

**RICARDO CZELUSNIAK DA SILVA**

*BENCHMARK EM BANCO DE DADOS MULTIMÍDIA:  
ANÁLISE DE DESEMPENHO EM RECUPERAÇÃO DE  
OBJETOS MULTIMÍDIA*

**CURITIBA 2006**

**RICARDO CZELUSNIAK DA SILVA**

***BENCHMARK EM BANCO DE DADOS MULTIMÍDIA:  
ANÁLISE DE DESEMPENHO EM RECUPERAÇÃO DE  
OBJETOS MULTIMÍDIA***

Dissertação apresentada como requisito parcial à obtenção do grau de mestre. Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientadora: Dr<sup>a</sup> Maria Salete M. G. Vaz

**CURITIBA 2006**

## **AGRADECIMENTOS**

Agradeço a Deus, por sempre me dar forças para o alcance de todos os meus objetivos, inclusive a realização deste trabalho.

Agradeço também à minha orientadora e amiga Professora Dr<sup>a</sup>. Maria Salete Marcon Gomes Vaz, a qual sempre esteve disponível para a orientação de toda a minha formação e em especial neste trabalho.

Agradecimento especial à minha noiva Yamara Gardingo pela compreensão da minha ausência e falta de dedicação durante o desenvolvimento desse trabalho.

Agradeço também a todas as pessoas que contribuíram direta e indiretamente para a realização deste trabalho.

Meus amigos do mestrado um abraço especial a todos vocês que sempre estiveram juntos nessa grande etapa de nossa vida.

## SUMÁRIO

LISTA DE FIGURAS.....	iii
LISTA DE QUADROS.....	iv
LISTA DE SIGLAS.....	v
RESUMO.....	vi
ABSTRACT.....	vii
<b>1. INTRODUÇÃO .....</b>	<b>1</b>
<b>2. SISTEMAS DE BANCO DE DADOS PARA APLICAÇÕES MULTIMÍDIA .....</b>	<b>4</b>
2.1. MULTIMÍDIA E OS TIPOS DE OBJETOS.....	4
2.2. GERENCIAMENTO DE OBJETOS MULTIMÍDIA .....	8
2.3. INDEXAÇÃO DE OBJETOS MULTIMÍDIA ATRAVÉS DE METADADOS .....	10
2.4. PESQUISA E RECUPERAÇÃO DE OBJETOS MULTIMÍDIA.....	14
2.5. ARMAZENAMENTO DE OBJETOS MULTIMÍDIA .....	18
<b>3. BENCHMARK - AVALIAÇÃO DE DESEMPENHO EM BANCO DE DADOS .....</b>	<b>22</b>
3.1. DEFINIÇÃO.....	22
3.2. METODOLOGIA .....	24
<b>4. ANÁLISE DE DESEMPENHO EM RECUPERAÇÃO DE OBJETOS MULTIMÍDIA NO POSTGRESQL.....</b>	<b>30</b>
4.1. AMBIENTE EXPERIMENTAL.....	30
4.2. ANÁLISE DE DESEMPENHO EM RECUPERAÇÃO DE IMAGENS .....	36
4.2.1. ARMAZENAMENTO POR REFERÊNCIA EXTERNA .....	36
4.2.2. ARMAZENAMENTO DE DADOS MULTIMÍDIA NÃO INTERPRETADOS - CAMPO <i>BLOB</i> .....	39
4.2.3. ARMAZENAMENTO ATRAVÉS DE OBJETOS.....	43
4.3. ANÁLISE COMPARATIVA.....	45
<b>5. CONCLUSÕES E TRABALHOS FUTUROS .....</b>	<b>50</b>
<b>REFERÊNCIAS.....</b>	<b>53</b>

## LISTA DE FIGURAS

Figura 1: Distinção entre conteúdo dos objetos.....	9
Figura 2: Metadados de Objeto Multimídia.....	11
Figura 3: Recuperação através de um identificador. ....	15
Figura 4: Recuperação através de sentenças condicionais.....	16
Figura 5: Recuperação por similaridade.....	16
Figura 6: Arquitetura de recuperação semântica (LOH, WIVES e FRAINER, 1999). ....	17
Figura 7: Utilização de <i>check-point</i> . (DEMURJIAN et. al, 1985).....	25
Figura 8: Esquema de dados do TPC-App. ....	28
Figura 9: Ambiente da análise. ....	31
Figura 10: Estrutura <i>check-point</i> em php.....	32
Figura 11: Conexão entre PHP e PostgreSQL.....	32
Figura 12: Resultados das consultas do Conjunto A com armazenamento de referência externa em ambiente mono-usuário .....	37
Figura 13: Resultados das consultas do Conjunto B com armazenamento de referência externa em ambiente mono-usuário .....	38
Figura 14: Resultados da consulta 4A e 4B com armazenamento através de referência externa em ambiente multi-usuário.....	38
Figura 15: Inserção de imagem no campo <i>bytea</i> escrito em PHP. ....	40
Figura 16: Resultados das consultas do Conjunto A com armazenamento através do campo <i>bytea</i> em ambiente mono-usuário .....	40
Figura 17: Resultados das consultas do Conjunto B com armazenamento através do campo <i>bytea</i> em ambiente mono-usuário .....	41
Figura 18: Resultados da consulta 4A com armazenamento através do campo <i>bytea</i> em ambiente multi-usuário.....	42
Figura 19: Resultados da consulta 4B com armazenamento através do campo <i>bytea</i> em ambiente multi-usuário.....	42
Figura 20: Armazenamento e recuperação de objeto utilizando o campo <i>oid</i> . ....	43
Figura 21: Resultados das consultas do Conjunto A com armazenamento através do campo <i>oid</i> em ambiente mono-usuário .....	44
Figura 22: Resultados das consultas do Conjunto B com armazenamento através do campo <i>oid</i> em ambiente mono-usuário .....	45

## LISTA DE QUADROS

Quadro 1: Comparativo entre as formas de armazenamento.....	20
Quadro 2: Grupos de consultas definidos por BORAL e DEWITT, (1984). .....	26
Quadro 3: Consultas representativas de cada grupo .....	26
Quadro 4: Grau de povoamento das tabelas.....	33
Quadro 5: Consultas para análise .....	35
Quadro 6: Comparação entre os métodos de armazenamento no Conjunto A de consultas em ambiente mono-usuário .....	46
Quadro 7: Comparação entre os métodos de armazenamento no Conjunto B de consultas em ambiente mono-usuário .....	47
Quadro 8: Comparação entre os métodos de armazenamento com a Consulta 4A em ambiente multi-usuário.....	48
Quadro 9: Comparação entre os métodos de armazenamento com a Consulta 4B em ambiente multi-usuário.....	48
Quadro 10: Proposta de uso dos métodos de armazenamento em PostgreSQL.....	51

## LISTA DE SIGLAS

BLOB	<i>Binary Large Object</i>
BMP	<i>Bitmap</i>
CDROM	<i>Compact Disc Read Only Memory</i>
DDL	<i>Data Definition Language</i>
FGDC	<i>Federal Geographic Data Committee</i>
GIF	<i>Graphics Interchange Format</i>
JPG	<i>Joint Photographic Experts Group</i>
MPEG	<i>Motion Picture Expert Group</i>
OLTP	<i>On-Line Transaction Processing</i>
RAM	<i>Random Access Memory</i>
SGBDM	<i>Sistema Gerenciador de Banco de Dados Multimidia</i>
SGBD	<i>Sistema Gerenciador de Banco de Dados</i>
SOAP	<i>Simple Object Access Protocol</i>
SQL	<i>Structured Query Language</i>
SUB	<i>System Under Benchmark</i>
TPC	<i>Transaction Processing Performance Council</i>
XML	<i>Extensible Markup Language</i>

## RESUMO

As aplicações multimídia, que necessitam de gerenciamento de objetos, fizeram com que o desempenho, nos Sistemas Gerenciadores de Banco de Dados - SGBDs, assumisse um papel importante para o funcionamento correto dessas aplicações. Assim esse trabalho apresenta uma avaliação no desempenho dos métodos de recuperação dos objetos multimídia, em uma base de dados com dez mil objetos cadastrados, onde cada um dos objetos possui 600 *kilobytes* de tamanho. Para essa avaliação foram utilizados três métodos possíveis no SGBD PostgreSQL, o método de referência externa, armazenamento por dados não interpretados (*bytea*) e por objetos (*oid*). A avaliação foi realizada com quatro consultas distintas e em dois ambientes diferentes, mono-usuário e multi-usuário. Os resultados dessa avaliação mostraram que o desempenho nos métodos de armazenamento, os quais guardam os objetos dentro da base de dados, não foram satisfatórios. O tempo de resposta de um dos métodos foi de 3,5 minutos, em um ambiente mono-usuário. Além desse resultado, concluiu-se também que o armazenamento através de objeto, com campo de identificação de objeto, possui o pior tempo de recuperação para vários objetos, entre os métodos comparados.



## ABSTRACT

The multimedia applications, which require object management, made the performance in the Databases Management Systems (DMS) take a relevant role for the correct functioning of such applications. Thus, this work presents an evaluation of the retrieval methods for multimedia objects performance, in a databases comprising 10,000 objects, in which each object is 600kb. Three possible methods were used in SGBD PostgreSQL for this evaluation, the external reference method, non-interpreted data storage (*bytea*) and objects (*oid*). Four distinct queries were carried out in two different environments, mono and multi-user. The results of this evaluation showed that the performance of storage methods that keep objects within the databases was not satisfactory. One of these methods response time was 3,5 minutes, in a mono-user environment. Besides this result, it was also concluded that the storage through object, with object identification field, presents the worst retrieval time for several objects, considered the methods compared.

## 1. INTRODUÇÃO

Com a quantidade de objetos multimídia crescendo na maioria das aplicações, uma das questões importantes a ser tratada é o método de gerenciamento desses objetos. Para isso, a utilização dos Sistemas Gerenciadores de Banco de Dados (SGBDs) para armazenar e gerenciar tais objetos é crucial. Para possibilitar o armazenamento e gerenciamento dos objetos multimídia em SGBDs, são utilizadas quatro formas distintas, cada qual com características e utilizações distintas.

Essas formas de armazenamento são classificadas como, armazenamento por referência, armazenamento por dados não interpretados, armazenamento através de funções externas e armazenamento através de orientação a objetos. (VAZ, 2000; FRANÇA, 2005)

A primeira forma armazena apenas a localização do objeto multimídia no sistema operacional dentro do campo destinado a imagem, a segunda forma armazena os objetos em forma de uma cadeia binária, a terceira forma utiliza funções externas aos SGBDs para prover o armazenamento, e a quarta forma permite a criação de tipos de objetos e utilização de herança entre os mesmos.

Com a utilização dos SGBDs em aplicações multimídia, eles passaram a ser os principais fatores para o desempenho das mesmas. Um bom desempenho na recuperação dos objetos multimídia influi diretamente no desempenho da aplicação, pois os Sistemas Gerenciadores de Banco de Dados podem retardar o retorno dos objetos multimídia para a aplicação, proporcionando assim uma demora geral no sistema.

Devido ao desempenho proporcionado pelos SGBDs, nas buscas de objetos multimídia, influenciar diretamente nas aplicações que o utilizam, faz-se necessário conhecer o impacto de cada um dos tipos de armazenamento durante a consulta dos objetos. Para isso se faz necessária a realização de avaliações de desempenho sobre os mesmos.

Para avaliar o desempenho dos sistemas informatizados pode-se utilizar estruturas de *Benchmark*, as quais são utilizadas para fornecer padrões e

---

metodologias de avaliação, mantendo toda a análise dentro das mesmas características. (VIEIRA, DURÃES e MADEIRA, 2005)

Tendo em vista o exposto, essa dissertação possui como objetivo geral a avaliação do desempenho dos principais métodos de armazenamento de objetos multimídia, com base em *Benchmarks* descritos na literatura. Essa avaliação teve como estudo de caso no SGBD PostgreSQL (POSTGRESQL, 2005), o qual receberá uma base de dados com oito tabelas sendo que uma delas possui dez mil imagens cadastradas com 600 *kilobytes* cada uma.

A avaliação será composta por quatro consultas, onde na primeira foi recuperado um dentre os dez mil objetos, na segunda foram recuperados cem objetos dentre dez mil, na terceira um objeto foi recuperado através de junções entre tabelas e a quarta consulta foi recuperado vinte dentre dez mil objetos cadastrados. A quarta consulta foi criada para simular os sistemas de vendas eletrônicas que exibem seus produtos de vinte em vinte.

Essas consultas foram aplicadas em dois ambientes distintos, o primeiro ambiente foi o mono-usuário onde apenas um usuário consulta o SGBD, o segundo ambiente foi o multi-usuário onde trinta usuários realizaram ao mesmo tempo as consultas no SGBD.

Com a realização dessa análise pretendeu-se conhecer a influência no desempenho das consultas de cada uma das formas de armazenamento de objetos multimídia disponibilizadas pelo PostgreSQL. Através desses dados pode-se indicar quando cada uma das formas de armazenamento deve ser utilizada, além de apontar possíveis problemas de desempenho com a utilização de alguma delas. Assim, pode-se também, abrir precedente, para uma reestruturação da implementação das formas de armazenamento pelo SGBD PostgreSQL.

Para apresentar esta análise, esta dissertação foi estruturada como segue. Além deste capítulo inicial têm-se mais quatro capítulos. No Capítulo 2 é conceituado o termo mídia, apresentando seus principais componentes, além de abordar os principais conceitos utilizados em banco de dados multimídia e as formas de armazenamento dos mesmos. No Capítulo 3 é definido o conceito de *benchmark*, além de ser exposto as metodologias, encontradas na literatura,

utilizadas pelos *benchmarks* inerentes a avaliação de SGBD. No Capítulo 4 é apresentada a avaliação de recuperação de objetos multimídia realizada no PostgreSQL. No Capítulo 5 são apresentadas as conclusões e perspectivas de trabalhos futuros.

## **2. SISTEMAS DE BANCO DE DADOS PARA APLICAÇÕES MULTIMÍDIA**

A crescente utilização de objetos multimídia, em sistemas informatizados, fez com que surgisse a necessidade dos Sistemas Gerenciadores de Banco de Dados (SGBDs) em tratar tais objetos. Sendo assim, técnicas de indexação e armazenamento são utilizadas para possibilitar um melhor aproveitamento dos SGBDs, no tratamento de objetos multimídia.

A indexação dos objetos multimídia resume-se em descrevê-los antes do armazenamento, onde as características necessárias para se obter sua descrição, em um determinado contexto, são atribuídas ao mesmo. Através dessa descrição, indexação, atribuída ao objeto é que são baseadas as consultas dos mesmos.

As formas de armazenamento dos objetos multimídia nos SGBDs influenciam diretamente nas condições de pesquisas permitidas e no desempenho das mesmas. Assim, faz-se necessário esclarecer suas características e possibilidades de pesquisa.

