

UNIVERSIDADE FEDERAL DE PERNAMBUCO

**GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
CENTRO DE INFORMÁTICA**

2007.1

**Análise de Serviços de Manipulação de Objetos
Multimídia em Sistemas Virtuais de Ensino**

TRABALHO DE GRADUAÇÃO

Aluno – Antonio Josivaldo do Nascimento Filho, ajnf@cin.ufpe.br.

Orientador – Prof. Phd Fernando da Fonseca de Souza, fdfd@cin.ufpe.br.

Co-orientador – Armando Soares Sousa, ass2@cin.ufpe.br

20 de Agosto de 2007

UNIVERSIDADE FEDERAL DE PERNAMBUCO

**GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
CENTRO DE INFORMÁTICA**

2007.1

ANTONIO JOSIVALDO DO NASCIMENTO FILHO

**ANÁLISE DE SERVIÇOS DE MANIPULAÇÃO DE OBJETOS
MULTIMÍDIA EM SISTEMAS VIRTUAIS DE ENSINO**

Este trabalho foi apresentado à graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

**Orientador: Prof. Phd Fernando da Fonseca de Souza
Co-orientador: Armando Soares Sousa**

20 de Agosto de 2007

Dedico,

***a Deus
à minha família
aos meus professores
aos meus amigos***

© Antonio Josivaldo do Nascimento Filho, 2007.

Todos os direitos reservados.

Agradecimentos

Agradeço primeiramente a Deus por ter permitido o ingresso na Universidade e ter me sustentado em todos os momentos. Aos meus pais, Ana e Josivaldo, por sempre investirem na minha educação e sempre me incentivarem em todas as circunstâncias. À minha irmã, Amanda, que sempre me deu palavras animadoras que me impulsionaram a estudar mais. À minha noiva, Bruna, que sempre foi compreensiva nos momentos que tive que me ausentar pela universidade e que sempre me animou em momentos difíceis. A todos os meus familiares que de alguma forma me ajudaram.

Ao meu orientador, professor Phd Fernando Fonseca, pela oportunidade de trabalhar neste importante projeto e pelas importantes ressalvas dadas durante o trabalho de pesquisa.

Ao meu co-orientador, Armando Soares, que com as suas cobranças e observações me ajudou a manter o ritmo de progresso para a conclusão deste trabalho.

Ao professor Doutor Franck Bellemain que sempre me ajudou durante minha graduação e que me deu a oportunidade de trabalhar dois anos com sua pesquisa em ambientes virtuais de ensino.

Aos meus amigos pelos inesquecíveis momentos de companheirismo e descontração durante o curso.

Muito Obrigado!

Sumário

1.	Introdução.....	1
1.1	Objetivos	2
1.2	Estrutura.....	2
2.	Objetos Multimídia	4
2.1	Imagem Digital.....	4
2.1.1	Características Fundamentais de Imagens.....	6
2.1.2	Padrões de Compressão de Imagem Digital.....	8
2.2	Áudio Digital.....	10
2.2.1	Características fundamentais de áudio.....	11
2.2.2	Padrões de Compressão de Áudio Digital.....	13
2.3	Vídeo Digital	16
2.3.1	Características de Vídeo Digital	17
2.3.2	Padrões de Compressão de Vídeo Digital.....	20
2.4	Considerações Finais	28
3.	Pesquisa comparativa de Serviços WEB de manipulação de objetos multimídia.....	29
3.1	Dados comparativos.....	29
3.1.1	Serviços de Imagem Digital.....	29
3.1.2	Serviços de Áudio Digital	31
3.1.3	Serviços de Vídeo Digital.....	32
3.2	Considerações Finais	33
4.	Análise dos Protótipos.....	35
4.1	Ambiente de Desenvolvimento.....	36
4.2	Escopo dos Protótipos.....	36
4.3	Diagrama de Classes	38
4.4	Protótipo 1	40
4.4.1	Estrutura do Banco de dados	42
4.5	Protótipo 2	42
4.5.1	Estrutura do Banco de dados	44
4.6	Protótipo 3.....	44
4.6.1	Estrutura do Banco de dados	46
4.7	Protótipo 4	46
4.7.1	Estrutura do Banco de dados	48
4.8	Considerações Finais	49
5.	Análise dos testes e resultados obtidos	50
5.1	Plano de Testes	50
5.1.1	Descrição do Servidor.....	50
5.1.2	Etapas dos Testes	51
5.2	Resultados.....	52
5.2.1	Testes nos Cenários Utilizando Arquivos de Imagens.....	53
5.2.2	Testes nos Cenários Utilizando Arquivos de Áudios.....	55

5.2.3	Testes nos Cenários Utilizando Arquivos de Vídeos.....	58
5.3	Considerações Finais	60
6.	Conclusões	63
6.1	Contribuições	63
6.2	Dificuldades Encontradas	64
6.3	Trabalhos Futuros	65
	Referências	66
	Assinaturas.....	73

Lista de Figuras

Figura 1: Etapas de Digitalização de uma imagem. Adaptado de [GATTASS, 2005].	5
Figura 2: Imagem colorida, em tons de cinza e bitonal. Adaptado de [WIKIPEDIA, 2007].	7
Figura 3: Exemplo de resolução em pixels. Adaptado de [WIKIPEDIA, 2007].	7
Figura 4: Exemplo de resolução em dpi (respectivamente 600, 300, 150, 60, 30). Adaptado de [BESSER & TRANT, 1995].	7
Figura 5: Áudio Analógico (Vermelho) em Digital (Cinza). Adaptado de [WIKIPEDIA, 2007].	11
Figura 6: Codificador MP3. Adaptado de [CHIARIGLIONE, 1996].	15
Figura 7: Proporção entre diversas resoluções. Adaptado de [WIKIPEDIA, 2007].	18
Figura 8: Proporção entre diversos aspect ratios. Adaptado de [WIKIPEDIA, 2007].	18
Figura 9: Relação Mídia x Espaço Ocupado. Adaptado de [FERNANDES, 2006].	20
Figura 10: Decodificador MPEG-1. Adaptado de [CHIARIGLIONE, 1996].	23
Figura 11: Componentes do MPEG-4. Adaptado de [TODESCO, 2000].	25
Figura 12: Usos do H.264. Adaptado de [APPLE, 2006].	26
Figura 13: Comparação entre H.264 e MPEG4 part 2. Adaptado de [APPLE, 2006].	27
Figura 14: Diagrama de Classes dos Protótipos.	39
Figura 15: Arquitetura do Protótipo 1.	41
Figura 16: Tabelas do Protótipo 1.	42
Figura 17: Arquitetura do Protótipo 2.	43
Figura 18: Tabelas do Protótipo 2.	44
Figura 19: Arquitetura do Protótipo 3.	45
Figura 20: Tabelas do Protótipo 3.	46
Figura 21: Arquitetura do Protótipo 4.	48
Figura 22: Tabelas do Protótipo 4.	48
Figura 23: Inserção de Imagem (10 usuários).	53
Figura 24: Consulta de Imagem (10 usuários).	53
Figura 25: Inserção de Imagem (100 usuários).	54
Figura 26: Consulta de Imagem (100 usuários).	54
Figura 27: Inserção de Áudio (10 usuários).	55
Figura 28: Consulta de Áudio (10 usuários).	56
Figura 29: Inserção de Áudio (100 usuários).	56
Figura 30: Consulta de Áudio (100 usuários).	56
Figura 31: Inserção de Vídeo (10 usuários).	58
Figura 32: Consulta de Vídeo (10 usuários).	58
Figura 33: Inserção de Vídeo (100 usuários).	59

Figura 34: Consulta de Vídeo (100 usuarios). 59

Lista de Quadros

Quadro 1: Exemplos de bit rate em áudio digital. Adaptado de [WIKIPEDIA, 2007].....	11
Quadro 2: Comparação na etapa de inserção de imagens.....	30
Quadro 3: Comparação na etapa de consulta de imagens.....	30
Quadro 4: Comparação na etapa de inserção de áudios.....	31
Quadro 5: Comparação na etapa de consulta de áudios.....	31
Quadro 6: Comparação na etapa de inserção de vídeos.	32
Quadro 7: Comparação na etapa de consulta de vídeos.....	33
Quadro 8: Cenários de Testes.....	51
Quadro 9: Comparação do Desempenho do Protótipos.	54
Quadro 10: Comparação do Desempenho do Protótipos.	57
Quadro 11: Comparação do Desempenho do Protótipos.	60
Quadro 12: Comparação Final dos Protótipos.	61

1. Introdução

Notoriamente, a área de ambientes de ensino a distância tem crescido a cada dia. Diversas instituições, principalmente de ensino, têm dedicado investimentos com o intuito de utilizarem as vantagens obtidas com o uso destes sistemas.

