR2, TITULO VARCHAR2, DESCRICAO VARCHAR2, TAMANHO NUMBER, DURACAO NUMBER) IS  videoObj ORDSYS.ORDVideo;  ctx\_video RAW(4000) := NULL;  idVideo number;  BEGIN  INSERT INTO MelhoresMomentos(ID, ARQUIVO\_VIDEO, DURACAO, TITULO, DESCRICAO, TAMANHO)  VALUES(SEQ\_VIDEO.nextVal, ORDSYS.ORDVideo.init(), DURACAO, TITULO, DESCRICAO, TAMANHO);  Select SEQ\_VIDEO.CURRVAL into idVideo from dual;  SELECT m.arquivo\_video INTO videoObj  FROM MelhoresMomentos m  WHERE m.id = idVideo FOR UPDATE ;    videoObj.setSource ('FILE', 'G102IF695\_EQ05', NOMEARQUIVO) ;  videoObj.import (ctx\_video) ;  videoObj.setProperties (ctx\_video) ;  UPDATE MelhoresMomentos m SET m.arquivo\_video = videoObj  WHERE m.id = idVideo;  COMMIT ;  end insereVideo; |

* 1. **Recuperação**

|  |
| --- |
| public static final String cadastrarAudio = "{call insereAudio(?, ?, ?)}";  public static final String cadastrarImagem = "{call insereImagem(?, ?, ?)}";  public static final String cadastrarDocumento = "{call insereDocumento(?, ?)}";  public static final String cadastrarVideo = "{call insereVideo (?, ?, ?, ?, ?)}";  public static final String cadastrarJogo = "{call insereJogo (?, ?, ?, ?, ?, ?, ?, ?, ?)}";  public static final String listarJogo = "SELECT \* FROM Jogo ORDER BY data"; |

# Consultas Baseadas no Contexto

* ComparacaoScore

|  |
| --- |
| create or replace procedure ComparacaoScore (  idImagem integer, imagemNova varchar2  )is  imagem1 ORDSYS.ORDImage;  imagem2 ORDSYS.ORDImage;  imagem\_sig1 ORDSYS.ORDImageSignature;  imagem\_sig2 ORDSYS.ORDImageSignature;  score1 Integer;  score2 Integer;  score3 Integer;  score4 Integer;  resultado FLOAT;  BEGIN    insert into Imagem (Id, Arquivo\_Imagem, Arquivo\_Imagem\_sig, titulo, tamanho) values (-1, ORDSYS.ORDImage.init('FILE', 'G102IF695\_EQ05', imagemNova), ORDSYS.ORDImageSignature.init(), '', 0);  SELECT Arquivo\_Imagem, Arquivo\_Imagem\_sig INTO imagem1, imagem\_sig1 FROM Imagem WHERE id = idImagem FOR UPDATE;  SELECT Arquivo\_Imagem, Arquivo\_Imagem\_sig INTO imagem2, imagem\_sig2 FROM Imagem WHERE id = -1 FOR UPDATE;    imagem\_sig1.generateSignature (imagem1);  imagem\_sig2.generateSignature (imagem2);    score1 := ORDSYS.ORDImageSignature.evaluateScore  (imagem\_sig1,imagem\_sig2,'color="1.0",texture=0,shape=0, location=0');    score2 := ORDSYS.ORDImageSignature.evaluateScore  (imagem\_sig1,imagem\_sig2,'color="0.0",texture=1.0,shape=0, location=0');    score3 := ORDSYS.ORDImageSignature.evaluateScore  (imagem\_sig1,imagem\_sig2,'color="0.0",texture=0,shape=1.0, location=0');    score4 := ORDSYS.ORDImageSignature.evaluateScore  (imagem\_sig1,imagem\_sig2,'color="0",texture=0,shape="0.01", location="0.99"');  resultado := (score1 + score2 + score3 + score4)/4;  DBMS\_OUTPUT.PUT\_LINE( CONCAT('Média do Score: ', resultado));  delete from Imagem where id = -1;  END ComparacaoScore; |