Para descrever as técnicas e métodos usados nos SGBDs para aplicações multimídia, esse capítulo está dividido como segue. Na Seção 2.1 são expostos os conceitos e características de objetos multimídia. Na Seção 2.2 são abordadas as formas de gerenciamento desses objetos. Na Seção 2.3 são apresentados o conceito de metadados e a indexação dos objetos através dos mesmos. Na Seção 2.4 são expostos os tipos de pesquisa e recuperação dos objetos multimídia, com suas respectivas características. Na Seção 2.5 são apresentadas as formas de armazenamento dos objetos nos SGBDs.

### **2.1. MULTIMÍDIA E OS TIPOS DE OBJETOS**

O termo multimídia está amplamente difundido e empregado em diversas tecnologias, como sistemas, jogos eletrônicos, hardwares de computadores, entre outros que processam, armazenam e ou exibem mídias eletrônicas.

De acordo com CHRISTEL, M. citado por JAIMES *et al* (2005), multimídia pode ser definida como sendo a combinação de diferentes tipos de

mídia, tais como texto, imagens, vídeo, animação e som, dentro de uma aplicação que resultará em uma forma de comunicação. Como, por exemplo, um comercial de televisão, onde estão presentes os mais diversos tipos de mídias.

MA, W. citado por JAIMES *et al* (2005) define texto, imagem e áudio como sendo tipos de mídia e as animações e vídeos como sendo objetos multimídia, pois esses reúnem mais de um tipo de mídia. Com isso, considera páginas na internet como sendo objetos multimídia, já que apresentam textos, imagens, sons e animações.

O termo mídia possui alguns significados, os quais são descritos a seguir: (VAZ apud, STEINMETZ, 2000) (ÖZDEN, RASTOGI e SILBERSCHATZ, 1997)

- *Mídia de Percepção*: Esse tipo de mídia refere-se ao formato na qual a informação é apresentada para seus observadores. A imagem, o som e o texto são exemplos desse tipo de mídia. Essas mídias podem ser divididas em contínuas e não contínuas. As contínuas são dependentes do tempo. O som e o vídeo são classificados como contínuas. As mídias não contínuas independem do tempo. A imagem é classificada como uma mídia de percepção não contínua. No decorrer dessa dissertação as mídias de percepção também serão tratadas como objetos multimídia.
- *Mídia de Representação*: Esse tipo de mídia refere-se ao formato em que elas são representadas computacionalmente. A mídia de percepção imagem pode ser representada nos formatos GIF, JPEG, BMP, entre outros.
- *Mídia de Apresentação*: Esse tipo de mídia abrange os dispositivos que promovem a entrada e saídas de dados. Os monitores, caixas de som e impressoras, os quais promovem a saída de dados e informação, e os teclados, mouse e scanners, que promovem a entrada de dados, são mídias de apresentação.

- *Mídia de Armazenamento*: Esse tipo de mídia abrange os dispositivos que promovem o armazenamento de dados. As fitas, discos rígidos e CDROM são algumas das mídias de armazenamento.
- *Mídia de Transmissão*: Esse tipo de mídia refere-se aos meios físicos de transmissão de dados. As redes de computadores são consideradas mídias de transmissão.

Em relação às mídias de percepção, essas possuem diversos formatos de dados. Na seqüência, serão abordados os principais formatos (CERVI, SILVA e PAVAN, 2005):

- *Texto*: Caracteriza-se por ser a mídia mais utilizada para interação entre sistemas informatizados e os usuários. Essa mídia é representada por uma cadeia de caracteres. Quando utilizada com outros tipos de mídia podem tornar-se descritores das mesmas.
- *Gráfico*: Caracteriza-se por representar desenhos, a partir de outros dados e/ou informações. O gráfico é formado por linhas, curvas, círculos, retângulos, entre outros. Também possui atributos que definem cor, textura, entre outras características.
- *Imagem*: Corresponde a desenho ou fotografia que está codificada em forma de *bitmap* (mapa de *bits*). O mapa de *bits* é uma matriz de  $n \times m$ , onde cada célula (*pixel*) possui um valor para descrever seu conteúdo. As imagens são qualificadas de acordo com sua resolução, que é a quantidade de *pixel* em uma polegada quadrada (*dpi*). Quando uma imagem possui resolução de 300 *dpi*, significa que essa imagem possui 300 *pixel* em uma polegada quadrada. Com o intuito de diminuir o tamanho físico ocupado pelas imagens gravadas em disco, foram criados métodos de compactação. Esses métodos são cálculos matemáticos que procuram reduzir o número de *pixels* armazenados, sem alterar suas características.

- *Áudio*: É um dado registrado a partir de vibrações do ar em uma cadeia de *bits* de maneira digitalizada. Os microfones são os responsáveis por captar essas vibrações e transformá-las em sinais elétricos. Esses sinais podem ser enviados, por exemplo, para os computadores, onde são convertidos em uma cadeia de *bits*, formando assim o áudio digital. O áudio pode ser classificado em estruturado, onde é caracterizado por uma seqüência de dados de áudio juntamente com seus atributos de notas, tom, duração, entre outras características.
- *Animação*: São seqüências de imagens e/ou desenhos disponibilizados em um espaço temporal, as quais podem ou não utilizar áudio.
- *Vídeo*: Um conjunto de dados fotográficos disponibilizados em um espaço temporal. Os dados fotográficos são chamados de *frames*, que são imagens estáticas. No movimento de vídeo, *frames* são exibidos em um espaço de tempo. Dependendo do tipo de vídeo, em um segundo podem ser exibidos 33 *frames*.

A multimídia alterou a maneira como os problemas e questões são expostos e interpretados. Isso ocorreu, devido à riqueza de informação quando utilizados textos, imagens e animações nas aplicações, tornando-as mais atrativas quando comparada com as que apresentam apenas um tipo de mídia.

A expansão da multimídia deve-se ao avanço tecnológico que proporcionou um ambiente favorável para sua utilização. A criação de redes de alta velocidade, mídias de armazenamento de alta capacidade, compressões nas mídias de percepção, computadores pessoais com boa capacidade de processamento e armazenamento, além de ferramentas para manipulação dos mais diversos tipos de mídia são alguns dos avanços. Assim, diversas áreas utilizam a multimídia como forma de melhorar a apresentação da informação ao seu usuário final.

Na educação, é freqüente a utilização da multimídia na criação de materiais didáticos. Esses materiais vão desde fitas de vídeos e áudios até sistemas de educação à distância. Esses sistemas têm com uma das



funcionalidades repassar as informações de forma clara e objetiva aos usuários, tornando o ambiente interessante e atrativo. Assim, destaca-se o uso diversos objetos multimídia na construção de aulas, materiais didáticos, exercícios e provas. (SILVA, 2004).

Na medicina, existem sistemas que cadastram simultaneamente dados convencionais, como nome e endereço (entre outros) dos pacientes e dados não convencionais, como imagens de raios-X, vídeos de cirurgias, resultados de exames. Com isso aumenta-se o detalhamento da informação sobre os pacientes, facilitando diagnósticos e futuras intervenções médicas.

Outra área que merece destaque na utilização da multimídia é o comércio eletrônico, o qual é realizado através da internet. Cada produto possui várias imagens, dos mais diversos ângulos, com o objetivo de atrair o comprador para suas principais características físicas. O tamanho, a cor e o formato são algumas dessas características.

Com a crescente utilização dessas aplicações que fazem uso de objetos multimídia, um dos fatores importantes é o uso de banco de dados multimídia.

O uso desses bancos de dados é tão inerente em algumas aplicações multimídia, como por exemplo, em sistemas imobiliários, quanto os bancos de dados convencionais em aplicações de cadastros de funcionários. Isso se deve à necessidade de se manter a consistência e a integridade dos objetos multimídia durante toda a existência das aplicações. Necessidade essa que aumenta com a utilização de sistemas que realizam com frequência as operações de exclusão, inserção e edição de objetos multimídia.

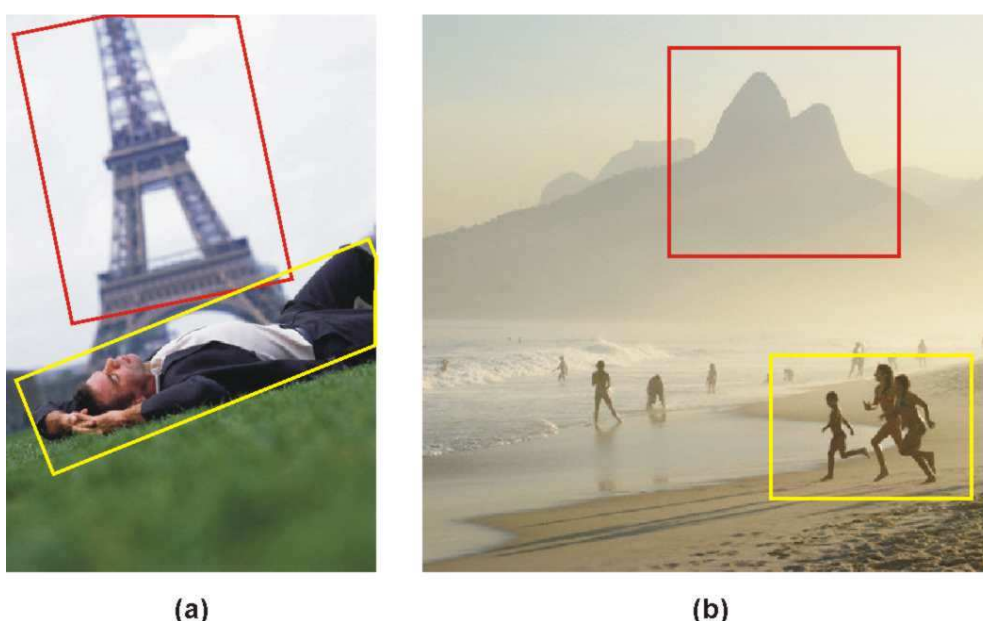
## 2.2. GERENCIAMENTO DE OBJETOS MULTIMÍDIA

Os sistemas de bancos de dados para aplicações multimídia possuem as mesmas características dos Sistemas Gerenciadores de Banco de Dados (SGBD) não convencionais. Devido às diferenças encontradas entre os objetos multimídia, os sistemas gerenciadores devem possuir mais capacidade do que quando gerenciam dados convencionais. (GHAFOOR, 1995)

Duas principais características dos Sistemas de Banco de Dados para aplicações Multimídia (SGBDM) são expostas a seguir: (FRANÇA, 2005)

1. Prover consultas baseadas em conteúdo dos objetos.
2. Gerenciar dados contínuos, como áudio e vídeo, que possuem o fator tempo, o qual resulta na necessidade de gerenciar o sincronismo.

Nas consultas baseadas no conteúdo dos objetos, o sistema deverá ser capaz de fazer a distinção entre as características dos objetos. Ao solicitar ao sistema de banco de dados, a(s) imagem(s) de pessoa(s) em Paris, o mesmo deve ser capaz de distinguir o conteúdo das imagens como é observado na Figura 1.



**Figura 1: Distinção entre conteúdo dos objetos.**

Na Figura 1 (a) é observada a Torre Eiffel, a qual significa que é uma imagem de Paris e também há uma pessoa, satisfazendo aos dois critérios da consulta. Ou seja, o fato de ser imagem de Paris e existir uma pessoa na mesma.

Na Figura 1 (b) são encontradas pessoas, mas ao invés de se encontrar alguma referência sobre Paris é encontrado o Pão de Açúcar, na Cidade do Rio de Janeiro. Essa imagem é descartada, pois não satisfaz os critérios da consulta.

No gerenciamento de dados contínuos deve-se levar em consideração que os objetos serão, além de encontrados, disponibilizados à medida que

forem recuperados. Com isso, o sincronismo de recuperação e de exibição dos mesmos deve ser feito, tornando a visualização dos dados em tempo real.

O sincronismo da mídia de dados contínuos torna-se relevante devido a sua exibição ser baseada na quantidade de quadros (*frames*) por segundo. O sistema deve recuperar os quadros armazenados com a mesma velocidade necessária para exibição. (ÖZDEN, RASTOGI e SILBERSCHATZ, 1997)

ÖZDEN, RASTOGI e SILBERSCHATZ (1997) apontam ainda, a necessidade de os SGBDM possuírem capacidade para armazenar grandes quantidades de dados complexos. Um objeto de vídeo, por exemplo, com uma hora e quarenta minutos de duração e com compressão em MPEG-I, pode requerer 1.25 gigabytes de espaço.

### 2.3. INDEXAÇÃO DE OBJETOS MULTIMÍDIA ATRAVÉS DE METADADOS

Segundo VAZ (2000), “A indexação permite identificar e representar o conteúdo ou o assunto de um documento de forma a caracterizá-lo. Um documento deve ser indexado tendo-se em mente não apenas o seu armazenamento, mas principalmente sua futura busca e recuperação”.

Sendo assim, a indexação tende a ser um dos requisitos para a pesquisa e recuperação da informação multimídia. Visando a estruturação durante a indexação de objetos multimídia pode ser utilizado o conceito de metadados (MILSTEAD,1999; VAZ,2000), como sendo dados que descrevem outros dados, os quais podem ser utilizados como atributos para descrever diversos tipos de dados.

Os metadados, quando associados aos objetos são considerados descritores dos mesmos, pois aumentam o conhecimento sobre seu conteúdo e suas características. (MILSTEAD,1999).



Metadado	Valor
País:	<i>França</i>
Data de criação:	<i>22/12/2005</i>
Tipo do objeto:	<i>imagem</i>
Resolução:	<i>72 dpi</i>
Direitos:	<i>público</i>

**Figura 2: Metadados de Objeto Multimídia**

A Figura 2 exemplifica a utilização de metadados para a indexação de objetos multimídia. Para cada tipo de metadados, valores condizentes com a imagem são atribuídos. Os metadados podem descrever características semânticas dos objetos como, por exemplo, o tipo de metadados *país*, que descreve o país que a foto está representando.

Os metadados também podem descrever características não semânticas, que são as características que não podem ser extraídas ao observar o objeto, como por exemplo, a data de criação.

Os metadados podem ser classificados em vários tipos, de acordo com sua finalidade de descrição. A seguir são descritos alguns tipos (NISO, 2004; VAZ, apud BÖHMS, 2000):

#### **Tipos gerais de metadados:**

- *Metadados descritivos:* são os metadados que vão descrever o próprio objeto, procurando identificá-lo da melhor maneira possível. Os metadados observados na Figura 2 são considerados descritivos.
- *Metadados estruturais:* esses metadados definem a relação entre os objetos e/ou a relação entre os metadados. Os metadados que determinam quais objetos fazem parte de um objeto composto, e os metadados que determinam quais metadados podem ser usados a um determinado objeto, são exemplos de metadados estruturais.
- *Metadados administrativos:* são os metadados que auxiliam na administração do sistema como um todo, pois são eles que determinam quando e como os metadados e objetos estarão

disponíveis. Os metadados que determinam quais são os tipos de usuários que possuem permissão a um determinado objeto multimídia, são exemplos de metadados administrativos.