Nestes ambientes, o aprendiz e o tutor podem até estar em países diferentes e isso não afeta o desenrolar do curso. Além de reduzir drasticamente os custos com salas de aulas tradicionais, esta abordagem também possibilita um aumento do público-alvo de diversas instituições.

Os ambientes de ensino a distância utilizam dois pilares fundamentais para possibilitar a disseminação do conhecimento a distância: o primeiro é o gerenciamento e armazenamento de informações e o outro é a transferência de informações via Internet.

A nossa pesquisa visa analisar os serviços de manipulação de objetos multimídia em sistemas virtuais de ensino, não ficando restrita a apenas ao armazenamento e consulta destes objetos, durante este trabalho analisamos a manipulação de dados multimídia em sua totalidade, partindo da base de dados até a visualização do conteúdo do objeto multimídia no navegador do usuário final. Desta forma, a nossa pesquisa é relevante para os dois pilares de ambientes de ensino a distância.

Por fim, de acordo com os resultados obtidos, analisamos as quatro abordagens testadas e indicamos uma abordagem que apresentou os melhores resultados para o contexto do ambiente de ensino a distância AMADEUS_MM, projeto no qual esta pesquisa se realiza. (Projeto apoiado pelo CNPq, Processo Nº 507487/2004-4) [AMADEUS, 2007]. Esta abordagem indicada será uma das abordagens que serão testadas minuciosamente pelo Mestrando Armando Soares Sousa em sua dissertação "Servidor de dados multimídia para gerenciamento de

conteúdo em sistemas virtuais de ensino" que tem como foco propor um sistema de manipulação de objetos multimídia para o AMADEUS_MM.

1.1 Objetivos

Seguindo esse contexto, este trabalho de graduação tem como objetivos analisar, documentar algumas arquiteturas para a manipulação de objetos multimídia e indicar, dentre as estudadas, uma possível arquitetura de manipulação de objetos multimídia em servidores de dados que se enquadre no âmbito de sistemas virtuais de ensino, e como já foi dito esta arquitetura indicada dependendo de outros testes mais apurados poderá ser refinada e integrada ao ambiente de ensino a distância AMADEUS_MM [AMADEUS, 2007].

Basicamente, este trabalho de graduação dará ênfase a quatro aspectos:

1 – Abordar aspectos conceituais sobre os três tipos de objetos multimídia que serão utilizados como base de pesquisa: imagens, áudios e vídeos;

2 – Realizar uma pesquisa comparativa dos principais serviços WEB de manipulação destes objetos;

3 – Definir, especificar e implementar protótipos baseados nas diferentes abordagens de armazenamento e manipulação destes objetos na WEB estudadas;
e

4 – Analisar e realizar testes de modo a indicar as vantagens e desvantagens de cada protótipo tendo em vista indicar um, dentre os protótipos construídos, que melhor se enquadre ao contexto de sistemas virtuais de ensino.

1.2 Estrutura

Além deste capítulo inicial, que apresenta o contexto e os objetivos da pesquisa, este trabalho possui mais cinco capítulos.

O Capítulo 2 aborda os conceitos relativos a objetos multimídia: os diversos padrões de compressão e as características fundamentais de áudios, imagens e vídeos digitais.

O Capítulo 3 apresenta uma pesquisa comparativa sobre os mais usados serviços de dados multimídias presentes na WEB.

O Capítulo 4 aborda a implementação de quatro possíveis opções de armazenamento e manipulação de objetos multimídia em servidores de dados.

O Capítulo 5 mostra uma análise de testes com as quatro abordagens estudadas no capítulo anterior, e que foram desenvolvidas utilizando as seguintes tecnologias: Java Server Pages (JSP) [JSP, 2007], Apache Tomcat [TOMCAT, 2007] e PostgreSQL [POSTGRESQL, 2007]. Também foi utilizado o Framework JMeter [JMETER, 2007] para a realização de testes de carga com possíveis cenários da aplicação.

Por fim, o Capítulo 6, que apresenta as conclusões do trabalho de pesquisa, suas contribuições, as principais dificuldades encontradas e uma proposta de trabalhos futuros.

2. Objetos Multimídia

Neste capítulo abordaremos a base conceitual necessária ao entendimento da importância da manipulação de objetos multimídia em servidores de dados multimídia. Nele iremos estudar as características fundamentais de imagens, áudios e vídeos digitais bem como iremos abordar diversos padrões de compressão existentes, visto que são de extrema importância por implicarem diretamente com o tamanho do objeto e a sua qualidade.

2.1 Imagem Digital

O termo imagem digital se refere aos diversos padrões de armazenamento de imagens em um meio digital. A imagem digital possibilita o compartilhamento de fotos, figuras, desenhos, entre outros, na Internet.

Como acontece com áudio e vídeo digital, que iremos abordar adiante, apenas realizar a transformação da imagem contínua para uma digital não é suficiente. Para permitir o compartilhamento de imagens na WEB, é necessário realizar uma etapa de compressão da imagem, para diminuir o espaço de armazenamento e conseqüentemente a quantidade de informação que é necessária para manipular imagens na Internet.

Conforme GONZALEZ & WOODS (1992): “Uma imagem digital é uma matriz onde os índices das linhas e colunas identificam um ponto na imagem e o valor do elemento da matriz identifica a informação de cor ou brilho naquele ponto”. Os elementos de tal arranjo digital são chamados de elementos da imagem, picture elements, pixels ou pels, com os dois últimos sendo comumente usados como abreviação de “picture elements”.

Segundo GATTASS (2005): “O processo de digitalização de imagens compreende três importantes etapas (Figura 1):

- Amostragem, nesta etapa ocorre à conversão da imagem contínua em um conjunto de coordenadas discretizadas;
- Quantização, nesta etapa tem à codificação de valores contínuos de cores em um intervalo discreto; e
- Codificação, que é a representação, em seqüência de números, da imagem digital.

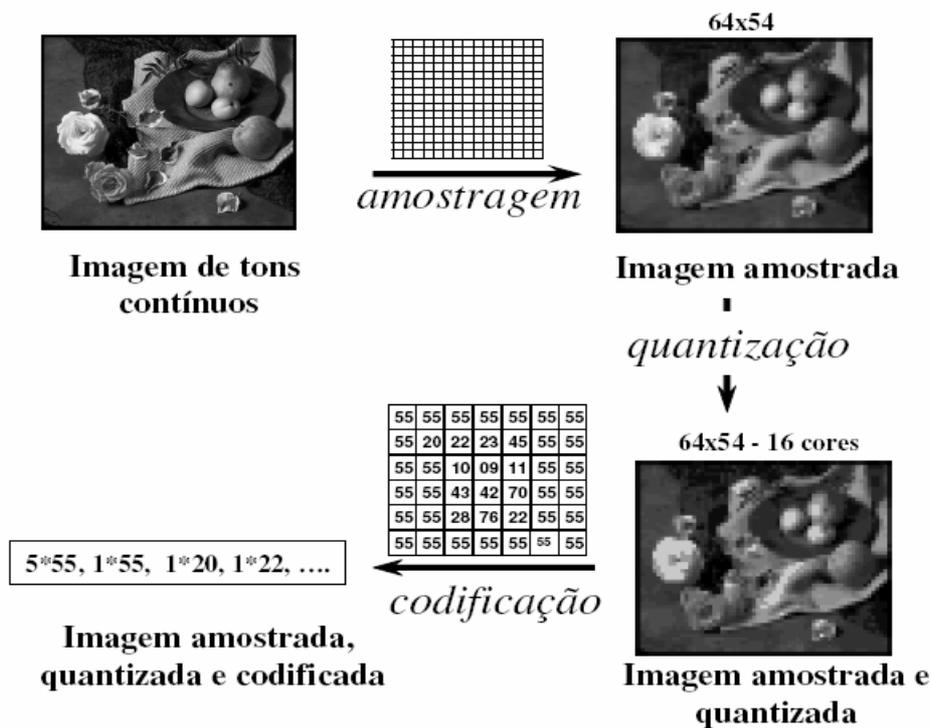


Figura 1: Etapas de Digitalização de uma imagem. Adaptado de [GATTASS, 2005].