* ComparaFotoCor

|  |
| --- |
| create or replace procedure comparaFotoCor (  idImagem integer, imagemNova varchar2  )is  imagem1 ORDSYS.ORDImage;  imagem2 ORDSYS.ORDImage;  imagem\_sig1 ORDSYS.ORDImageSignature;  imagem\_sig2 ORDSYS.ORDImageSignature;  resultado Integer;  BEGIN    insert into Imagem (Id, Arquivo\_Imagem, Arquivo\_Imagem\_sig, titulo, tamanho) values (-1, ORDSYS.ORDImage.init('FILE', 'G102IF695\_EQ05', imagemNova), ORDSYS.ORDImageSignature.init(), '', 0);  SELECT Arquivo\_Imagem, Arquivo\_Imagem\_sig INTO imagem1, imagem\_sig1 FROM Imagem WHERE id = idImagem FOR UPDATE;  SELECT Arquivo\_Imagem, Arquivo\_Imagem\_sig INTO imagem2, imagem\_sig2 FROM Imagem WHERE id = -1 FOR UPDATE;    imagem\_sig1.generateSignature (imagem1);  imagem\_sig2.generateSignature (imagem2);    -- Compara as images  resultado := ORDSYS.ORDImageSignature.isSimilar(imagem\_sig1,imagem\_sig2,'color=1.0,texture=0.0,shape=0.0,location=0.0',20);    if resultado = 1 THEN  DBMS\_OUTPUT.PUT\_LINE('As imagens possuem cores similares!');  ELSE  DBMS\_OUTPUT.PUT\_LINE('As imagens não possuem cores similares.');  END IF;  delete from Imagem where id = -1;  COMMIT ;  END comparaFotoCor; |

* ComparaFotoLocation

|  |
| --- |
| create or replace  procedure comparaFotoLocation (  idImagem integer, imagemNova varchar2  )is  imagem1 ORDSYS.ORDImage;  imagem2 ORDSYS.ORDImage;  imagem\_sig1 ORDSYS.ORDImageSignature;  imagem\_sig2 ORDSYS.ORDImageSignature;  resultado Integer;  BEGIN    insert into Imagem (Id, Arquivo\_Imagem, Arquivo\_Imagem\_sig, titulo, tamanho) values (-1, ORDSYS.ORDImage.init('FILE', 'G102IF695\_EQ05', imagemNova), ORDSYS.ORDImageSignature.init(), '', 0);  SELECT Arquivo\_Imagem, Arquivo\_Imagem\_sig INTO imagem1, imagem\_sig1 FROM Imagem WHERE id = idImagem FOR UPDATE;  SELECT Arquivo\_Imagem, Arquivo\_Imagem\_sig INTO imagem2, imagem\_sig2 FROM Imagem WHERE id = -1 FOR UPDATE;    imagem\_sig1.generateSignature (imagem1);  imagem\_sig2.generateSignature (imagem2);    -- Compara as images  resultado := ORDSYS.ORDImageSignature.isSimilar(imagem\_sig1,imagem\_sig2,'color=0,texture=0,shape="0.01",location="0.99"',20);    if resultado = 1 THEN  DBMS\_OUTPUT.PUT\_LINE('As imagens possuem localização similares!');  ELSE  DBMS\_OUTPUT.PUT\_LINE('As imagens não possuem localização similares!');  END IF;  delete from Imagem where id = -1;  COMMIT ;  END comparaFotoLocation; |

* ComparaFotoShape

|  |
| --- |
| create or replace  procedure comparaFotoShape (  idImagem integer, imagemNova varchar2  )is  imagem1 ORDSYS.ORDImage;  imagem2 ORDSYS.ORDImage;  imagem\_sig1 ORDSYS.ORDImageSignature;  imagem\_sig2 ORDSYS.ORDImageSignature;  resultado Integer;  BEGIN    insert into Imagem (Id, Arquivo\_Imagem, Arquivo\_Imagem\_sig, titulo, tamanho) values (-1, ORDSYS.ORDImage.init('FILE', 'G102IF695\_EQ05', imagemNova), ORDSYS.ORDImageSignature.init(), '', 0);  SELECT Arquivo\_Imagem, Arquivo\_Imagem\_sig INTO imagem1, imagem\_sig1 FROM Imagem WHERE id = idImagem FOR UPDATE;  SELECT Arquivo\_Imagem, Arquivo\_Imagem\_sig INTO imagem2, imagem\_sig2 FROM Imagem WHERE id = -1 FOR UPDATE;    imagem\_sig1.generateSignature (imagem1);  imagem\_sig2.generateSignature (imagem2);    -- Compara as images  resultado := ORDSYS.ORDImageSignature.isSimilar(imagem\_sig1,imagem\_sig2,'color=0,texture=0,shape="1.0",location=0',30);    if resultado = 1 THEN  DBMS\_OUTPUT.PUT\_LINE('As imagens possuem shape similares!');  ELSE  DBMS\_OUTPUT.PUT\_LINE('As imagens não possuem shape similares!');  END IF;  delete from Imagem where id = -1;  COMMIT ;  END comparaFotoShape; |