**Tipos de metadados voltados para objetos multimídia:**

- *Metadados para representação de tipos de mídia:* São responsáveis por descrever as necessidades técnicas para a representação dos objetos multimídia. As mídias de um mesmo grupo, em geral, possuem várias maneiras de serem digitalizadas e posteriormente armazenadas. As mídias do tipo imagem podem necessitar de diferentes algoritmos para sua representação, dependendo do formato de compressão que foi utilizado no momento da digitalização. Esses metadados fornecem informações suficientes para que o objeto armazenado seja tratado de forma correta no momento de sua exibição. Os metadados que armazenam os *plugins* necessários para a exibição das mídias, são exemplos de metadados para representação de tipos de mídia.
- *Metadados para classificação de conteúdo:* São responsáveis pelo fornecimento de uma classificação do objeto de acordo com seu conteúdo. Essa classificação vai depender dos objetivos e regras implementados pelo sistema. Como por exemplo, em imagens de estrelas obtidas através de telescópios, onde os metadados de classificação poderiam ser o tipo de telescópio utilizado, o tipo da lente utilizada, entre outros atributos que classificam as imagens de acordo com as ferramentas utilizadas.
- *Metadados para composição de documentos:* São utilizados para prover um relacionamento entre os objetos multimídia utilizados na composição de um documento. Alguns documentos multimídia como páginas de internet são compostos por diversas mídias, as quais podem estar presentes em mais de uma página. Esses metadados promovem um conhecimento da interligação das mídias e dos documentos.

---

O uso de metadados na indexação de objetos multimídia facilita a organização dos objetos, a pesquisa de informações relevantes e a interoperabilidade entre sistemas. Além disso, eles podem promover uma identificação digital no objeto, pois muitos padrões de metadados possuem identificadores únicos para cada um dos seus objetos. (NISO, 2004)

Apesar das facilidades, existem cuidados que devem ser empregados quando se gerencia o uso de metadados. O vocabulário utilizado na atribuição de valores aos metadados é um dos pontos que se deve observar, pois os usuários no momento da atribuição desses valores podem utilizar gírias regionais, bem como cometer erros de grafia, o que compromete a pesquisa e recuperação dos objetos. Sendo assim, quando necessário deve-se usar vocabulário controlado para garantir a padronização dos valores.

Nem todos os metadados necessitam usar vocabulário controlado, pois a abrangência de valores pode ser indefinida, sendo inviável a construção do vocabulário. O tipo de metadados *título*, que pode descrever o título de livros digitalizados, é um tipo que não necessita de vocabulário controlado.

Para organizar e gerenciar o uso dos metadados e homogeneizar as descrições de objetos em sistemas de banco de dados para as aplicações, que possuem o mesmo propósito ao armazenar as mídias, utilizam-se os padrões de metadados.

Esses padrões são conjuntos de metadados organizados com o intuito de descrever um escopo específico de dados. De acordo com NISO (2004), padrões de metadados identificam um conjunto de descrições dos objetos através de uma estrutura bem definida, a qual é composta por regras e vocabulários controlados.

Atualmente, existem diversos padrões de metadados utilizados em diferentes recursos, como por exemplo, o Padrão *FGDC (Federal Geographic Data Committee)* (FGDC, 2005) que é utilizado para descrever objetos geoespaciais e o Padrão *Dublin Core* (DCMI, 2005) que é considerado um padrão abrangente, pois possui metadados que podem descrever as características básicas de vários tipos de objetos.

O processo de indexação dos objetos multimídia, através de metadados, seja organizado conforme padrões ou não, pode ser realizado de duas formas: manual ou automática.

Na forma manual, a indexação se limita aos conhecimentos do usuário, ou seja, os valores dos metadados são atribuídos de acordo com o conceito que o usuário tem sobre o objeto. Essa forma de indexação pode gerar metadados com valores inconsistentes, pois a escolha do valor que representará o tipo de metadados depende dos critérios de interpretação que os usuários desenvolvem sobre os objetos, que nem sempre serão os mesmos entre os usuários.

A forma automática consiste em algoritmos que extraem dos objetos informações como resolução de imagem, frequência do som, entre outras. Na utilização da forma automática é necessária grande capacidade de processamento, devido à complexidade imposta pelos algoritmos de extração de características, ou seja, os valores dos metadados.

Outro ponto de relevante valor é a necessidade de padronização de extração de características entre os diversos tipos de mídias, a qual permite que todas as mídias tenham o mesmo critério de indexação.

#### 2.4. PESQUISA E RECUPERAÇÃO DE OBJETOS MULTIMÍDIA

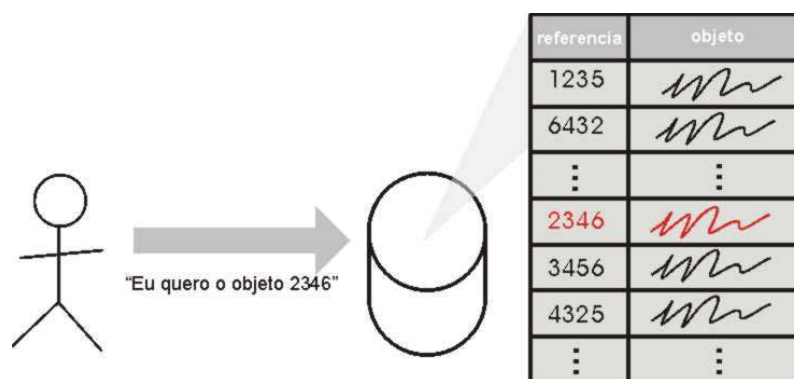
Em sistemas de banco de dados multimídia, a pesquisa e a recuperação de objetos são discutidas na literatura, posto que os objetos possuem mais características do que pode se representar através de atributos. De acordo com ÖZDEN, RASTOGI e SILBERSCHATZ (1997), muitos sistemas necessitam que a pesquisa e a recuperação sejam realizadas baseando-se no conteúdo das mídias, devido à riqueza de características que ele possui quando comparado com seus atributos.

Entretanto, diversos sistemas comerciais promovem de forma satisfatória a pesquisa e recuperação das mídias através de seus metadados. O sistema *WEB* da Corbis (CORBIS, 2005), que disponibiliza uma base de dados de imagens, é um exemplo desse tipo de pesquisa.

Dentre os diversos tipos de pesquisas de objetos multimídias destacam-se quatro métodos: (VAZ, 2000)

1. Recuperação através de um identificador
2. Recuperação por sentenças condicionais
3. Recuperação por similaridade
4. Recuperação semântica

No primeiro tipo o objeto é recuperado através de um identificador que geralmente é o índice da tabela a qual o objeto pertence. Nessa recuperação, o usuário ou o sistema já conhece o valor do identificador do objeto, como mostra a Figura 3.

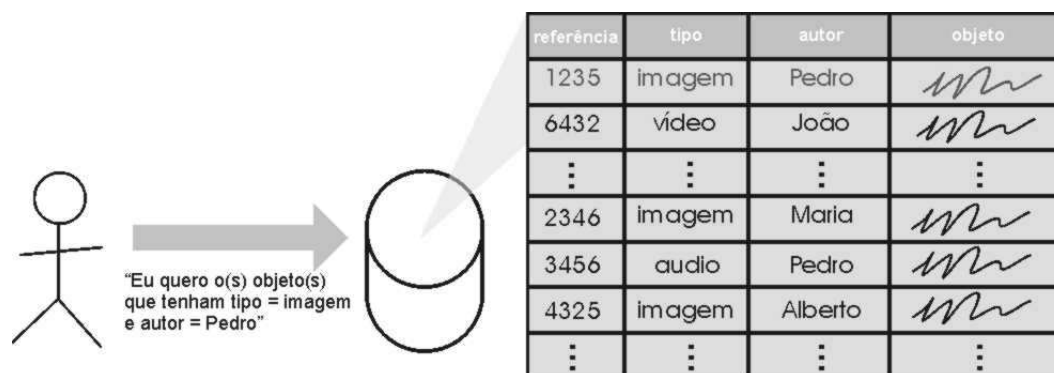


**Figura 3: Recuperação através de um identificador.**

Na Figura 3, o usuário ou o sistema solicita ao banco de dados o objeto cuja referência é "2346" e o obtém em uma localização direta por índices. Esse método de recuperação é favorecido pela utilização dos índices, os quais permitem localizar o objeto desejado examinando apenas um pequeno conjunto dos registros. (GARCIA-MOLINA, ULLMAN e WIDOM, 2001)

No segundo tipo os objetos a serem recuperados precisam obedecer às regras condicionais impostas na consulta. Essas regras podem variar desde pequenas operações lógicas como igualdade, até condições de *joins* entre tabelas.

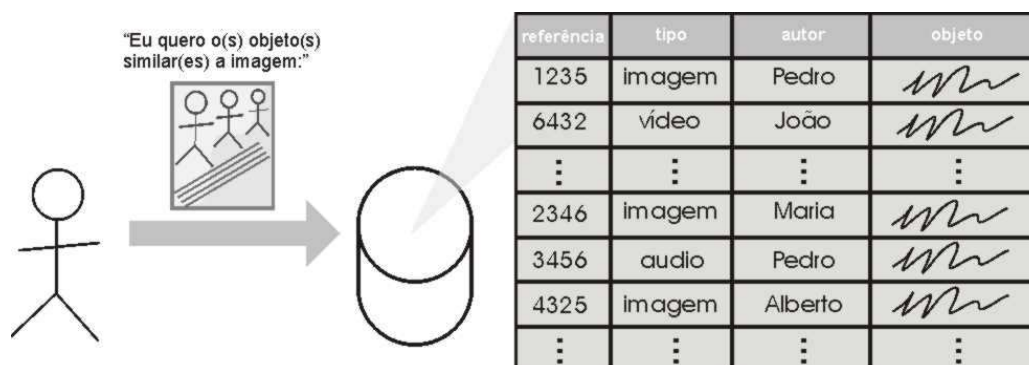




**Figura 4: Recuperação através de sentenças condicionais.**

A Figura 4 esboça um usuário ou sistema solicitando um objeto através de uma consulta que será recuperada de acordo com a sentença condicional “tipo = imagem e autor = Pedro”. Com isso, o único objeto que satisfaz essa condição de pesquisa é o objeto com referência “1235”, sendo o objeto retorno da consulta.

No terceiro tipo a pesquisa também pode ser realizada através de comparações com sentenças condicionais, entretanto nem todas as sentenças devem ser satisfeitas. O que define quais objetos serão recuperados na pesquisa é o grau de similaridade imposto para a consulta. Os objetos que apresentarem similaridade igual ou superior ao objeto especificado são retornados.

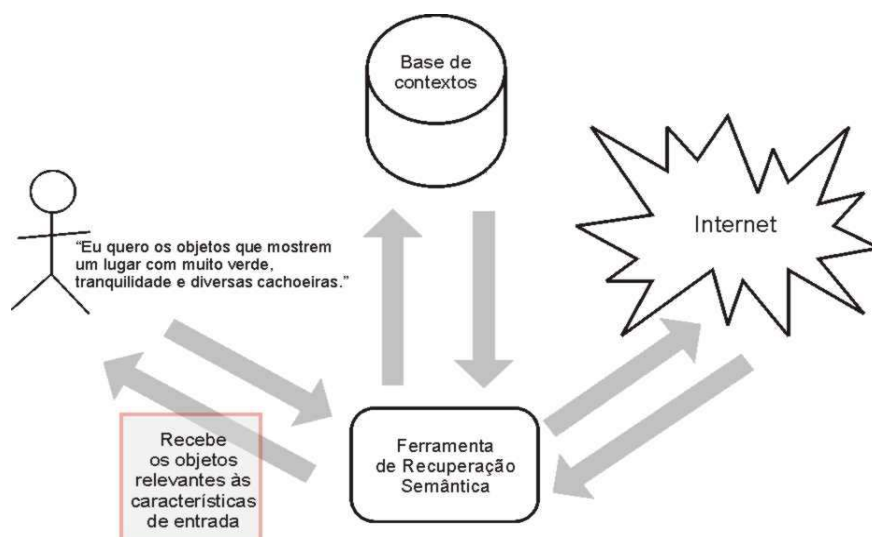


**Figura 5: Recuperação por similaridade.**

A Figura 5 representa uma pesquisa por similaridade levando em consideração um objeto como entrada.

Nesse caso, o usuário fornece uma imagem como critério de pesquisa e, baseado em um grau de similaridade, os objetos da base de dados serão escolhidos ou não, para fazer parte do resultado.

Na recuperação semântica os objetos são recuperados através de conceitos e não por dados colocados em expressões condicionais. Essa forma de recuperação tende a possuir um melhor nível de abstração uma vez que os objetos serão recuperados pela sua representação semântica.



**Figura 6: Arquitetura de recuperação semântica (LOH, WIVES e FRAINER, 1999).**

A Figura 6 demonstra uma arquitetura de recuperação semântica para objetos da internet (LOH, WIVES e FRAINER, 1999), onde o usuário faz a requisição dos objetos, através de frases ou diretamente pelos conceitos.

O funcionamento dessa arquitetura inicia-se com a ferramenta de recuperação procurando, na base de contextos, as peculiaridades de cada palavra fornecida pelo usuário.

Em seguida é realizada uma pesquisa na internet com base em todos os conceitos identificados, os objetos encontrados nas pesquisas são comparados com os conceitos identificados através de operadores de conjunções e disjunções dos elementos. Após a comparação, os documentos são ordenados de acordo com seu grau de relevância e entregues ao usuário.

Os dois primeiros tipos, ou seja, recuperação através de um identificador e recuperação por sentenças condicionais, são tipos de recuperação capazes de serem implementados com o auxílio de metadados.

No primeiro tipo o objeto pode possuir um tipo de metadados que é seu identificador único, nesse caso o usuário entrará com o valor desse tipo e será recuperado o objeto que possuir exatamente esse valor.

No segundo método, o usuário escolhe os valores para os diversos metadados que descrevem os objetos e especifica a condição existente entre eles. A recuperação se dá pela satisfação da condição imposta aos valores de cada tipo de metadados escolhido.

Os outros dois tipos de pesquisas são realizados através de algoritmos mais complexos, não dependendo apenas de comparações de valores de metadados.

## 2.5. ARMAZENAMENTO DE OBJETOS MULTIMÍDIA

Manter a consistência e integridade dos dados em um SGBD é um dos princípios fundamentais para um resultado satisfatório durante a pesquisa e/ou recuperação. A maneira e a estrutura como os dados são armazenados no banco de dados influencia na consistência e integridade dos dados.

Em relação aos objetos multimídia isso se torna ainda mais relevante, tendo em vista que esses dados possuem grandes quantidades de *bytes*, podendo chegar a ter vários *gigabytes* de tamanho.