A seguir, analisaremos três dos principais padrões de armazenamento e compressão usados em imagens digitais na Internet. Os seguintes padrões foram escolhidos, por serem os mais utilizados:

- JPEG, Padrão criado pelo Joint Photographic Experts Group (JPEG) [JPEG, 20007] que utiliza lossy compression (compressão com perda de informação);
 - GIF, Padrão criado pela Comuserve [COMPUSERVE, 2007] que utiliza lossless compression (compressão sem perda de informação);
- e

- PNG, Padrão livre regulamentado pela W3C [W3C, 2007], utiliza lossless compression e é o principal concorrente do GIF.

2.1.1 Características Fundamentais de Imagens

Nesta seção iremos descrever algumas características importantes de imagens digitais.

2.1.1.1 Color Depth (Profundidade da Cor)

Profundidade da cor, ou *color depth*, é um termo da computação gráfica que descreve a quantidade de bits usados para representar a cor em um único pixel numa imagem bitmap [WIKIPEDIA, 2007].

Conforme CORNELL (2000), existem três tipos de imagens de acordo com a profundidade da cor:

- Imagem bitonal - Representada por pixels consistindo de um bit cada, onde dois tons podem ser representados (tipicamente preto e branco), usando o valor 0 para preto e 1 para branco ou vice-versa;
- Imagem em tons de cinza - Composta de pixels representados por múltiplos bits de informação, tipicamente estendendo de dois a oito bits ou mais; e
- Imagem colorida - Composta de pixels representados por múltiplos bits de informação, tipicamente estendendo de oito bits a vinte e quatro. Numa imagem de 24 bits, estes são divididos em três grupos: oito bits para o vermelho, oito bits para o verde e oito bits para o azul. As combinações destes bits são utilizadas para representar as outras cores. Uma imagem de 24 bits oferece 16,7 milhões de cores.

Com a Figura 2 podemos perceber a diferença de qualidade que a profundidade de cor acarreta para uma imagem.



Figura 2: Imagem colorida, em tons de cinza e bitonal. Adaptado de [WIKIPEDIA, 2007].

2.1.1.2 Resolução

A Resolução de imagem descreve o nível de detalhe que uma imagem comporta. Quanto maior a resolução, melhor a percepção de detalhes de uma imagem [WIKIPEDIA, 2007].

A resolução pode ser medida com uma matriz de pixels (horizontal x vertical) ou em *dots per inch* (dpi – pontos por polegada).

As Figura 3 e Figura 4 nos mostram exemplos de imagens nos dois sistemas citados.

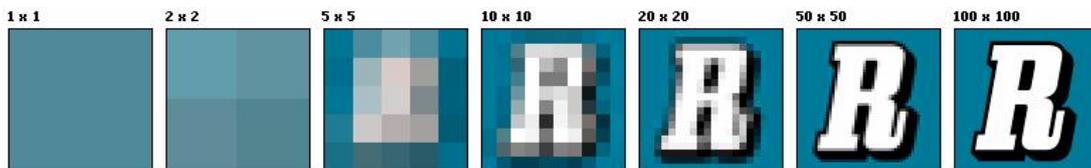


Figura 3: Exemplo de resolução em pixels. Adaptado de [WIKIPEDIA, 2007].



Figura 4: Exemplo de resolução em dpi (respectivamente 600, 300, 150, 60, 30). Adaptado de [BESSER & TRANT, 1995].

2.1.1.3 Compressão de imagem

A compressão de imagem é uma atividade de suma importância para a viabilidade do compartilhamento de imagens digitais de alta resolução na Internet, dado à largura de banda que seria necessária caso não houvesse uma compressão.

O principal foco da compressão de imagem é através da percepção visual humana eliminar valores discretizados da imagem reduzindo assim a quantidade de informação que a representa.

Existem dois tipos de compressão de imagem: Lossless (é usada quando se tem imagens de baixa resolução ou imagens que precisam de alta fidelidade com a imagem original, já que quando é realizada a descompressão a imagem original é restaurada, exemplos de padrões que a utilizam: GIF [COMPUSERVE, 2007] e PNG [W3C, 2007]) e Lossy (é usada em imagens com alta resolução e que não precisam de alta fidelidade com a imagem original, visto que ao realizar a descompressão a imagem original não é restabelecida, exemplo de padrão que a utiliza: JPEG [JPEG, 2007]).

2.1.2 Padrões de Compressão de Imagem Digital

Nesta seção iremos analisar os três principais padrões de compressão de imagens que são utilizados na Internet: JPEG, GIF e PNG.

2.1.2.1 JPEG

Padrão criado pelo grupo JPEG em 1992 [JPEG, 2007], tem 24 bits por pixel (sendo oito bits para vermelho, oito para o verde e oito para o azul), como já foi dito utiliza *lossy compression* e é o padrão mais utilizado na Internet para representar fotografias.

O JPEG tem suporte na maioria de plataformas de computador e pelos navegadores de Internet. Com compressão com perdas, a qualidade das imagens dos arquivos compactados é reduzida quando comparada com o arquivo original e não pode ser restaurada, exceto através do próprio original [JPEG, 2007].

O algoritmo de compressão do JPEG permite definir um percentual de qualidade da imagem, cabendo ao usuário escolher o melhor custo benefício entre qualidade da imagem (maior tamanho do arquivo) ou pior qualidade (menor tamanho do arquivo).

2.1.2.2 GIF (Graphics Interchange Format)

Padrão criado pelo grupo CompuServe em 1987 [COMPUSERVE, 2007], tem uma paleta de cor com 256 cores por pixel, como já foi dito utiliza *lossless compression* e ainda é um dos principais padrões para representar ícones, imagens de baixa resolução e pequenas animações.

O GIF teve um papel de fundamental importância na história da Internet já que propiciou o aparecimento de imagens (de tamanho pequeno) em épocas onde a largura de banda ainda era muito baixa.

Em 1994, houve uma disputa entre a Unisys [UNISYS, 2007] detentora da patente do algoritmo de compressão (LZW - Lempel-Ziv-Welch) e a CompuServe [COMPUSERVE, 2007] criadora do padrão, onde a Unisys obteve o direito de cobrar uma licença de uso a todos os sistemas que implementasse um codificador para GIF. Este fato é considerado um marco [WIKIPEDIA, 2007], para que surgisse um padrão concorrente livre: PNG [W3C, 2007].

2.1.2.3 PNG (Portable Network Graphics)

O padrão foi criado em 1995 pela W3C [W3C, 2007], contudo só começou a ganhar popularidade a partir do ano de 1999, para ser um concorrente livre do GIF.

O PNG permite diversas paletas de cores como: 24 bits por pixel (sendo oito bits para vermelho, oito para o verde e oito para o azul), tons de cinza e outras. Semelhantemente ao GIF, o PNG utiliza *lossless compression*, no entanto o algoritmo (*deflate*) de compressão utilizado no PNG é de uso livre.

O PNG suporta *alpha channel* que é uma técnica para combinar a imagem com o background para criar um efeito de transparência [WIKIPEDIA, 2007].

O PNG atualmente tem suporte largamente nos navegadores, sem necessidade de instalar nenhum *plugin* e está cada vez mais ocupando o lugar do GIF por apresentar melhor qualidade de imagem e melhor compressão.