* ComparaFotoTextura

|  |
| --- |
| create or replace  procedure comparaFotoTextura (  idImagem integer, imagemNova varchar2  )is  imagem1 ORDSYS.ORDImage;  imagem2 ORDSYS.ORDImage;  imagem\_sig1 ORDSYS.ORDImageSignature;  imagem\_sig2 ORDSYS.ORDImageSignature;  resultado Integer;  BEGIN    insert into Imagem (Id, Arquivo\_Imagem, Arquivo\_Imagem\_sig, titulo, tamanho) values (-1, ORDSYS.ORDImage.init('FILE', 'G102IF695\_EQ05', imagemNova), ORDSYS.ORDImageSignature.init(), '', 0);  SELECT Arquivo\_Imagem, Arquivo\_Imagem\_sig INTO imagem1, imagem\_sig1 FROM Imagem WHERE id = idImagem FOR UPDATE;  SELECT Arquivo\_Imagem, Arquivo\_Imagem\_sig INTO imagem2, imagem\_sig2 FROM Imagem WHERE id = -1 FOR UPDATE;    imagem\_sig1.generateSignature (imagem1);  imagem\_sig2.generateSignature (imagem2);    -- Compara as images  resultado := ORDSYS.ORDImageSignature.isSimilar(imagem\_sig1,imagem\_sig2,'color=0,texture="1.0",shape=0,location=0',10);    if resultado = 1 THEN  DBMS\_OUTPUT.PUT\_LINE('As imagens possuem textura similares!');  ELSE  DBMS\_OUTPUT.PUT\_LINE('As imagens não possuem textura similares!');  END IF;  delete from Imagem where id = -1;  COMMIT ;  END comparaFotoTextura; |

# Anexos

* **Script de Criação das tabelas**

1. **Jogo**

|  |
| --- |
| CREATE TABLE Jogo (  Id int PRIMARY KEY,  TimeA varchar2(100),  TimeB varchar2(100),  PlacarA int,  PlacarB int,  data date,  idSumula int,  idMelhoresMomentos int,  idImagem int,  idNarracao int,  FOREIGN KEY(idSumula) REFERENCES Súmula (id),  FOREIGN KEY(idMelhoresMomentos) REFERENCES MelhoresMomentos (id),  FOREIGN KEY(idImagem) REFERENCES Imagem (id),  FOREIGN KEY(idNarracao) REFERENCES Narração (id)  ) |

1. **MelhoresMomentos**

|  |
| --- |
| CREATE TABLE MelhoresMomentos (  id int PRIMARY KEY,  arquivo\_video ORDSYS.ORDVideo,  duracao int,  tamanho int,  descricao varchar2(300),  titulo varchar2(100)  ) |

1. **Imagem**

|  |
| --- |
| CREATE TABLE Imagem (  id int PRIMARY KEY,  arquivo\_Imagem ORDSYS.ORDImage,  titulo varchar2(100),  tamanho int  ) |

1. **Narração**

|  |
| --- |
| CREATE TABLE Narração (  id int PRIMARY KEY,  arquivo\_Audio ORDSYS.ORDAudio,  tamanho int,  duracao int  ) |

1. **Súmula**

|  |
| --- |
| CREATE TABLE Súmula (  id int PRIMARY KEY,  arquivo\_Documento ORDSYS.ORDDoc,  titulo varchar2(100)  ) |