Devido a essas características, os SGBDs possuem diferentes formas de armazenar dados multimídia, assim como todos os tipos de dados não convencionais. A seguir são abordados esses mecanismos de armazenamento (VAZ, 2000;FRANÇA, 2005):

- *Referências externas:* Nessa forma de armazenamento o banco de dados possui uma referência para o arquivo que fica armazenado externamente a ele. Geralmente metadados fazem a descrição do objeto e um campo possui o endereço no qual está armazenado o objeto. Devido ao armazenamento externo, o banco de dados não possui controle sobre o objeto, com isso ele não consegue prover consistência e integridade dos dados.
- *Armazenamento de dados multimídia não interpretados (campo BLOB):* Essa forma de armazenamento utiliza o campo do banco de dados com o tipo *BLOB* para o armazenamento dos objetos multimídia. Nesse campo os dados são armazenados como uma seqüência de *bytes* não interpretados, sendo impossível extrair as

características semânticas dos objetos, diretamente do campo. Nesse caso o SGBD se torna apenas um sistema de armazenamento dos dados, não sendo possível interpretar a mídia diretamente no banco de dados. Devido a isso as aplicações precisam implementar formas de abstração dos dados e técnicas de exibição. Para as mídias contínuas, o uso desse campo não permite a capacidade de exibir a mídia, à medida que ela é recuperada, sendo necessário extraí-la totalmente para em seguida iniciar sua exibição. De acordo com GARCIA-MOLINA, ULLMAN e WIDOM (2001), uma das maneiras de proporcionar a recuperação, em tempo real nesse tipo de campo, é implementando uma estrutura de índice, fazendo com que cada segundo possua um índice dentro do *BLOB*, visto que, nesse tipo de armazenamento, o SGBD possui controle sobre os objetos armazenados, o que permite a gestão da consistência e integridade dos dados.

- *Funções externas:* As funções externas são usadas para suprir a falta de recursos para manipular dados multimídia nos SGBDs. Alguns bancos de dados implementam ou utilizam funções externas para permitir extensões em linguagens de consulta e representação de objetos multimídia.
- *Sistemas orientados a objetos ou relacionais estendidos:* Nos sistemas orientados a objetos o programador pode definir tipos de dados e referenciá-los para as aplicações, além da construção hierárquica de tipos de dados e os relacionamentos entre as classes. Pela possibilidade de criação de tipos, essa alternativa de armazenamento mostra-se uma das mais adequadas. Contudo as mídias contínuas ainda permanecem com o problema da falta de gerenciamento de acesso em tempo real.

Cada uma das formas de armazenamento expostas anteriormente possui características próprias, sendo possível destacar seus pontos positivos e negativos.

A seguir é apresentado o Quadro 1 que mostra um comparativo entre as formas de armazenamento:

	Referência Externa	Campos <i>BLOB</i>	Funções Externas	Orientação a Objetos
Consistência e integridade	não	sim	sim	sim
Armazenamento interno	não	sim	sim	sim
Acesso em tempo real	sim	não	não	não
Pesquisa através da semântica	sim	não	não	sim

**Quadro 1: Comparativo entre as formas de armazenamento.**

A Quadro 1 mostra de forma comparativa as principais características de cada uma das formas de armazenamento expostas. A referência externa é a única forma que possibilita acesso em tempo real ao objeto multimídia, isso se deve ao fato do objeto ficar fora do SGBD. Essa característica faz com que esse tipo de armazenamento seja apropriado para aplicações que necessitam disponibilizar os dados dos objetos em partes, de acordo com a necessidade do usuário. Um exemplo desse tipo de aplicação são as aplicações baseadas em *stream*, as quais vão produzindo os objetos para os usuários a medida que esse objeto vai sendo lido. A desvantagem de se utilizar esse método é que ao armazenar o objeto multimídia dentro de pastas no sistema operacional e fora do SGBD, acarreta em alguns problemas como:

- Ao realizar *backup* da base de dados é necessário procurar as pastas onde estão armazenados os objetos multimídia, e copiá-las de forma manual, ou implementar uma rotina para realizar essa cópia.
- Ao excluir o registro da base de dados que contenha a referência para o objeto multimídia, não exclui automaticamente o objeto.
- Quando várias pessoas acessam o servidor onde se encontra o SGBD, aumenta o grau de vulnerabilidade dos objetos armazenados nas pastas.

Com base nesses problemas, a referência externa é considerada uma forma de armazenamento inconsistente, onde registros podem referenciar objetos inexistentes e objetos podem estar sem registros que os referencie.

Os armazenamentos por *BLOB* e funções externas possuem aparentemente as mesmas características como mostra a Quadro 1. Esses dois métodos armazenam os objetos internamente ao SGBD, o que garante a integridade dos dados e neutraliza os problemas apresentados no método da referência externa. Entretanto, devido ao tipo do campo utilizado para realizar o armazenamento, só é possível ler o objeto e suas características semânticas quando ele estiver totalmente recuperado do SGBD, o que impossibilita o acesso em tempo real ao objeto.

Esses dois tipos de armazenamento podem ser utilizados em aplicações que não exigem acesso aos objetos em tempo real, e nas aplicações que utilizam metadados para indexar os objetos, pois com metadados, não é preciso ler o objeto para descobrir suas características.

O armazenamento de objetos multimídia através da orientação a objetos em geral possui as mesmas características dos dois últimos métodos abordados. No entanto, essas características dependem da maneira como o SGBD trata a orientação a objeto, onde pode variar principalmente sua característica de disponibilizar a semântica dos objetos dentro do próprio SGBD.

Além das características abordadas, os métodos de armazenamento possuem diferenças em relação ao desempenho, quando consultas que buscam objetos multimídia são executadas. Dessa forma, o Capítulo 3 apresenta a definição e as metodologias para a realização da análise de desempenho em banco de dados, para possibilitar no Capítulo 4, a construção de uma análise de desempenho nas formas de armazenamento.

### 3. BENCHMARK - AVALIAÇÃO DE DESEMPENHO EM BANCO DE DADOS

#### 3.1. DEFINIÇÃO

As avaliações em sistemas informatizados são realizadas com a utilização de *benchmarks* (VIEIRA, DURÃES e MADEIRA, 2005), os quais proporcionam padrões na avaliação de desempenho de sistemas. VIEIRA, DURÃES e MADEIRA consideram que os *benchmarks* devem possuir as seguintes características para serem considerados válidos.

- *Representatividade*: os *benchmarks* devem representar sistemas reais, ou seja, não devem ser aplicados em sistemas simulados.
- *Portabilidade*: os *benchmarks* devem ser portáveis para diferentes plataformas, proporcionando assim, comparativos de desempenhos entre diversos distribuidores de sistemas.
- *Repetibilidade*: quando um *benchmark* é aplicado no mesmo ambiente, mais de uma vez, ele deve produzir resultados semelhantes.
- *Escalabilidade*: essa característica prevê que os *benchmarks* realizem avaliações em ambientes com diferentes capacidades.
- *Não Intrusividade*: na necessidade de avaliar outro ambiente deve-se realizar o mínimo ou nenhuma alteração nesse novo ambiente.
- *Simplicidade*: os *benchmarks* devem ser de fácil implementação e utilização pelos usuários que farão as avaliações.

Diversos *benchmarks* para realizar medidas de desempenho, nos mais variados sistemas, como servidores *WEB*, SGBDs, aplicações gráficas, entre outros foram idealizados e implementados.

O trabalho realizado por VIEIRA, DURÃES e MADEIRA (2005) especificam e validam *benchmarks* de confiabilidade em sistemas transacionais. Entendem-se sistemas transacionais como sistemas que possuem “motor transacional”, sendo seu maior representante os sistemas gerenciadores de banco de dados.

Na especificação de um *benchmark* de confiabilidade são apresentados os principais componentes que o constituem, os quais são expostos na seqüência:

- *Medidas*: é o componente responsável por quantificar os resultados obtidos durante as avaliações. Elas podem ser subdivididas em duas classes: condicionais ou incondicionais. As medidas condicionais demonstram o funcionamento do sistema e são utilizadas para realizar comparações entre os ambientes. São obtidas através dos resultados das experiências. As medidas incondicionais vão representar a confiabilidade global do sistema, como por exemplo, disponibilidade, integridade, entre outros.
- *Workload*: é o responsável por medir o trabalho do sistema transacional, ou seja, é um sistema que interage com o SGBD para analisar seu desempenho de trabalho. Esse componente é o que mais se assemelha com o propósito da dissertação, uma vez o objetivo é a avaliação de recuperação de objetos multimídia.
- *Faultload*: componente utilizado quando o *benchmark* é voltado para confiabilidade de sistemas. Tem como finalidade identificar e avaliar as causas das falhas apresentadas no sistema, podendo ser considerado como um dos elementos mais complicados de serem identificados, devido à diversidade de elementos causadores e a complexidade das falhas.
- *Procedimentos e regras*: são variáveis dependentes da finalidade do *benchmark*, contanto algumas podem ser utilizadas de forma geral, como as regras para divulgação de resultados finais, construção e aplicação dos *benchmarks* de acordo com as especificações e igualdade de precisão das medidas nos testes realizados em diferentes ambientes.
- *Ambiente experimental*: é a descrição completa do ambiente e plataforma necessária para se executar o *benchmark*.

VIEIRA, DURÃES e MADEIRA (2005) também propõem quatro etapas para especificar um *benchmark* de confiabilidade, são elas:



1. Identificação do domínio da aplicação
2. Caracterização do SUB (*System Under Benchmark*)
3. Definição do componente de medida
4. Definição dos outros componentes

A identificação do domínio é necessária devido à grande quantidade de sistemas. A definição do domínio das aplicações que farão parte do *benchmark* irá facilitar a decisão de quais componentes o categorizam.

Não é possível identificar os componentes que serão utilizados para a avaliação do sistema sem conhecer seu domínio de aplicação, pois muitos componentes de avaliação estão ligados diretamente aos métodos de funcionamento do sistema.

A segunda etapa caracteriza os componentes do SUB, sendo eles as características e funcionalidades esperadas no domínio das aplicações definidas na primeira etapa. Pode-se reunir a terceira e a quarta etapa em uma única, pois essas definem os componentes utilizados no *benchmark*.

A diferença entre as duas etapas é que na terceira, são definidas as medidas que serão obtidas durante a avaliação. Na quarta etapa são definidos os outros componentes que podem depender diretamente das medidas definidas como, por exemplo, as falhas que ocorreram no sistema durante a avaliação, e devem ser expressas nas medidas definidas.

Apesar do trabalho de VIEIRA, DURÃES e MADEIRA (2005), ser voltado para avaliação de confiabilidade, as características descritas mostram de forma genérica, os componentes e etapas a serem seguidos no desenvolvimento de um *benchmark*.

### 3.2. METODOLOGIA

DEMURJIAN et al, (1985) apresentam uma metodologia para avaliar sistemas gerenciadores de banco de dados, na qual é enfatizada uma maneira de obter o tempo gasto entre a requisição de uma consulta e o retorno da sua resposta.

Nessa metodologia, o autor utiliza o conceito de *check-point*, que consiste em criar pontos de coletas de tempo nas requisições de consultas ao

banco de dados através de uma linguagem de programação. Os pontos de coletas de tempo são gravações do relógio do sistema em uma variável. São utilizados dois pontos de coleta de tempo, sendo um no momento em que se faz a requisição e o outro quando se obtém a resposta. Com os dois pontos de tempos coletados, calcula-se a diferença entre eles e como resultado tem-se o tempo gasto na realização da consulta, como apresenta a Figura 7.

```

1  main()
2  {
3      int    starttime,
4          endtime,
5          elapsedtime;
6      int    i;
7
8      ##    char    lstname[20];
9      ##    int     salary;
10
11     ##    ingres empdbase
12     ##    range of e is employee
13
14     starttime = gettime();
15
16     ##    retrieve (lstname=e.lname, salary=e.sal)
17     ##    {
18     ##        /* Disregard results by not doing any */
19     ##        /* processing here. */
20     ##    }
21
22     endtime = gettime();
23
24     ##    exit
25
26     elapsedtime = endtime - starttime;
27
28     printf("s=%d, e=%d, rt=%d0, starttime,
29           endtime, elapsedtime);
30 }

```

Figura 7: Utilização de *check-point*. (DEMURJIAN et. al, 1985)

O trecho de código apresentado na Figura 7 é escrito na linguagem C. Nas Linhas 10 e 12 pode-se observar, respectivamente, o *check-point* que recebe o tempo do relógio do sistema antes da realização da consulta e o *check-point* que recebe o tempo do relógio do sistema depois da realização da consulta.

A Linha 11, que fica entre os *check-point*, é a responsável pela requisição da consulta. Quando nessa linha, o sistema só conseguirá executar a próxima linha ao obter a resposta do sistema de banco de dados.

Com essa característica é possível medir o tempo gasto na execução de uma consulta, pois o tempo inicial é marcado e, em seguida, a linha da requisição da consulta é executada, entretanto, a linha do tempo final não é executada enquanto não se obtém a resposta da consulta.

BORAL e DEWITT (1984) definem que 21 consultas são suficientes para tornar um *benchmark* consistente e representativo. Essas consultas foram separadas em grupos de acordo com o consumo apresentado na utilização do processador e na utilização de entrada e saída dos dados do disco rígido. Os grupos são os que seguem:

	Utilização do Processador	Utilização do Disco
Grupo 1	Baixo	Baixo
Grupo 2	Baixo	Alto
Grupo 3	Alto	Baixo
Grupo 4	Alto	Alto

**Quadro 2: Grupos de consultas definidos por BORAL e DEWITT, (1984).**

Como observado no Quadro 2, o Grupo 1 representa as consultas que apresentam baixo uso nos dois quesitos, processador e disco. O Grupo 2, representa as consultas que apresentam pouco uso do processador e alto uso do disco. O Grupo 3 representa as consultas que demonstram alto uso processador e baixo uso do disco. O Grupo 4, representa as consultas que apresentam alta taxa de utilização nos dois quesitos, processador e disco.

Para definir quais consultas melhor representam cada um dos quatro grupos, foram analisadas as 21 consultas em um *benchmark* que considerou a taxa de utilização do processador e do disco. Assim, as consultas que melhor representam cada grupo são as expostas no Quadro 3.

	Consultas
Grupo 1	<i>selecione 1 linha de 10.000 usando índice clusterizado</i>
Grupo 2	<i>selecione 100 linhas de 10.000 usando índice não clusterizado</i>
Grupo 3	<i>selecione 1.000 linhas de 10.000 que estão ligadas com outras 1.000 linhas usando índice clusterizado</i>
Grupo 4	<i>função agregada em uma tabela com 10.000 linhas</i>

**Quadro 3: Consultas representativas de cada grupo**

No Quadro 3, as consultas estão escritas de forma genérica. De acordo com as conclusões de BORAL e DEWITT (1984) as consultas do Grupo 1 e 2

são representativas em qualquer sistema de banco de dados, entretanto as do Grupo 3 e 4 dependem do algoritmo utilizado pelo SGBD.

A utilização de índices nas consultas representadas no Quadro 3 diz respeito a selecionar as linhas através de uma coluna que seja a chave primária, pois ao se criar uma chave primária em uma coluna, automaticamente é criado um índice na mesma.

A diferença entre se utilizar um índice não *clusterizado* e um *clusterizado*, é que quando são utilizados índices *clusterizados*, os dados permanecem fisicamente ordenados e na mesma página de seus índices. Quando são utilizados índices não *clusterizados* eles apenas identificam o local físico onde a informação está armazenada.