2.2 *Áudio Digital*

O termo áudio digital se refere a diversos formatos digitais de armazenamento de sons. O áudio digital é a principal base para o compartilhamento de músicas na Internet. Levando em conta esse cenário, apenas realizar a transformação de áudio analógico para digital não é interessante, visto que tornaria muito custoso o armazenamento e transferência do arquivo de som. Para solucionar este problema é necessário realizar, assim como vimos em imagem digital, uma etapa de compressão de áudio.

A seguir iremos estudar as características fundamentais de áudios digitais bem como iremos analisar alguns dos atuais padrões de compressão de áudio mais usados na Internet. Dentre os padrões escolhidos temos:

- MPEG (Moving Pictures Expert Group), consideraremos os seguintes formatos da família MPEG: MPEG-1 Audio Layer III, MPEG-2 AAC e MPEG-4 AAC [MPEG, 2007];
- WMA (Windows Media Audio), Coleção de vários *codecs* proprietários da Microsoft [WMA, 2007];
- RealAudio, Formato de vídeo proprietário desenvolvido pela RealNetworks [REALNETWORKS, 2007] que é dividido em RealAudio e RealMedia; e
- Ogg Vorbis, Formato de vídeo livre desenvolvido pela Xiph.Org Foundation [XIPH, 2007].

A Figura 5 mostra um exemplo de uma transformação de um áudio analógico em digital.

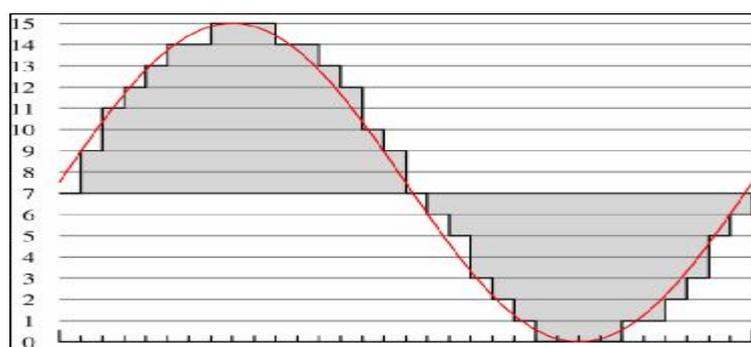


Figura 5: Áudio Analógico (Vermelho) em Digital (Cinza). Adaptado de [WIKIPEDIA, 2007].

2.2.1 Características fundamentais de áudio

Nesta seção apresentaremos algumas características importantes de áudios digitais.

2.2.1.1 Bit Rate

O conceito de bit rate em áudio e vídeo representa a quantidade de bits que são usados para codificar a informação que será transmitida em um segundo. Quanto maior bit rate, o áudio apresentará uma melhor qualidade e maior tamanho do arquivo. No Quadro 1 podemos perceber a relação do bit rate com a qualidade do áudio digital.

Quadro 1: Exemplos de bit rate em áudio digital. Adaptado de [WIKIPEDIA, 2007]

Tipo de Som	Bit Rate
Som de um telefone	8 Kbit/s
Compact Disk (CD)	1441 Kbit/s
Rádio AM (Amplitude Modulation)	32 Kbit/s
Rádio FM (Frequency Modulation)	96 Kbit/s
MP3 com qualidade similar ao CD	192 Kbit/s
MP3 com qualidade similar ao som analógico	224 - 320 Kbit/s

2.2.1.2 Sample Rate

O áudio analógico é um sinal contínuo, contudo ao ser transformado em um sinal digital que é discreto, pedaços do sinal contínuo são perdidos e a *sample*

rate, a frequência que amostras do sinal contínuo em um dado instante (*sample*) são convertidas para um sinal digital. Quanto maior a *sample rate* melhor será a qualidade do som digital, visto que terá perdido menos *samples* do som analógico. A *sample rate* é medida em Kiloherz (KHz). Por exemplo, um som digital com 44,100 kHz de *sample rate* apresenta a qualidade de um Compact Disk (CD) [GIBSON, 2006].

2.2.1.3 Compressão de Áudio

Semelhantemente à compressão de imagem, a compressão áudio é uma atividade de suma importância para a viabilidade do compartilhamento de áudio digital na Internet dado à largura de banda que seria necessária caso não houvesse uma compressão.

A compressão é realizada na etapa de codificação do áudio analógico para digital e se dá através de conceitos da psico-acústica que estuda como as pessoas percebem o som. O principal foco da compressão de áudio é remover sons que não são percebidos pelo ouvido humano, dentre estes sons podemos citar sons que estão fora do limiar da audição humana ou mesmo sons que são mascarados por outros sons.

Existem dois tipos de compressão auditiva: *lossless* (quando realizada a descompressão, o sinal original é restaurado). E o segundo tipo: *lossy* (são removidos dados do sinal original e depois de realizada a descompressão, o sinal obtido é diferente do original).

Como o nosso enfoque é em formatos de compressão utilizados na WEB, iremos estudar apenas formatos que utilizam compressão *lossy*, visto que os mesmos são usados amplamente no compartilhamento de áudio na Internet.

2.2.2 Padrões de Compressão de Áudio Digital

Nesta seção analisaremos alguns dos mais importantes padrões de compressão de áudio digital.

2.2.2.1 WMA (Windows Media Audio)

O Windows Media Audio [WMA, 2007] é um padrão proprietário desenvolvido pela Microsoft [MICROSOFT, 2007]. Atualmente, o WMA contém quatro padrões dentre os quais podemos citar dois como mais importantes: WMA Standard (segundo codificador de áudio mais utilizado em sons na WEB e principal concorrente do MP3 [MPEG, 2007]) e WMA Professional (concorrente do AAC como *codec* que apresenta sons de alta qualidade).

O padrão WMA na maioria dos casos apresenta qualidade semelhante ao MP3 utilizando metade do *bit rate* e conseqüentemente utilizando metade do tamanho de um arquivo MP3 [WMA, 2007]. Contudo, apesar dessa informação, não podemos considerar esse formato como melhor ou pior que o MP3, visto que este formato insere mais distorções em músicas que o MP3 [WIKIPEDIA, 2007].

2.2.2.2 RealAudio

O RealAudio é um padrão proprietário desenvolvido pela RealNetworks [REALNETWORKS, 2007]. A sua primeira versão foi lançada em 1995.

O principal uso desse formato se dá com a utilização de serviços de *streaming* de áudio na Internet. O RealAudio em suas primeiras versões utilizava o protocolo aberto Real Time Streaming Protocol (RTSP) criado pela Internet Engineering Task Force (IETF) [IETF, 2007] para o gerenciamento da conexão com o servidor de *stream* de áudio. No entanto, a transferência do áudio se processava através de um protocolo proprietário denominado Real Data Transport (RDT). Atualmente, o RealAudio dá suporte a *streaming* através do protocolo HTTP [REALNETWORKS, 2007].

2.2.2.3 OGG Vorbis

O OGG Vorbis é um padrão aberto e livre desenvolvido pela fundação Xiph.org, sendo parte do projeto Ogg. Começou a ser desenvolvido em 1998 e foi lançada a versão 1.0 em 2002 [XIPH, 2007].

O OGG Vorbis apresenta melhor desempenho em algumas faixas de *bit rates* que os padrões mais consolidados no mercado como MP3 e WMA, [XIPH, 2007], bem como também se mostra um concorrente à altura dos padrões mais atuais que apresentam excelente qualidade de som como AAC (MPEG-4 AAC) [MPEG, 2007] e WMA Professional [WMA, 2007].

O OGG Vorbis tem se mostrado um forte concorrente dos principais formatos e apresenta uma diferença dos demais que é ser um padrão aberto (permitindo a customização de acordo com as necessidades de desenvolvedores de formatos de compressão).

2.2.2.4 MP3 (MPEG-1 Audio Layer III)

O MP3 é um dos primeiros padrões de compressão de áudio e atualmente ainda é o padrão mais utilizado no mundo. A sua origem se deu através do trabalho de diversos pesquisadores (principalmente de universidades da Alemanha) e do grupo MPEG que formalizou o padrão.

O instituto Fraunhofer [FRAUNHOFER, 2007] e a Thomson SA [THOMSON, 2007] detém uma patente (reconhecida nos Estados Unidos e Japão) que obriga quem implementar ou distribuir codificadores ou decodificadores usando o MP3 pagar Royalties. Este foi o principal incentivo para a criação de um padrão aberto: o OGG Vorbis.

A Figura 6 representa a arquitetura geral de um codificador para o padrão MP3:

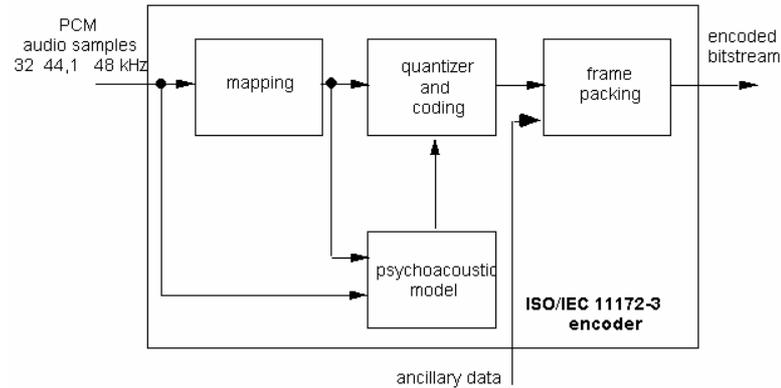


Figura 6: Codificador MP3. Adaptado de [CHIARIGLIONE, 1996].

O MP3, como já foi dito, ainda é o principal padrão de compressão de áudio utilizado para o compartilhamento de áudio na Internet. No entanto vem perdendo o seu espaço para outros formatos mais novos e eficientes como OGG Vorbis [XIPH, 2007], MPEG-4 AAC [MPEG, 2007], WMA Professional [WMA, 2007], entre outros.

2.2.2.5 MPEG-2 AAC (Advanced Audio Coding)

Este padrão também é conhecido como MPEG-2 part 3 [MPEG, 2007]. O MPEG-2 AAC (Advanced Audio Coding) foi definido como uma evolução dos padrões MPEG-1 Audio, obtendo taxas de compressão bem superiores ao seu antecessor. Esse padrão permite o uso de até 48 canais principais de áudio e 16 canais de baixa frequência [FERNANDES, 2006]. Pode manipular maiores frequências de *sampling* que o MP3 (acima de 96 KHz contra 48 KHz do MP3). Além de apresentar compatibilidade com o MP3 [MPEG, 2007].

2.2.2.6 MPEG-4 AAC (Advanced Audio Coding)

Este padrão também é conhecido por MPEG-4 part 7 [MPEG, 2007]. O MPEG-4 AAC é uma atualização do MPEG-2 AAC com três extensões: PNS (Perceptual Noise Substitution), LTP (Long Term Predictor), Twin VQ (Transform-Domain Weighted Interleave Vector Quantization) que permitem

que o MPEG-4 AAC tenha um melhor desempenho que o MPEG-2 AAC e exija um menor custo computacional para codificar o áudio [MPEG, 2007].

Atualmente, existe uma otimização do MPEG-4 AAC que apresenta os melhores índices de eficiência e se chama HE-AAC (High Definition Advanced Audio Coding) [MPEG, 2007].

2.3 Vídeo Digital

O termo vídeo digital se refere a diversos formatos de armazenamento de animações. O vídeo digital de fato foi criado quando a Sony [SONY, 2007] em 1983 lançou o formato Sony D-1 que transformava o vídeo analógico em digital sem fazer nenhuma etapa de compressão, o que tornou esse processo bastante oneroso tendo em vista o armazenamento [WIKIPEDIA, 2007].

Atualmente, esse formato ainda pode ser encontrado em grandes redes de televisão, embora o mesmo esteja em um processo de substituição por formatos que utilizam compressão e que são notavelmente mais baratos.

A seguir iremos estudar as características fundamentais de vídeos digitais bem como analisar alguns dos atuais padrões de compressão de vídeo mais usados na Internet. Dentre os padrões escolhidos temos:

- MPEG, Analisaremos os seguintes padrões da família MPEG: MPEG-1, MPEG-2 e MPEG-4 [MPEG, 2007];
- WMV, Coleção de vários *codecs* proprietários da Microsoft [WMV, 2007];
- RealVideo, padrão de vídeo proprietário desenvolvido pela RealNetworks que é dividido em RealAudio e RealMedia [REALNETWORKS, 2007], como vimos anteriormente;
- FLV, padrão de vídeo proprietário criado pela Adobe (Macromedia) [ADOBE, 2007];

- Ogg Theora, padrão de vídeo livre desenvolvido pela Xiph.Org Foundation [XIPH, 2007];
- H.263, padrão de vídeo livre criado pela ITU-T Video Coding Experts Group (VCEG) [ITU-T, 2007]; e
- H.264, padrão de vídeo livre criado pelo Joint Video Team (JVT) grupo formado com pesquisadores do ITU-T VCEG e do MPEG [ITU-T, 2007].

2.3.1 Características de Vídeo Digital

Nesta seção iremos estudar algumas características importantes de vídeos digitais.

2.3.1.1 Frame Rate (taxa de quadros)

Para se determinar o *frame rate* é utilizada a unidade FPS (Frames per Second – Quadros por segundo), a qual mede quanta informação é usada para armazenar e mostrar a animação de um vídeo [WEBOPEDIA, 2007]. Quanto maior o FPS, melhor é a qualidade do vídeo e maior é o tamanho de armazenamento.

2.3.1.2 Resolução

Em vídeo analógico a resolução é dada por linhas. Contudo, em vídeo digital a resolução é obtida através de pixels (um pixel é o menor ponto que forma uma imagem digital [WIKIPEDIA, 2007]).

A resolução é dada pela quantidade de pixels dispostos na horizontal e na vertical. Dentre as resoluções comumente usadas em monitores temos 640 x 480 (307.200 pixels), 800 x 600 (480.000 pixels) e 1024 x 768 (786.432 pixels). Na Figura 7 podemos notar a proporção entre algumas resoluções.

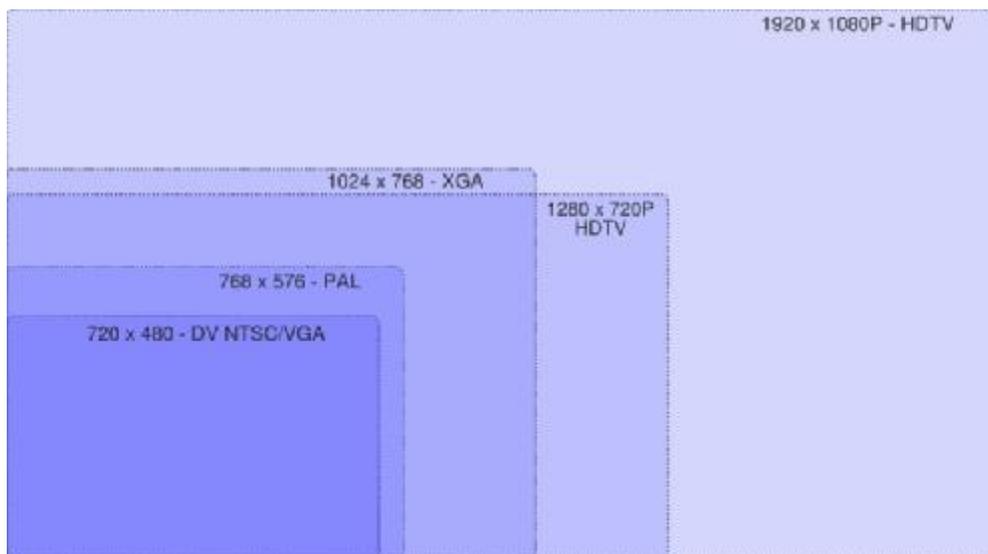


Figura 7: Proporção entre diversas resoluções. Adaptado de [WIKIPEDIA, 2007].

2.3.1.3 Aspect Ratio

Em um espaço bi-dimensional, o *aspect ratio* mede o quão distante a maior dimensão é da menor [WIKIPEDIA, 2007]. Em vídeo digital, o *aspect ratio* é aplicado para demonstrar a proporção da tela, ao se dividir a quantidade de pixels dispostos horizontalmente pela quantidade de pixels dispostos na vertical.