Na atualidade, uma das organizações que promovem *benchmarks* para avaliar o processamento de transações e bancos de dados é a *Transaction Processing Performance Council* (TCP). Ela promove diversas avaliações em ambientes diferentes, dentre as quais são destacados os TPC-C, TPC-H, TPC-W e TPC-App. (TPC, 2005)

O *benchmark* TPC-C é um *workload* para ambientes OLTP (*On Line Transaction Processing*). VIEIRA, DURÃES e MADEIRA (2005) definem ambientes OLTP, como um ambiente em que clientes estão ligados a um servidor através de uma rede, na qual os clientes submetem suas transações ao servidor e aguardam suas respostas. Esse método de avaliação testa tanto desempenho em consultas quanto inserções e atualizações, e a medida utilizada é a quantidade de transações por minuto.

O TPC-H avalia o suporte da base de dados para um sistema de apoio a decisões, *Data Warehouse* (MACHADO, 2004). É avaliado o desempenho do banco de dados no processamento de grandes quantidades de dados convencionais, através de consultas que simulam a atividade real, como perguntas sobre épocas de promoções, lucro possível, entre outros.

O TPC-W avalia todas as atividades de um servidor *WEB*, simulando um comércio eletrônico pela internet. Ele controla, de forma detalhada, todos os componentes associados a essa estrutura. A medida utilizada para avaliar esses componentes é a quantidade de interações na *WEB* por segundo.

Entretanto, esse *benchmark* se tornou obsoleto para a organização TPC e em seu lugar foi inserido o TPC-App que abrange as características e possibilidades de negócio dos servidores *WEB*, como por exemplo, capacidade de conexões, gerenciamento de transações, entre outros.

O ambiente de avaliação do TPC-App inclui os ambientes de avaliação comercial, as múltiplas sessões de negócios on-line, o uso de XML e SOAP (POTTS e KOPACK, 2003) para troca de dados, execução de múltiplas transações, uso de uma base de dados com tamanhos variados de tabelas, com diferentes atributos, relacionamentos, entre outros. (TPC-APP, 2005)

O esquema de dados utilizado no *Benchmark* TPC-App possui um campo, na tabela de itens, que é responsável por referenciar as imagens de cada um dos itens cadastrados. Essa estrutura dará suporte à análise de desempenho de consultas em banco de dados.

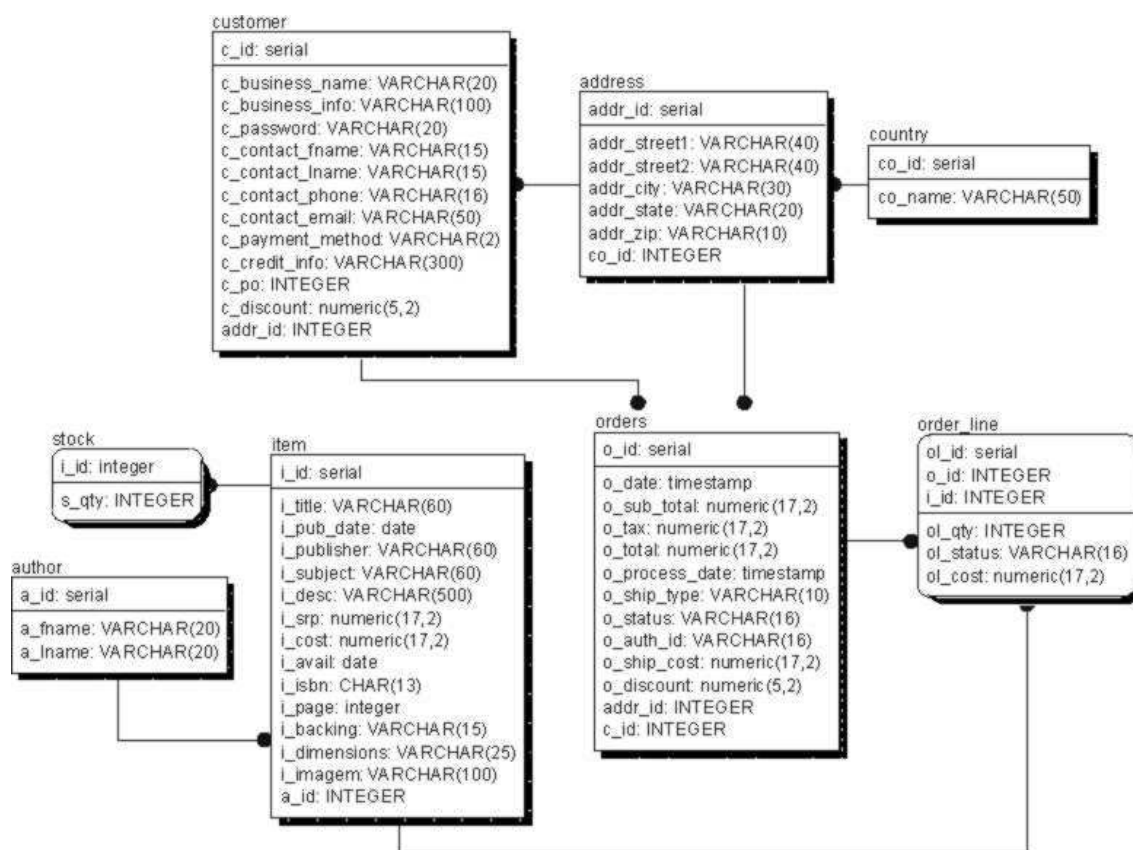


Figura 8: Esquema de dados do TPC-App.

A Figura 8 apresenta o esquema de dados utilizado no *benchmark* TCP-App. A modelagem utilizada na construção desse esquema é a IDEF1X,

---

contanto, foram acrescentados os tipos de dados de cada campo, ao lado de seus respectivos nomes, com a finalidade de proporcionar uma melhor visualização da estrutura física da base de dados como um todo.

A idéia geral das regras de negócios do sistema representado pelo esquema é a de comércio eletrônico, para livraria, onde os clientes podem realizar compras de exemplares de livros pela internet, e entregar sua encomenda no endereço desejado.

O esquema é composto por oito tabelas, a tabela intitulada *customer*, possui as informações necessárias de cadastros dos clientes do sistema. Nessa tabela o campo *addr\_id* é incluído através da ligação com a tabela *address*, o que torna possível a atribuição de um endereço aos usuários. A tabela *address* é responsável por armazenar os campos que representam os endereços utilizados pelos compradores e também na construção do pedido, onde indicará o endereço de entrega. A tabela *orders* possui os campos para armazenar os dados necessários na construção de um pedido de livros, a ligação existente entre essa tabela e a tabela *order\_line* possibilita que um pedido possua exemplares de títulos diferentes. Os exemplares são definidos na tabela *item* onde o atributo *i\_image* guarda a imagem da capa do livro.

Como visto esse esquema possui características suficientes para dar suporte ao um sistema de comércio eletrônico. Devido a essas características o esquema será implementado no próximo capítulo, para prover suporte às análises dos métodos de armazenamento dos objetos multimídia.

## 4. ANÁLISE DE DESEMPENHO EM RECUPERAÇÃO DE OBJETOS MULTIMÍDIA NO POSTGRESQL

A utilização de objetos multimídia nos SGBDs podem interferir em seu desempenho de recuperação. Como visto anteriormente existem quatro técnicas para se armazenar objetos multimídia utilizando um Sistema Gerenciador de Banco de Dados. O armazenamento por referência externa, onde apenas o caminho da localização do objeto no sistema operacional é armazenado. O armazenamento através de funções externas, que são funções externas ao SGBD. O armazenamento por dados não interpretados, onde os objetos são armazenados através de um campo do tipo *BLOB*. E ainda, o armazenamento através de objetos, que permite o relacionamento entre os objetos e a criação de tipos.

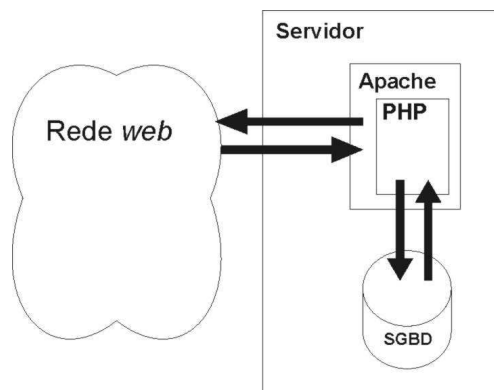
Com o objetivo de avaliar o desempenho da recuperação dos objetos multimídia no SGBD PostgreSQL, esse capítulo está estruturado como segue. Na Seção 4.1 são especificados o ambiente e as técnicas para realizar a análise. Na Seção 4.2 são expostos os tipos de armazenamento dos objetos multimídia permitidos pelo PostgreSQL, bem como o desempenho na recuperação dos objetos em cada um dos tipos. Na Seção 4.3 é realizada uma análise comparativa entre os resultados obtidos na Seção 4.2.

### 4.1. AMBIENTE EXPERIMENTAL

A análise de desempenho de recuperação de objetos multimídia foi baseada em recuperar imagens de uma base de dados mantida no Sistema Gerenciador de Banco de Dados PostgreSQL (POSTGRESQL, 2005).

O ambiente utilizado na análise foi um ambiente de aplicações *WEB*, como mostra a Figura 9, o qual é composto por um servidor de páginas Apache (APACHE, 2005), na Versão 1.3.34, interagindo com a Linguagem de Programação PHP (PHP, 2005), na Versão 5, a qual ira interagir diretamente com o PostgreSQL, na Versão 8.1.3.

A escolha desse ambiente deve-se à crescente utilização de aplicações multimídia na Internet, onde as requisições de imagens e outros objetos multimídia são amplamente utilizados em sistema de lojas virtuais, sistemas educacionais entre outros.



**Figura 9: Ambiente da análise.**

O ambiente apresentado na Figura 9 não recebeu influência temporal significativa por sofrer interações via rede, pois a aplicação de análise consistiu em o PHP receber um pedido de uma consulta e em seguida solicitar ao SGBD a execução da mesma e guardar o tempo decorrido até o retorno da sua resposta. Com isso, não existiu tráfego intenso de informações entre o servidor e a rede WEB. Além disso, a rede WEB foi simulada em uma rede local, devido à análise estar focada no tempo de recuperação dos objetos por parte do SGBD.

O ambiente de análise apresentado na Figura 9 foi implementado em uma máquina com processador *Pentium4 2.4 Gigahertz*, com *512 Megabytes* de memória RAM (*Random Access Memory*) e disco rígido de *60 Gigabytes*. O sistema operacional utilizado foi o *Linux* com a distribuição *Slackware 10.2* com *kernel 2.4.31* e *swap* em disco de *1 Gigabyte*.

O método para medir o tempo de execução das consultas foi o de *checkpoint* utilizado por DEMURJIAN (1985), onde é gravado o tempo em que a consulta é requisitada e o tempo de chegada da resposta. Logo após, a diferença entre os dois tempos resultou no tempo de execução da consulta.



```

1 function microtime_float()
2 {
3     list($usec, $sec) = explode(" ", microtime());
4     return ((float)$usec + (float)$sec);
5 }
6 $inicio = microtime_float;
7 pg_query("select");
8 $fim = microtime_float;
9 $tempo_execucao = $fim - $inicio;

```

**Figura 10: Estrutura *check-point* em php.**

A Figura 10 mostra o trecho do código, escrito em PHP, onde foi calculado o tempo de execução da consulta. A função *microtime\_float()* (PHP MANUAL, 2005) que inicia na Linha 1 e termina na Linha 5, foi construída para retornar o tempo atual em segundos, contados desde as zero hora de primeiro de janeiro de mil novecentos e setenta. Para se calcular o tempo de execução da consulta, que acontece na Linha 7, através da função *pg\_query()*, a variável *\$inicio* na Linha 6 recebeu o valor do tempo atual em segundos. Após isso, a Linha 7 executa a consulta e só será executada a Linha 8, que é a linha em que a variável *\$fim* recebe o tempo atual, após a execução da consulta completa na base de dados. Então, a Linha 9 realiza a diferença entre a variável *\$fim* e *\$inicio*, obtendo assim o tempo de duração da consulta.

Para realizar a conexão entre o PHP e o PostgreSQL foram utilizadas as funções *pg\_connect* e *pg\_dbname*, as quais são disponibilizadas pelo PHP para esse SGBD.

```

1 $localhost="localhost";
2 $dbname="nomeBD";
3 $usuario="nomeUsuario";
4 $password="senha";
5 $porta = "5432";
6
7 if(!($id = pg_connect("dbname=$dbname port=$porta host=$localhost user=$usuario password=$password"))) {
8     echo "Não foi possível estabelecer uma conexão com o gerenciador PostgreSQL.";
9     exit;
10 }
11
12 if(!($con = @pg_dbname($id)) {
13     echo "Não foi possível estabelecer uma conexão com o gerenciador PostgreSQL.";
14     exit;
15 }

```

**Figura 11: Conexão entre PHP e PostgreSQL.**

Na Figura 11 é apresentado o trecho de código, em PHP, para prover a conexão ao SGBD PostgreSQL. Nas cinco primeiras linhas, as variáveis são

atribuídas com as configurações de conexão. A primeira linha atribui à variável o endereço do SGBD, na segunda linha é atribuído o nome do banco de dados, na terceira linha é atribuído o nome do usuário, na quarta linha é atribuído a senha correspondente ao usuário e na quinta linha é atribuído a porta de conexão.

Na sétima linha é criada a conexão com o SGBD através da função *pg\_connect* tendo como parâmetros as variáveis setadas nas cinco primeiras linhas. Na linha doze é criada uma conexão com a base de dados através da função *pg\_dbname*, utilizando a conexão estabelecida com o gerenciador que foi apresentada na sétima linha.

A análise de recuperação dos objetos multimídia foi realizada com o esquema do *benchmark* TCP-App, apresentado na Figura 8. Assim, foram realizadas as análises em um esquema que representa uma base de dados que pode ser empregado em um sistema real.

O grau de povoamento de cada uma das tabelas das bases de dados também seguiu as regras fornecidas pelo *benchmark* TPC-App, as quais são expostas no Quadro 4. (TPC-APP, 2005)

Tabelas	Número de linhas
<i>customer</i>	1.920
<i>country</i>	92
<i>adress</i>	2.688
<i>orders</i>	19.200
<i>order_line</i>	105.600
<i>author</i>	0.25 * número de itens
<i>Item</i>	número de itens
<i>Stock</i>	número de itens

**Quadro 4: Grau de povoamento das tabelas**

O Quadro 4 mostra a quantidade de linhas para cada uma das tabelas da base de dados utilizada na análise. A quantidade de linhas nas tabelas *author*, *item* e *stock* dependem diretamente da quantidade de itens que será utilizada para povoar a tabela *item*.

A tabela *item* é a tabela central da análise, por possuir o objeto multimídia, como mostra o esquema apresentado na Figura 8. Sendo assim, essa tabela possuirá dez mil linhas, pois o princípio de BORAL e DEWITT (1984) considera que para possuir consultas expressivas para um *benchmark*, elas devem ser realizadas em tabelas com dez mil linhas. Além disso, este valor expressa um número razoável de itens quando comparados com sistemas reais de comércio eletrônico.

A partir desse princípio e com base no grau de povoamento apresentado no Quadro 4, consegue-se estabelecer o número de linhas da tabela *author* com 2.500 linhas, da tabela *item* com 10.000 linhas e da tabela *stock* com 10.000 linhas, com base no grau de povoamento abordado no Quadro 4.