Os mais usados padrões de *Aspect Ratio* são:

- Padrão 4:3 ou 1.33:1 - Padrão usado pelos aparelhos de TV comuns e a grande maioria dos monitores. Este padrão foi definido pela Academia de Artes de Animação de Figuras e Ciências (Academy of Motion Picture Arts and Sciences [AMPAS, 2007]); e
- Padrão 16:9 ou 1.78:1 - Padrão internacional usado para HDTV (High-Definition Television).

Na Figura 8 podemos analisar a proporção entre alguns *aspect ratios*.

4 : 3	$\sqrt{2} : 1$	3 : 2	5 : 3	16 : 9
=	=	=	=	=
1. $\bar{3}$	1.414...	1.5	1. $\bar{6}$	1. $\bar{7}$

Figura 8: Proporção entre diversos aspect ratios. Adaptado de [WIKIPEDIA, 2007].

2.3.1.4 Qualidade de Vídeo

A medição da qualidade de um vídeo não é uma tarefa muito precisa. Segundo VIANELLO (2006) podemos dividir a avaliação da qualidade em dois tipos de abordagens:

- Objetiva, que faz uso de técnicas como *Signal-to-noise ratio* (SNR) e *Peak Signal-to-noise ratio* (PSNR), as quais através de cálculos matemáticos tentam definir o percentual de “ruído” na construção da imagem; e
- Subjetiva, a qual através da definição de métricas e pontuações, alguns especialistas observam os vídeos e atribuem notas. Ao final é realizada uma média e se obtém o resultado da avaliação da qualidade.

2.3.1.5 Bit Rate

Tem o conceito similar a bit rate de áudio, medindo o número médio de bits que um segundo de vídeo (ou áudio) pode consumir. O bit rate pode ser medido através da unidade bits / segundos.

Quanto maior for o *bit rate*, o vídeo apresentará melhor qualidade de imagem e conseqüentemente terá maior tamanho. Em média, um VideoCD apresenta 1 Mbit/s, um DVD 5 Mbit/s e uma HDTV apresenta de 15 a 100 Mbit/s [WIKIPEDIA, 2007].

2.3.1.6 Compressão de Vídeo

A compressão de vídeo é uma atividade de suma importância para a viabilidade do vídeo digital. A Figura 9 mostra o tamanho necessário para o armazenamento de um vídeo digital sem compressão.

Mídia	1 hora de áudio em qualidade de CD	1 hora de vídeo em qualidade VHS	1 hora de TV	1 hora de TV de alta definição
Espaço ocupado	635 MB	24,3 GB	97 GB	389 GB

Figura 9: Relação Mídia x Espaço Ocupado. Adaptado de [FERNANDES, 2006].

Como podemos perceber, caso não existisse a compressão de vídeo digital, a manipulação de vídeos na Internet seria uma tarefa inviável dada à largura de banda necessária.

A compressão é realizada na etapa de codificação do vídeo analógico para digital. Existem diversos formatos para a compressão de vídeos. Contudo, neste trabalho, apenas iremos abordar os formatos mais utilizados para a compressão de vídeo digital na WEB.

Assim como em imagens e áudios, temos para vídeos dois tipos de compressão, *lossless* e *lossy*. Seguindo os mesmos conceitos, a primeira apresenta uma compressão sem degradar a qualidade do vídeo e a segunda ocorre com perda de informação do vídeo original.

A seguir, iremos listar e analisar brevemente os mais importantes dos padrões de compressão de vídeo.

2.3.2 Padrões de Compressão de Vídeo Digital

Nesta seção analisaremos alguns dos mais importantes padrões de compressão de vídeos digitais.

2.3.2.1 WMV (Windows Media Video)

O Windows Media Video é um padrão proprietário desenvolvido pela Microsoft [WMV, 2007]. Em 2003 a Microsoft enviou a especificação do *codec* para

a Society of Motion Picture and Television Engineers (SMPTE) [SMPTE, 2007]. O formato se tornou aberto, contudo continuou sendo proprietário.

A grande vantagem de se utilizar esse formato é a integração dos vídeos com programas notadamente conhecidos da Microsoft, como Windows Media Player e Internet Explorer [MICROSOFT, 2007].

2.3.2.2 RealVideo

O RealVideo é um padrão proprietário desenvolvido pela RealNetworks [REALNETWORKS, 2007]. O RealVideo normalmente é usado em conjunto com o padrão de áudio RealAudio, produzido pela mesma RealNetworks. A sua primeira versão foi lançada em 1997 usando como base H.263 [H.263, 1998]. Atualmente, o RealVideo não se utiliza mais do padrão H.263. Ele tem como base agora o *codec* proprietário RV40, que foi desenvolvido também pela RealNetworks [REALNETWORKS, 2007].

O principal uso desse padrão se dá com a utilização de serviços de *streaming* de vídeo. O RealVideo utiliza o protocolo aberto Real Time Streaming Protocol (RTSP) criado pela Internet Engineering Task Force (IETF) [IETF, 2007] para o gerenciamento da conexão com o servidor de *stream* de vídeo. No entanto, a transferência do vídeo se processa através de um protocolo proprietário denominado Real Data Transport (RDT), impedindo assim que tocadores de outras empresas, que não tenham suporte a RDT *stream*, reproduzam o vídeo.

2.3.2.3 FLV (Flash Live Video)

O Flash Live Video é um padrão proprietário desenvolvido pela Adobe [ADOBE, 2007]. O Flash Live Video em versões anteriores à versão oito usava como *codec* uma variante do H.263 [H.263, 1998]. No entanto, as mais novas versões do Flash Vídeo, além da variante do H.263, usam o *codec* On2 True Motion VP6 [ON2, 2007], o qual apresenta uma maior compressão que o H.263.

O Flash Video é o formato mais utilizado pelos serviços de manipulação de vídeos na Internet por apresentar bons índices de compressão e qualidade da imagem, além de se privilegiar do fato da Adobe [ADOBE, 2007] ter um *plugin* (*Adobe Flash Player*) que permite que vídeos neste formato possam ser vistos diretamente nos navegadores (*browsers*) dos usuários.

2.3.2.4 H.263

O H.263 é um padrão de compressão desenvolvido pela ITU-T Video Coding Experts Group (VCEG) [ITU-T, 2007] em 1995, com o foco na utilização em videoconferências.

Atualmente, esse formato ainda é usado em videoconferências baseadas nos protocolos RTP (Real-time Transport Protocol), RTSP (Real Time Streaming Protocol) e SIP (Session Initiation Protocol). Também é utilizado nos conteúdos de vídeos no formato Flash (FLV), tendo um papel importante nos sistemas de compartilhamento de vídeos como YouTube [YOUTUBE, 2007], GoFish [GOFISH, 2007], Metacafe [METACAFE, 2007], Google Video [GOOGLE VIDEO, 2007], entre outros.

2.3.2.5 OGG Theora

O OGG Theora é um formato livre desenvolvido pela fundação Xiph.org [XIPH, 2007], sendo parte do projeto Ogg. É baseado no *codec* VP3 (que era proprietário e equivalente em qualidade ao MPEG-1) e que foi doado pela On2 Technologies [ON2, 2007] ao lançar em domínio público todas as licenças que possuía do VP3. Assim, a Xiph.org [XIPH, 2007] pôde desenvolver o OGG Theora usando como base o VP3.

O OGG Theora é comumente usado com o OGG Vorbis, responsável pela camada de áudio. OGG Theora é um formato recente, contudo tem se mostrado

um concorrente à altura dos demais padrões já consolidados no mercado [XIPH, 2007].

2.3.2.6 MPEG-1

O padrão MPEG-1 especifica a codificação de áudio e vídeo para entrega à taxa de 1,5 MBps, para aplicações em CD-ROM e desktop [GOULART, 2003].

O MPEG-1 é dividido em duas partes: MPEG-1 Video e MPEG-1 Audio (que iremos analisar na seção de áudio digital) como podemos perceber através da Figura 10.