Com a finalidade de homogeneizar a análise, as imagens terão o mesmo tamanho em *bytes*, seiscentos *kilobytes* para isso será utilizada a ferramenta de geração de imagens, *applmage*, a qual é disponibilizada pela TPC.

As consultas utilizadas para realizar a análise foram baseadas nas consultas propostas por BORAL e DEWITT (1984), apresentadas no Quadro 3. Nesse trabalho as consultas serão divididas em dois grupos, onde no primeiro grupo, as consultas serão realizadas em um ambiente mono-usuário, ou seja, apenas um usuário consultando o SGBD. No segundo grupo as consultas serão realizadas em um ambiente multi-usuário, vários usuários consultando o SGBD. Nesse trabalho foram utilizados trinta usuários.

As consultas utilizadas na análise multi-usuário e mono-usuário são as mesmas, o que difere entre elas é a quantidade de usuários que estavam executando.

CONJUNTO A	
<i>Consulta 1A</i>	select * from item where i_id = [valor_randomico]
<i>Consulta 2A</i>	select * from item limit 100 offset [valor_randomico <= 9000]
<i>Consulta 3A</i>	select i.* from item i inner join order_line ol on ol.i_id = i.i_id inner join orders o on o.o_id = ol.o_id inner join customer c on c.c_id = o.c_id where c.c_id = [valor_randomico]
<i>Consulta 4A</i>	select * from item limit 20 offset [valor_randomico <= 9980]

CONJUNTO B	
<i>Consulta 1B</i>	select i_title, i_pub_date, i_publisher, i_subject, i_desc, i_srp, i_cost, i_avail, i_isbn, i_page, i_backing, i_dimensions, a_id from item where i_id = [valor_randomico]
<i>Consulta 2B</i>	select i_title, i_pub_date, i_publisher, i_subject, i_desc, i_srp, i_cost, i_avail, i_isbn, i_page, i_backing, i_dimensions, a_id from item limit 100 offset [valor_randomico <= 9900]
<i>Consulta 3B</i>	select i_title, i_pub_date, i_publisher, i_subject, i_desc, i_srp, i_cost, i_avail, i_isbn, i_page, i_backing, i_dimensions, a_id from item i inner join order_line ol on ol.i_id = i.i_id inner join orders o on o.o_id = ol.o_id inner join customer c on c.c_id = o.c_id where c.c_id = [valor_randomico]
<i>Consulta 4B</i>	select i_title, i_pub_date, i_publisher, i_subject, i_desc, i_srp, i_cost, i_avail, i_isbn, i_page, i_backing, i_dimensions, a_id from item limit 20 offset [valor_randomico <= 9980]

**Quadro 5: Consultas para análise**

O Quadro 5 apresenta as consultas que foram utilizadas para avaliar o desempenho na recuperação dos objetos multimídia. O conjunto de Consultas A é responsável por trazer todos os dados da tabela *imagem*. Já o conjunto de Consultas B, é responsável por resultar os mesmos dados de A, mas sem a presença do campo que contém os objetos multimídia.

As Consultas de 1 a 3 são aquelas baseadas nas definições de BORAL e DEWITT (1984), onde a primeira consulta resulta uma linha dentre as dez mil linhas da tabela *item*, a segunda resulta cem linhas dentre as dez mil linhas a partir do valor randômico e a terceira consulta resulta todos os itens comprados em todos os pedidos por um determinado cliente.

A Consulta 4, tanto no Conjunto A quanto no Conjunto B, é a consulta que representa a exigência da maioria dos sistemas de comércio eletrônico, onde o usuário pesquisa os itens oferecidos, e eles são expostos com uma média de vinte produtos por página. Assim, pretendeu-se simular o impacto de cada tipo de armazenamento em uma aplicação amplamente utilizada.

O Valor randômico utilizado como restrição das consultas apresentadas no Quadro 5 foi obtido uma única vez, antes do início da análise e utilizado durante todo o experimento. Assim, todas as consultas recuperaram o mesmo grupo de dados.

## 4.2. ANÁLISE DE DESEMPENHO EM RECUPERAÇÃO DE IMAGENS

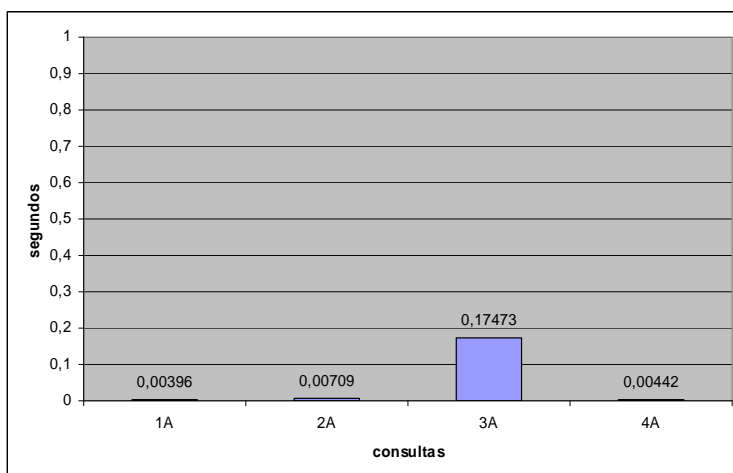
O SGBD PostgreSQL dá suporte a três tipos de armazenamento de objetos multimídia, sendo eles armazenamento por referência, armazenamento de campos não interpretados e armazenamento através de objetos, onde esse último não possui características de orientação a objetos, mas para cada objeto armazenado é criado um objeto interno com um identificador único.

Assim, a análise ocorreu com base no ambiente experimental exposto na Seção 4.1 e com os três tipos possíveis. Sendo assim, esta seção subdivide-se em três subseções, a Seção 4.2.1 que apresenta a análise do desempenho com o armazenamento por referência externa, a Seção 4.2.2 apresenta a análise do desempenho com dados não interpretados e a Seção 4.2.3 apresenta a análise do desempenho com o armazenamento através de objetos.

### 4.2.1. ARMAZENAMENTO POR REFERÊNCIA EXTERNA

No PostgreSQL, como em qualquer outro SGBD, a referência externa é realizada através de um campo texto, o qual contém o caminho de onde se encontra o objeto. Para realizar a análise nesse tipo de armazenamento, o campo *i\_image* da tabela *item*, foi definido com o tipo de dado *varchar(100)*, o qual representa uma *string* de tamanho variável de no máximo 100 caracteres.

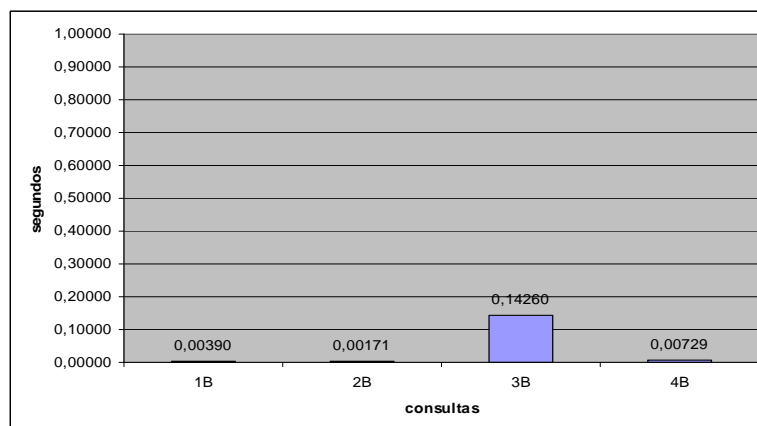
Devido ao fato desse tipo de armazenamento apenas armazenar o caminho do objeto multimídia, o desempenho das consultas foi abaixo de um segundo nos dois esquemas. A Figura 12 apresenta o gráfico demonstrativo do tempo ao realizar as consultas do Conjunto A, expostas no Quadro 5, em ambiente mono-usuário.



**Figura 12: Resultados das consultas do Conjunto A com armazenamento de referência externa em ambiente mono-usuário**

O gráfico apresentado na Figura 12 mostra o desempenho das consultas do Conjunto A em segundos. Apesar da diferença apresentada no gráfico pela consulta 3A, a qual é a consulta que despreendeu mais tempo, pois utiliza junções para encontrar a informação desejada, todas as consultas ficaram abaixo de um segundo. Os tempos apresentados são valores aceitáveis de respostas de um banco de dados, pois para os usuários, esse tempo é imperceptível.

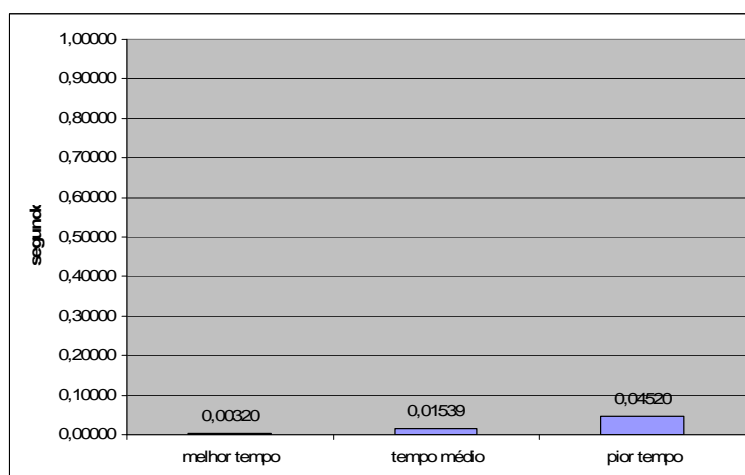
A execução das consultas do Conjunto B, que busca os campos da tabela *item* sem buscar o campo *i\_imagem*, em modo mono-usuário não apresentou grande diferença entre as consultas do Conjunto A, perceptível para o usuário final. Entretanto a diferença apresentada pelo Conjunto de consultas A foi praticamente de 4 vezes mais lenta que o Conjunto de consultas C. Esse resultado é mostrado na Figura 13.



**Figura 13: Resultados das consultas do Conjunto B com armazenamento de referência externa em ambiente mono-usuário**

No gráfico apresentado na Figura 13 é mostrado o desempenho em segundos das consultas, sem a recuperação do campo *i\_image*. Nesse caso, as consultas também ficaram abaixo de um segundo, e não apresentaram diferenças significativas com as consultas apresentadas na Figura 12. Isso deve-se ao campo apenas armazenar o caminho da imagem e não a imagem por completo.

As Consultas 4A e 4B, realizadas em ambiente multi-usuário, obtiveram resultados sem diferença expressiva entre elas. Assim, os resultados das duas consultas foram tabulados em um único gráfico.



**Figura 14: Resultados da consulta 4A e 4B com armazenamento através de referência externa em ambiente multi-usuário.**

Na Figura 14, onde o gráfico expressa melhor tempo, o tempo médio e o pior tempo, que representam tanto os resultados obtidos na Consulta 4A, quanto os resultados obtidos na Consulta 4B. Com esse gráfico observa-se que

o desempenho das consultas de Número 4, com armazenamento por referência externa, em ambiente multi-usuário, ficou abaixo de um décimo de segundo, variando em centésimos de segundo do melhor desempenho em uma das consultas obtida para o pior desempenho em uma das consultas.

Com base nas análises realizadas, observa-se que a execução das Consultas 4A e 4B, em ambiente multi-usuário, não apresentaram diferença expressiva quando comparadas com as mesmas consultas em ambiente monousuário, com a diferença ficando na casa dos centésimos de segundo. Essa aparente igualdade prova que o desempenho do armazenamento através de referência é satisfatório tanto no ambiente mono-usuário quanto no multi-usuário.

#### 4.2.2. ARMAZENAMENTO DE DADOS MULTIMÍDIA NÃO INTERPRETADOS - CAMPO *BLOB*

Os campos *BLOBs* são campos responsáveis por armazenar os dados não convencionais como seqüências binárias. No PostgreSQL o tipo do campo responsável por esse armazenamento é o campo *bytea*. Assim, o campo *i\_image* da tabela *item*, assumirá esse tipo de dado.

Quando esse campo é utilizado com uma interface *WEB* escrita em PHP, é necessária a colocação de *escapes*, que são contra barras seguidos de um conjunto de letras, nas cadeias de caracteres, para indicar, por exemplo, uma nova linha, além da necessidade de duplicar as contra barras, pois o PHP as trata como *escapes* internos.

Para tratar a inserção de *escapes* na cadeia de caracteres, a linguagem PHP disponibiliza a Função *pg\_escape\_bytea()*, que recebe como parâmetro o arquivo. Essa função deve ser aplicada antes da inserção do arquivo no campo, como mostra a Figura 15.



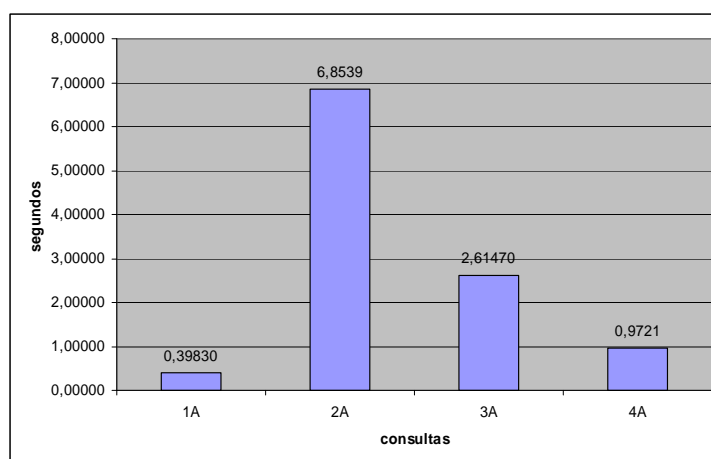
```
1 $fp = fopen("tmp_imagens/$foto_name", "r");
2 $imagem_temp = fread($fp, filesize("tmp_imagens/$foto_name"));
3 fclose($fp);
4 $imagem = pg_escape_bytea($imagem_temp);
5 $select = "insert into imagem (i_id,imagem) values ($codigo, '$imagem'::bytea); ";
```

**Figura 15: Inserção de imagem no campo *bytea* escrito em PHP.**

A Figura 15 mostra um trecho do código utilizado para povoar o banco de dados, onde na Linha 4 é utilizada a Função *pg\_escape\_bytea()* para a colocação de *escapes* no arquivo da imagem. Em seguida na Linha 5 é construída a consulta de inserção da imagem já contendo o padrão de escapes necessário.

A análise de recuperação, nesse tipo de armazenamento, também foi calculada apenas em relação ao tempo de execução da consulta, sem calcular o tempo para a retirada dos *escapes*.

A Figura 16 mostra o gráfico de desempenho em segundos de cada uma das consultas do Conjunto A.



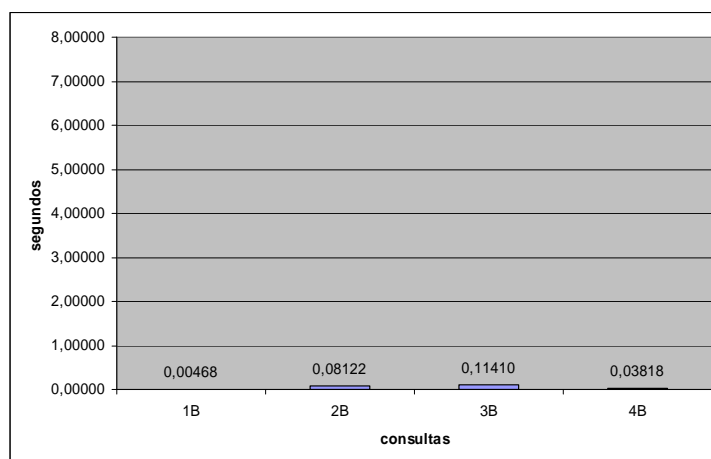
**Figura 16: Resultados das consultas do Conjunto A com armazenamento através do campo *bytea* em ambiente mono-usuário**

Como observado no gráfico, da Figura 16, a consulta 1A foi a que apresentou o menor tempo de resposta. Isso se deve ao fato dessa consulta buscar apenas uma imagem dentre dez mil cadastradas. A consulta 2A foi a que apresentou o maior tempo de resposta e era a que buscava o maior número de imagens dentre as cadastradas.

Com esses dois resultados é possível observar a influência da quantidade de imagens que se deseja buscar no desempenho da pesquisa. Além disso, a Consulta 3A que busca uma imagem através de junções em sua

consulta, quando comparada com a Consulta 1A que não utiliza junções, mostra que as que utilizam junções para buscar a imagem desejada sofrem influência no desempenho.

A Consulta 4A afirma a influência das imagens armazenadas em campos do tipo *bytea* no desempenho das pesquisas, pois o tempo de resposta fica praticamente proporcional com o da Consulta 2A. O desempenho das consultas do Conjunto B, que busca os campos da tabela *item* sem o campo que armazena a imagem, apresentou um excelente desempenho quando comparado com as consultas do Conjunto A, como mostra o gráfico da Figura 17.

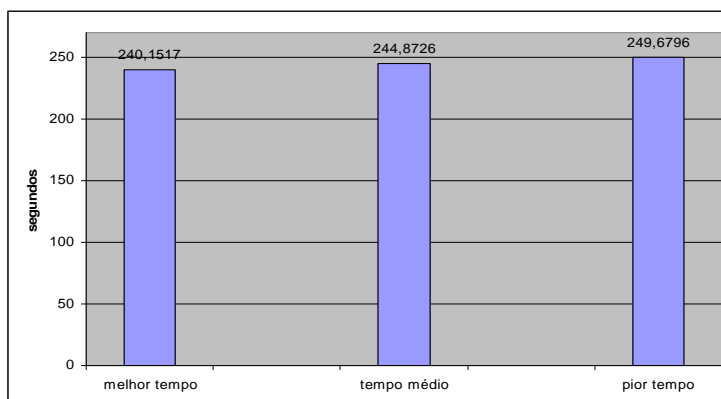


**Figura 17: Resultados das consultas do Conjunto B com armazenamento através do campo *bytea* em ambiente mono-usuário**

No gráfico apresentado na Figura 17, as consultas do Conjunto B demonstraram um excelente desempenho quando comparado com as consultas do Conjunto A, abordadas na Figura 16.

Assim, foi observado que a não requisição de objetos multimídia, nesse tipo de armazenamento, faz com que o tempo de recuperação dos dados fique abaixo de um segundo, tornando a consulta praticamente instantânea.

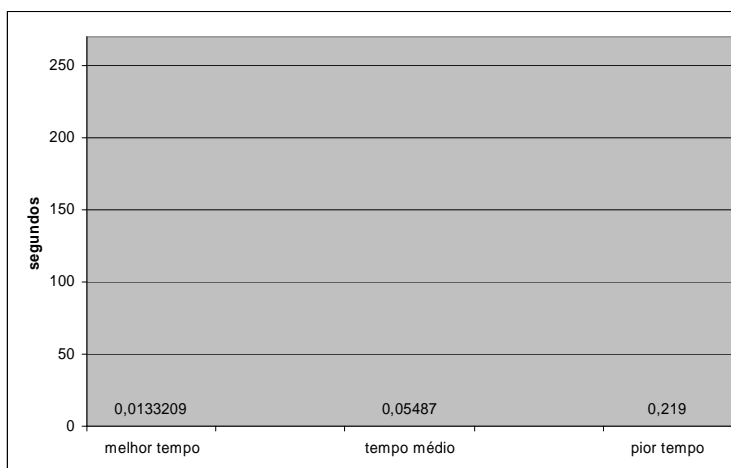
A aplicação da Consulta 4A, a qual retorna os dados convencionais e não convencionais da tabela *item*, em ambiente multi-usuário, demonstrou ineficiência na recuperação de objetos multimídia com armazenamento no campo *bytea*, como mostra a Figura 18.



**Figura 18: Resultados da consulta 4A com armazenamento através do campo *bytea* em ambiente multi-usuário**

O gráfico apresentado na Figura 18 mostra que o melhor tempo da Consulta 4A, que foi obtido por um dos usuários durante a análise, foi de 240,1517 segundos. O pior tempo ficou em 249,6796 segundos e o tempo médio de resposta entre os trinta usuários foi de 244,8726.

No ambiente multi-usuário também foi aplicada a Consulta 4B, a qual retorna apenas os dados não convencionais. Essa consulta se mostrou eficiente quando comparada com a Consulta 4A, pois o tempo médio de execução das Consultas 4B ficou na casa dos centésimos de segundo, se aproximando do tempo obtido em ambiente mono-usuário.



**Figura 19: Resultados da consulta 4B com armazenamento através do campo *bytea* em ambiente multi-usuário**

No gráfico apresentado na Figura 19, são mostrados os tempos da Consulta 4B, a qual retorna apenas os dados não convencionais da tabela *item*.

O melhor tempo de execução da consulta foi de 0,01332 segundos, o pior tempo foi de 0,2190 segundos e o tempo médio foi de 0,05487 segundos.

Com os resultados obtidos nas análises através do armazenamento por dados não interpretados, *bytea*, observa-se que a realização de consultas de objetos multimídia, em ambiente multi-usuário, se torna inviável, devido ao elevado tempo de resposta das mesmas.

Além desse fator, observa-se também que as consultas que não envolveram recuperação de objetos multimídia apresentaram praticamente o mesmo tempo de recuperação nos ambientes mono-usuário e multi-usuário, com a diferença na casa dos centésimos de segundos. Provando assim, que o objeto multimídia armazenado não influi em consultas que não o recuperam, mesmo que essas consultas sejam feitas na sua própria tabela.

#### 4.2.3. ARMAZENAMENTO ATRAVÉS DE OBJETOS

Uma outra forma de armazenar objetos multimídia no PostgreSQL é o armazenamento através de objetos internos, campo do tipo *oid*. Nesse tipo de armazenamento, o SGBD em questão atribui um identificador único ao campo que é destinado ao objeto multimídia. O objeto propriamente dito é armazenado em uma tabela interna chamada *pg\_largeobjects*, com o mesmo identificador atribuído ao campo destinado ao objeto.

Entretanto, para armazenar e recuperar o objeto com essa forma de armazenamento é necessário utilizar duas funções internas do PostgreSQL, *lo\_import* para o armazenamento do objeto e *lo\_export* para a recuperação do objeto.

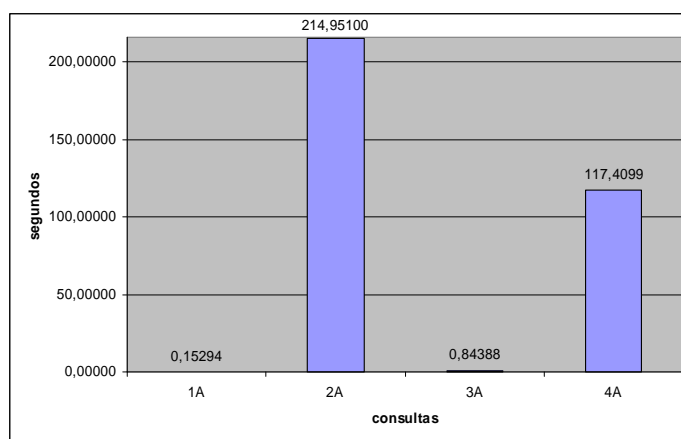
```
1 insert into imagem (i_id,imagem) values (1, lo_import('/tmp/img_1.jpg'));
2
3 select lo_export(imagem, '/tmp/img_1_saida.jpg') from imagem where i_id=1;
```

**Figura 20: Armazenamento e recuperação de objeto utilizando o campo *oid*.**

A Figura 20 apresenta duas linhas que representam a inserção e a consulta de uma imagem com o armazenamento através do campo *oid*. Na Linha 1, é inserida a imagem *img\_1.jpg* com código do item igual a 1. Pode ser notado que para a inserção ser possível a consulta utiliza a Função *lo\_import*, a

qual tem como parâmetro o caminho completo de onde se encontra o arquivo, no lugar do conteúdo da imagem. Na Linha 2, é recuperado o objeto multimídia da tabela *imagem*, cujo o *i\_id* é igual a 1. Para isso, a consulta utiliza a função *lo\_export* como um campo a ser recuperado, essa função possui como parâmetro o campo do tipo *oid* e o caminho de onde o objeto deve ser extraído.

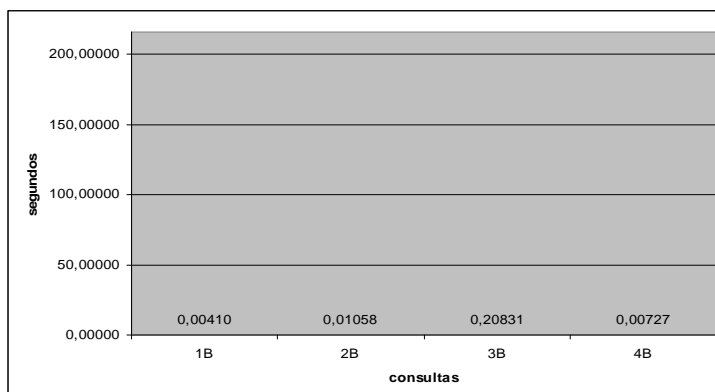
A recuperação das imagens, com esse tipo de armazenamento, apresentou tempo elevado em suas consultas como mostra a Figura 21.



**Figura 21: Resultados das consultas do Conjunto A com armazenamento através do campo *oid* em ambiente mono-usuário**

O gráfico apresentado na Figura 21 demonstra o desempenho de consultas na recuperação de imagens utilizando o campo *oid*. Nessa análise a Consulta 2A, que recupera cem imagens, demorou aproximadamente três minutos para ser realizada e a Consulta 4A, que recupera vinte imagens, demorou aproximadamente dois minutos para ser realizada. Entretanto, as Consultas 1A e 3A, que recuperam uma imagem, foram realizadas em menos de um segundo. Isso demonstra que a quantidade de imagens requisitadas nas consultas possui grande influência no desempenho desse tipo de armazenamento.

Na Figura 22 é apresentado o gráfico de desempenho das consultas do Conjunto B.



**Figura 22: Resultados das consultas do Conjunto B com armazenamento através do campo *oid* em ambiente mono-usuário**

O gráfico apresentado na Figura 22 demonstra o desempenho das consultas do Conjunto B. Com esse conjunto de consultas o desempenho se apresentou dentro da normalidade quando comparado com as outras análises.

Com a análise realizada na forma de armazenamento de objeto, com o campo *oid*, conclui-se que é uma forma inadequada para qualquer aplicação que envolva recuperação de diversos objetos multimídia, pois o tempo de resposta desse tipo de recuperação é inaceitável, passando de três minutos.

Devido ao grande tempo gasto nessas consultas, em ambiente mono-usuário, esse método de armazenamento não será testado em ambiente multi-usuário, onde a quantidade de objetos a serem recuperados seria trinta vezes maior. Isso comparado com a consulta realizada em ambiente mono-usuário e o tempo de resposta seria muito além do aceitável em uma consulta.

#### 4.3. ANÁLISE COMPARATIVA

Quando se deseja construir um banco de dados multimídia, um dos fatores importante é o método de armazenamento que se vai utilizar, pois dependendo do método adotado e a forma de acesso que a base de dados irá sofrer, o desempenho do SGBD pode ser comprometido.

Sendo assim, essa seção discute as diferenças, os pontos positivos e os negativos encontrados nos métodos de armazenamento expostos anteriormente.

No ambiente mono-usuário o comprometimento do desempenho do SGBD entre os métodos de armazenamento ocorreu como mostra o Quadro 6.

	Consulta 1A	Consulta 2A	Consulta 3A	Consulta 4A
Referência externa	0,00396	0,00709	0,17473	0,00442
Dados não interpretados	0,39830	6,85390	2,61470	0,97210
Objetos	0,15294	214,95100	0,84388	117,4099

**Quadro 6: Comparação entre os métodos de armazenamento no Conjunto A de consultas em ambiente mono-usuário**

Os valores apresentados no Quadro 6, como todo o padrão do trabalho, estão em segundos, a primeira linha mostra os resultados do armazenamento por referência externa, a segunda linha mostra os resultados por armazenamento de dados não interpretados, campo *bytea*, e a terceira linha mostra o armazenamento através de objeto, campo *oid*.

Devido à grande diferença de tempo entre a Consulta 2A no método de referência e a Consulta 2A no método de objetos, optou-se por utilizar uma tabela ao invés do gráfico para representar os resultados, pois as comparações ficariam pouco visíveis no gráfico.

Com a comparação apresentada no Quadro 6, observa-se que a referência externa obteve desempenho superior a todos os métodos e consultas analisadas. Entretanto, os métodos por dados não interpretados e por objetos apresentaram variações de acordo com as consultas que foram aplicadas.

Na Consulta 1A, que recupera um objeto dentre dez mil cadastrados, o armazenamento através de objetos foi aproximadamente dois décimos e meio mais rápido que o armazenamento por dados não interpretados. Já na Consulta 2A, a qual recupera cem objetos dentre dez mil cadastrados, o desempenho do armazenamento por dados não interpretados foi muito superior ao desempenho do armazenamento por objetos, sendo esse praticamente 208 segundos mais lento.

Na Consulta 3A, que recupera um objeto através de junções, o desempenho do armazenamento por dados não interpretados foi 2 segundos mais lento que o armazenamento por objetos. E na Consulta 4A, a qual recupera vinte objetos dentre dez mil cadastrados, o armazenamento por

dados não interpretados obteve novamente um melhor desempenho do que o armazenamento por objetos. Foi constatado nessa consulta, que o armazenamento por objetos foi, aproximadamente, 117 segundos mais lenta do que o armazenamento por dados interpretados.

Com essas comparações pode-se expor em que casos devem ser utilizados em um dos dois métodos de armazenamento que mantém o objeto multimídia na base de dados, em ambiente mono-usuário. O método de armazenamento por objetos, campo *oid*, é o indicado para ser aplicado quando é desejado recuperar apenas um objeto multimídia dentre os cadastrados, chegando a ser 2 segundos mais rápido que o armazenamento por dados não interpretados na consulta que utiliza junções. Entretanto esse método é inadequado para consultas que buscam grandes quantidades de objeto, pois com a comparação realizada nota-se o fraco desempenho desse método com o aumento de objetos multimídia a ser recuperado.