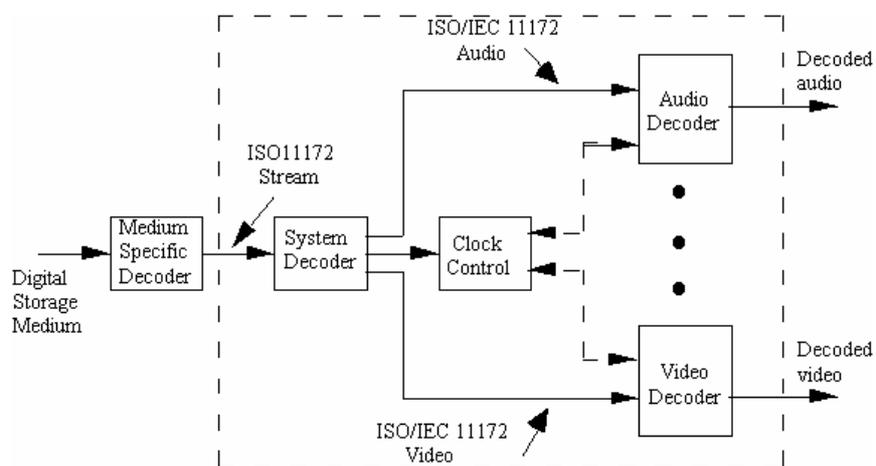


Figura 10: Decodificador MPEG-1. Adaptado de [CHIARIGLIONE, 1996].

O MPEG-1 Video foi projetado para o armazenamento de vídeos de até 1.5 MBps e com resolução 352x240 *pixels* e *frame rate* de 30fps ou resolução 352x288 pixels e frame rate de 25fps [WIKIPEDIA, 2007].

Este padrão foi desenvolvido com o intuito de armazenar vídeos em CD (VCD) propiciando uma qualidade de imagem similar a fitas VHS (Video Home System).

2.3.2.7 MPEG-2

O padrão MPEG-2 especifica a codificação genérica de sinais de vídeo, incluindo TV digital. MPEG-1 e MPEG-2 estão intimamente ligados ao sucesso de produtos e serviços comerciais largamente adotados, como gravadores e reprodutores para Video CD, DVD, difusão de áudio digital (Digital Audio Broadcasting – DAB [DAB,2007]) e MP3 (MPEG-1 Audio Layer III [MPEG, 2007]) [GOULART, 2003].

O padrão MPEG-3 seria utilizado para a padronização de HDTV, mas foi cancelado e suas funcionalidades foram incorporadas ao padrão MPEG-2.

O comitê desenvolvedor do MPEG-2 compreendeu que, para atender aos diferentes requisitos de todas as aplicações-alvo, era necessário criar uma série de níveis e perfis. Esses níveis e perfis facilitam a interoperabilidade entre aplicações que lidam com o padrão MPEG-2, pois especificam detalhes com os quais essas aplicações necessitam estar em conformidade. Os perfis definem diferentes esquemas de codificação; os níveis se referem principalmente à resolução do vídeo produzido [FERNANDES, 2006].

A mistura de perfis e níveis permite que o MPEG-2 se torne escalável, podendo ser usado em padrões de TV digital (alta resolução) ou em vídeos de baixa resolução.

2.3.2.8 MPEG-4

O padrão MPEG-4, tem como objetivo principal definir um padrão internacional para a codificação audiovisual que permita serviços de interação [MPEG, 2007].

O MPEG-4 tem uma codificação eficiente de vídeo e áudio. Além disso, este padrão permite a representação e manipulação individual de cada objeto (vídeo, áudio, textos, gráficos, entre outros) que compõe uma cena [MPEG, 2007].

Esta divisão dos objetos que compõem uma cena permite prover uma melhor interação entre o usuário e um determinado vídeo, ao contrário dos padrões MPEG-1 e MPEG-2, que apenas permitem a seqüência de quadros de imagem.

O padrão MPEG-4 também especifica meios de permitir a reutilização de objetos de mídia e a associação de direitos autorais [BATTISTA et al.,1999].

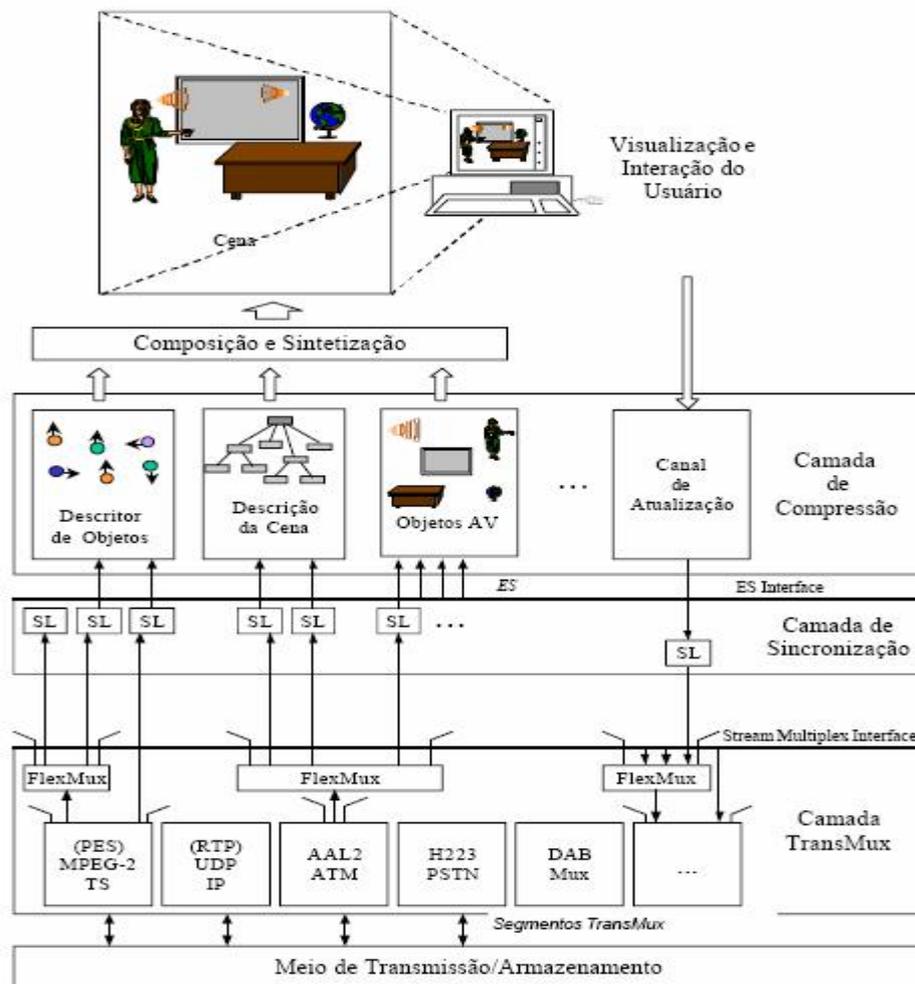


Figura 11: Componentes do MPEG-4. Adaptado de [TODESCO, 2000].

O padrão MPEG-4 com a sua divisão dos objetos de mídia, como podemos perceber na Figura 11, possibilita uma codificação ótima para cada objeto. Dessa forma, o padrão obtém um alto índice de compressão possibilitando uma alta qualidade de vídeo. Com isto, o MPEG-4 obteve uma grande aceitação no

mercado de vídeo digital. Atualmente, este padrão evoluiu com os esforços do Joint Video Team (JVT) [JVT, 2007] e se chama H.264 (MPEG-4 part 10).

2.3.2.9 H.264

O padrão H.264 é também conhecido como H.264/MPEG-4 AVC e MPEG-4 (part 10). O H.264 surgiu como a união de especialistas de dois dos mais importantes grupos de padronização de vídeo digital: ITU-T VCEG [ITU-T, 2007] e MPEG [MPEG, 2007], que formaram um grupo chamado Joint Video Team (JVT) [JVT, 2007].