Para aplicações em ambiente mono-usuário que necessitam recuperar grandes quantidades de objetos multimídia, o método de armazenamento por dados não interpretado, campo *bytea*, é o indicado. Essa característica fica visível na Consulta 2A, que recupera cem objetos, e na Consulta 4A, que recupera vinte objetos, pois elas foram 202 e 117 segundos mais rápidas do que as mesmas aplicadas com armazenamento através de *oid*.

No ambiente mono-usuário, ainda foi realizada a análise de recuperação dos dados da tabela *item* sem recuperar os objetos multimídia, representada pelas consulta do Conjunto B. No Quadro 7, é apresentado o comparativo dessas consultas.

	Consulta 1B	Consulta 2B	Consulta 3B	Consulta 4B
Referência externa	0,00390	0,00171	0,14260	0,00729
Dados não interpretados	0,00468	0,08122	0,11410	0,03818
Objetos	0,00410	0,01058	0,20831	0,00727

**Quadro 7: Comparação entre os métodos de armazenamento no Conjunto B de consultas em ambiente mono-usuário**

Com os valores expressos no Quadro 7, observa-se que a diferença de tempo nas quatro consultas entre os três métodos utilizados ficou na casa dos centésimos de segundo. Isso prova que o armazenamento de objetos



multimídia por dados não interpretados, o qual armazena os objetos na própria tabela, não afeta o desempenho de consultas que buscam apenas os dados convencionais.

Como abordado na seção anterior, a análise no ambiente multi-usuário foi realizada apenas com armazenamentos por dados não interpretados e por referência externa, devido ao elevado tempo para realizar a Consulta 4A no ambiente mono-usuário com o método de armazenamento por objeto.

A avaliação no ambiente multi-usuário foi realizada de acordo com as especificações apresentadas na Seção 4.1, onde trinta usuários realizaram a mesma consulta.

	Consulta 4A
Referência externa	0,01539
Dados não interpretados	244,87260

**Quadro 8: Comparação entre os métodos de armazenamento com a Consulta 4A em ambiente multi-usuário**

O Quadro 8 apresenta o comparativo da análise realizada em ambiente multi-usuário, onde a consulta realizada foi a 4A e o tempo expresso na tabela é a média dentre as trinta consultas realizadas. Com essa comparação nota-se que o armazenamento por dados não interpretados é mais lenta quando comparada com a referência externa, posto que a diferença entre as duas consultas se aproximou a 245 segundos.

Assim, pode-se afirmar que quando uma base de dados for sofrer acessos em ambiente multi-usuário deve-se optar pelo armazenamento externo, com isso garante-se um bom desempenho do SGBD.

Outra análise realizada foi com base na Consulta 4B, e o comparativo e demonstrado no Quadro 9.

	Consulta 4B
Referência externa	0,01539
Dados não interpretados	0,05487

**Quadro 9: Comparação entre os métodos de armazenamento com a Consulta 4B em ambiente multi-usuário**

A comparação dos métodos com a Consulta 4B no ambiente multi-usuário, mostrada no Quadro 9, confirmou mais uma vez que a consulta de

---

dados convencionais, em tabelas que possuem objetos multimídia armazenados, não sofre influência significativa no desempenho. Pois a diferença de velocidade para realizar a Consulta 4B nos dois tipos de armazenamento ficou na casa dos centésimos de segundo.

Com as comparações realizadas nessa seção conclui-se que o método que apresenta o melhor desempenho de recuperação, no PostgreSQL, é o de referência externa, onde os objetos multimídia são armazenados fora do SGBD. Apesar desse método não garantir a consistência e integridade dos dados, pode-se justificar o seu uso pelo alto desempenho quando comparado ao desempenho do armazenamento por dados não interpretados e por objetos.

Também ficou exposto que os objetos multimídia não afetam o desempenho da base de dados quando não requisitados nas consultas, como foi observado em vários momentos nessa seção ao analisar as consultas do Conjunto B.

Por outro lado, a requisição dos objetos multimídia com armazenamento através de dados não interpretados, principalmente no ambiente multi-usuário, se mostrou ineficiente do ponto de vista de desempenho, levando aproximadamente na média, quatro minutos para realizar a consulta, contra 0,01539 segundos do armazenamento por referência.

Além disso, o armazenamento através de objetos apresentou o desempenho para recuperar vários objetos multimídia no ambiente mono-usuário muito próximo ao desempenho do armazenamento por dados não interpretado em ambiente multi-usuário. Assim, pode-se considerar que o armazenamento por objeto é o mais ineficiente entre os analisados para recuperar vários objetos multimídia.

Como conclusões das análises e expostas nessa seção, pode-se afirmar que o armazenamento de objetos multimídia dentro das bases de dados do SGBD PostgreSQL, ainda é um recurso que deve ser aperfeiçoado em relação ao tempo gasto para realizar as consultas dos objetos.

## 5. CONCLUSÕES E TRABALHOS FUTUROS

Com a evolução da tecnologia e a implantação das redes de alta velocidade, cada vez mais os sistemas multimídias são utilizados, seja com fim educacional, comercial ou profissional. Em torno dessas aplicações, muito se discute sobre a maneira correta de realizar o armazenamento dos objetos multimídia para proporcionar uma busca eficiente e com bom tempo de desempenho.

Nesse trabalho foram apresentadas as principais formas de armazenamento para os objetos multimídia em Sistemas Gerenciadores de Banco de Dados. Com base nessas formas, desenvolveu-se uma análise no PostgreSQL, com a intenção de descobrir o impacto que cada uma proporcionava no desempenho do SGBD.

Através dos testes, foi possível comprovar a grande influência do armazenamento dos objetos multimídia dentro das bases de dados no desempenho das consultas que buscam os mesmos, principalmente em ambiente multi-usuário.

O método de armazenamento por referência externa é o método que realiza as buscas dos objetos com o melhor desempenho, com o tempo de resposta ficando na casa dos décimos de segundo. Entretanto, esse método não traz o objeto propriamente dito, mas sim a sua localização no sistema operacional.

Em relação ao armazenamento através de dados não interpretados, ele foi o método que apresentou o melhor desempenho dentre os dois métodos que armazenam o objeto multimídia dentro da base de dados. Esse método chegou a ser três minutos e meio mais rápido que o método de armazenamento por objeto, tempo esse obtido na consulta que busca cem objetos multimídia em ambiente mono-usuário.

Apesar de ser mais rápido do que o método de armazenamento por objeto, o método por dados não interpretados foi mais lento do que a referência externa, chegando a uma diferença de quatro minutos na consulta que busca vinte objetos multimídia em ambiente multi-usuário.

O método por armazenamento através de objetos foi o método que, dentre os três utilizados na análise, apresentou o pior desempenho no tempo de recuperação de vários objetos multimídia. O tempo de recuperação desse método foi superior aos demais nas consultas que buscavam cem e vinte objetos em ambiente mono-usuário.

Nessas consultas o armazenamento por objeto teve uma diferença de até três minutos e meio, quando comparado com a referência externa na consulta que busca cem objetos em ambiente mono-usuário.

Com as consultas que buscam apenas os dados convencionais da tabela *item*, sem requisitar o objeto multimídia, provou-se que a presença dos objetos multimídia em tabelas não possui grande influência nas consultas que não os requisitam. Posto que a diferença do tempo gasto nessas consultas não passou de alguns centésimos de segundo.

As diferenças encontradas no desempenho de recuperação dos objetos multimídia, mostrou que os métodos de armazenamento que guardam os dados não convencionais dentro das bases de dados no PostgreSQL devem ser usados com cautela.

Para utilizar esses métodos deve-se levar em consideração o ambiente de uso, mono-usuário ou multi-usuário, e a quantidade de objetos que devem ser recuperados, como mostra o Quadro 10.

	<b>Mono-usuário um objeto</b>	<b>Mono-usuário mais de vinte objetos</b>	<b>Multi-usuário mais de vinte objetos</b>
<b>Referência externa</b>	ACONSELHADO	ACONSELHADO	ACONSELHADO
<b>Dados não interpretados</b>	ACONSELHADO	NÃO ACONSELHADO	NÃO ACONSELHADO
<b>Objetos</b>	ACONSELHADO	NÃO ACONSELHADO	NÃO ACONSELHADO

**Quadro 10: Proposta de uso dos métodos de armazenamento em PostgreSQL.**

O Quadro 10 mostra que o método de referência externa é o método que pode ser usado em todos os ambientes analisados e com grandes quantidades de objetos a serem recuperados, devido ao tempo de recuperação ter sido o melhor em todas as análises.

Os métodos por armazenamento de dados não interpretados e por armazenamento através de objeto podem ser usados em ambiente mono-

---

usuário com busca a um objeto. Nos outros ambientes, o tempo de recuperação desses métodos não é satisfatório.

É válido ressaltar que em qualquer ambiente ou quantidade de objetos a serem recuperados, esses métodos possuem um desempenho menor que a referência externa.

Como proposta de trabalhos futuros pretende-se analisar o uso da memória RAM, do disco rígido e do processador, durante a execução das consultas, encontrando assim o *hardware* que possui maior grau de influência em base de dados multimídia. Com essa análise será possível propor uma configuração adequada de *hardware* para a utilização de SGBDM.

Outra proposta seria a realização do *benchmark*, criando a base de dados com a tabela *item* criada em uma *tablespace* localizada em outro disco rígido, ou seja, através do recurso de *tablespace* do PostgreSQL somente a tabela *item* será armazenada em um outro disco. Com esse método pretende-se analisar o desempenho do SGBDM tendo um disco exclusivo para o armazenamento e a pesquisa dos objetos multimídia.

Também, se considera como proposta de trabalho futuro a aplicação da análise em diferentes SGBDs, assim pode-se determinar se a falta de desempenho nos métodos de armazenamento por dados não interpretados e por objeto, é um problema exclusivo do SGBD PostgreSQL ou é pertinente aos demais SGBDs.

Por fim aplicar o *benchmark* em uma estrutura descentralizada onde o SGBD ficaria instalado em uma máquina dedicada a ele, separando a estrutura de servidor *WEB* do Sistema Gerenciador de Banco de Dados. Com essa estrutura pode-se obter o grau de influência da estrutura utilizada nesse trabalho para uma estrutura descentralizada.

## REFERÊNCIAS

APACHE. **The Apache Software Foundation.** Disponível em: <<http://www.apache.org/>> Acesso em: 10 nov. 2005.

BORAL, H.; DEWITT, D. J. **A Methodology for Database System Performance Evaluation.** Proceedings of the 1984 SIGMOD Conference, June, 1984. Disponível em: <<http://www.cs.wisc.edu/~dewitt/benchmarking/sigmod84.pdf>>. Acesso em: 15 nov. 2005.

CERVI, C. R.; SILVA, J. T.; PAVAN, W. **Proposta de uma Biblioteca Digital.** Disponível em: <<http://inf.unisul.br/~ines/workcomp/cd/pdfs/2411.pdf>>. Acesso em: 15 nov. 2005.

CORBIS. **Corbis: stock photography and digital pictures.** Disponível em: <<http://pro.corbis.com/>> Acesso em: 15 nov. 2005.

DEMURJIAN, A. S.; HSIAO, D. K.; KERR, D. S.; et al. **Performance Measurement Methodologies For Database Systems.** ACM, 1985.

DCMI. **The Dublin Core Metadata Initiative.** Disponível em: <<http://dublincore.org/>> Acesso em: 15 nov. 2005.

FGDC. **The Federal Geographic Data Committee.** Disponível em: <<http://www.fgdc.gov/>> Acesso em: 15 nov. 2005.

FRANÇA, B. M. **Banco de Dados Multimídia.** Disponível em: <[http://www.inf.ufrgs.br/~clesio/cmp151/cmp15120021/artigo\\_montgomery.pdf](http://www.inf.ufrgs.br/~clesio/cmp151/cmp15120021/artigo_montgomery.pdf)> Acesso em: 11 set. 2005.

GARCIA-MOLINA, H.; ULLMAN, J. D.; WIDOM, J. **Implementação de Sistemas de Banco de Dados.** Rio de Janeiro: Campus, 2001.

---

GHAFOOR, A. **Multimedia Database Management Systems**. ACM Computing Surveys, Vol. 27, No. 4, Dez. 1995.

JAIMES, A.; CHRISTEL, M.; SÉBASTIEN, G., et al. **Multimedia Information Retrieval: What is it, and why isn't anyone using it?** ACM, Singapura, Nov. 2005.

LOH, S.; WIVES, L. K.; FRAINER, A. S. **Recuperação Semântica de Documentos Textuais na Internet**. In: Conferencia Latioamericana de Informática, XXV, 1999. Assunción, Paraguay: Universidad Autónoma de Asunción, p. 827-836, 1999.

MACHADO, F. N. R.; **Tecnologia e Projeto de Data Warehouse**. São Paulo: Érica, 2004.

MILSTEAD, J.; Feldman, S. **Metadata: Cataloging by Any Other Name**. On Line Magazine, jan. 1999.

NISO, National Information Standards Organization. **Understanding Metadata**. Bethesda: NISO Press, 2004.

ÖZDEN, B.; RASTOGI, R.; SILBERSCHATZ A. **Multimedia Support for Databases**. Arizona: ACM, 1997.

PHP. **PHP: Hipertext Preprocessor**. Disponível em: <<http://www.php.net/>> Acesso em: 10 nov. 2005.

PHP MANUAL. **PHP: microtime – Manual**. Disponível em: <[http://www.php.net/manual/pt\\_BR/function.microtime.php](http://www.php.net/manual/pt_BR/function.microtime.php)> Acesso em: 30 nov. 2005.

---

POTTS, S.; KOPACK, M.; **Aprenda em 24 Horas Web Services**. São Paulo: Campus, 2003.

POSTGRESQL. **PostgreSQL: The world's most advanced open source database**. Disponível em: <<http://www.postgresql.org/>> Acesso em: 10 nov. 2005.

SILVA, R. C. **Avaliação da Plataforma .Net em Ambientes Free para Sistemas E-Learning**. Ponta Grossa, 2004. Monografia (Especialização em Informática) – Universidade Estadual de Ponta Grossa.

TPC. **The Transaction Processing Performance Council**. Disponível em: <<http://www.tpc.org/> > Acesso em: 10 nov. 2005.

TPC-APP. **TPC Benchmark App (Application Server) Especification Version 1.1.1**. Disponível em: < [http://www.tpc.org/tpc\\_app/spec/TPC-App\\_V1.1.1.pdf](http://www.tpc.org/tpc_app/spec/TPC-App_V1.1.1.pdf)> Acesso em: 22 nov. 2005.

VIEIRA, M.; DURÃES, J.; MADEIRA, H. **Especificação e Validação de Benchmarks de Confiabilidade para Sistemas Transacionais**. IEEE Latin America Transactions, Jun. 2005.

VAZ, M. S. M. G. **MetaMídia – Um Modelo de Metadados na Indexação e Recuperação de Objeto Multimídias**. Recife, 2000. Tese (Doutorado em Ciência da Computação) – Universidade Federal de Pernambuco.