O padrão H.264 é considerado a nova geração de compressão de vídeo usando MPEG-4 [APPLE, 2007]. O H.264 apresenta altos índices de qualidade de vídeo em uma grande variação de largura de banda (de 40Kbps até acima de 10Mbps).



Figura 12: Usos do H.264. Adaptado de [APPLE, 2006].

O H.264 está se consolidando cada vez mais no mercado como o principal padrão de compressão de vídeo por apresentar excelentes índices de compressão, aliado ao fato de poder atuar em toda a largura de banda, fato notado com a Figura 12, podendo ser usado em dispositivos móveis (3G) até dispositivos que apresentam alta-definição de imagem (HD).



Figura 13: Comparação entre H.264 e MPEG4 part 2. Adaptado de [APPLE, 2006].

Como podemos perceber, através da Figura 13, o H.264 apresenta o tamanho do quadro (*Frame*) quatro vezes maior que o MPEG-4 (part 2) utilizando o mesmo tamanho para o armazenamento.

Os impressionantes resultados obtidos pelo H.264 fizeram que ele fosse adotado em diversas organizações. Dentre elas podemos citar: MPEG [MPEG, 2007], ITU-T VCEG [ITU-T, 2007], DVD Forum [DVD, 2007], Blu-ray Disc Association [BDA, 2007], Digital Video Broadcasting [DVB, 2007], 3rd Generation Partnership Project [3GPP, 2007] e Internet Streaming Media Alliance [ISMA, 2007].

O padrão H.264 está se consolidando cada vez mais como principal padrão de compressão de vídeo. O fato do mesmo poder atuar em uma grande variação de largura de banda é de extrema importância para que algum dia possamos ter uma unificação dos padrões de compressão de vídeo digital.

2.4 Considerações Finais

Dentre os padrões de compressão de imagens digitais, apontamos o PNG como padrão que em um curto espaço de tempo irá substituir o padrão GIF, por apresentar melhores taxas de compressão e possibilitar imagens com 24 bits de cores por pixel contra apenas 256 do GIF. Para o padrão JPEG não encontramos um padrão de compressão com perdas de informação que ameace a posição do JPEG como padrão mais utilizado para a compressão de fotografias de alta resolução.

Dentre os padrões de áudios digitais temos o MP3 e o WMA como padrões mais utilizados para o compartilhamento de música na WEB, contudo notamos que outros padrões com altas taxas de compressão, MPEG-2 AAC, MPEG-4 AAC, WMA Professional, OGG Vorbis estão ganhando cada vez mais popularidade.

Dentre os padrões de vídeos ressaltamos a família MPEG e WMV como padrões mais utilizados, contudo é notório o crescimento do OGG Theora, padrão livre, como também é de fundamental importância destacar o H.264 que como mostramos apresenta excelentes resultados de compressão e pode ser usado em uma grande largura de banda. Apontamos também H.264 como o padrão de compressão de vídeo mais promissor dentre os estudados.

3. Pesquisa comparativa de Serviços WEB de manipulação de objetos multimídia

Neste capítulo realizaremos uma pesquisa comparativa com os serviços mais usados de manipulação de imagens, áudios e vídeos disponíveis na Internet com o intuito de estabelecermos as funcionalidades básicas que o nosso módulo de manipulação de objetos multimídia deva dar suporte.

3.1 Dados comparativos

Nesta seção, iremos agrupar e discutir alguns dados coletados sobre alguns dos principais serviços de manipulação de imagens, áudios e vídeos digitais encontrados na Internet.

3.1.1 Serviços de Imagem Digital

Nesta seção iremos analisar dois serviços de gerenciamento de álbuns largamente utilizados: Flickr [FLICKR, 2007] e Picasa [PICASA, 2007] e dois serviços de Fotologs (registro publicado na Internet com fotos colocadas em ordem cronológica, ou apenas inseridas pelo autor sem ordem, de forma parecida com um Blog [WIKIPEDIA, 2007]): Terra Fotolog [TERRA, 2007] e UOL Fotoblog [UOL, 2007]. No Quadro 2, apresentamos alguns resultados obtidos com uma comparação, na etapa de inserção de imagens, dos serviços escolhidos.

Quadro 2: Comparação na etapa de inserção de imagens.

Serviço	Padrões suportados	Tamanho máximo	Inserção por software externo	Inserção por e-mail
Flickr	Qualquer formato de imagem*	5 MB	Sim	Sim
Picasa	JPEG	Sem limite	Sim	Não
Terra Fotolog	Qualquer formato de imagem**	Sem limite	Não	Sim
Uol Fotoblog	JPEG	Sem limite	Não	Não

Observações:

* O Flickr sugere o uso dos seguintes padrões: JPEG, GIF, PNG, contudo quando o usuário insere uma imagem em outro formato ele internamente converte a imagem para JPEG.

** O Terra Fotolog sugere o uso de JPEG, contudo quando o usuário insere uma imagem em outro formato ele internamente converte a imagem para JPEG.

Analisando os dados mostrados no Quadro 2, decidimos que os nossos protótipos devem restringir a inserção de imagens aos padrões JPEG, GIF e PNG, a fim de não termos que aumentar o processamento do servidor de dados multimídia com aplicações para converter padrões de compressão. Resolvemos também não limitar o tamanho dos arquivos de imagens, uma vez que o nosso público-alvo, professores, pode querer utilizar imagens com altas-definições. E por fim não julgamos agregar valor criar softwares externos ao sistema para realizar inserções de imagens nem permitir a inserção por e-mail. No Quadro 3, apresentamos alguns resultados obtidos com uma comparação, na etapa de consulta de imagens, dos serviços escolhidos.

Quadro 3: Comparação na etapa de consulta de imagens.

Serviço	Padrões de compressão suportados	Tags	Data de Inserção	Licença	Nome	Autor	Descrição
Flickr	Não	Sim	Sim	Sim	Não	Sim	Sim
Picasa	Não	Não	Não	Não	Sim	Sim	Não
Terra Fotolog	Não	Não	Não	Não	Não	Sim	Não
Uol Fotoblog	Não	Sim	Não	Não	Não	Sim	Não

Observando os dados mostrados no Quadro 3, decidimos que os nossos protótipos devem implementar todos os tipos de consultas listadas, além de permitir consultas pela resolução e tamanho de armazenamento da imagem.

3.1.2 Serviços de Áudio Digital

Nesta seção do trabalho iremos analisar três serviços de compartilhamento de arquivos em geral, que são largamente utilizados para o compartilhamento de áudio digital: 4Shared [4SHARED, 2007], DivShare [DIVSHARE, 2007] e MediaMax [MEDIAMAX, 2007]. No Quadro 4, apresentamos alguns resultados obtidos com uma comparação, na etapa de inserção de áudios, dos serviços escolhidos.

Quadro 4: Comparação na etapa de inserção de áudios.

Serviço	Padrões suportados	Tamanho máximo	Inserção por software externo
4Shared	MP3, WMA, WAV, MID, RA OGG	50 MB	Não
DivShare	MP3	Sem limite	Sim
MediaMax	MP3, WMA	Sem limite	Sim

Analisando os dados mostrados no Quadro 4, decidimos que os nossos protótipos devem restringir a inserção de áudio aos padrões MP3, WMA e OGG, uma vez que o primeiro é indiscutivelmente o padrão mais utilizado na Internet, o segundo é o seu principal concorrente e o terceiro é um padrão livre e que vem crescendo cada vez mais. Resolvemos, novamente, não limitar o tamanho dos arquivos de áudio, similarmente à justificativa de não limitar o tamanho de imagens, já que os professores, nosso público-alvo, podem querer inserir áudio aulas de diferentes tamanhos. E por fim, mais uma vez, não julgamos agregar valor criar softwares externos ao sistema para realizar inserções de áudios. No Quadro 5, apresentamos alguns resultados obtidos com uma comparação, na etapa de consulta de áudios, dos serviços escolhidos.

Quadro 5: Comparação na etapa de consulta de áudios.

Serviço	Padrões Suportados	Tags	Descrição	Tamanho de arquivo	Autor
4Shared	Sim	Sim	Sim	Sim	Sim
DivShare	Não	Não	Não	Não	Sim
MediaMax*	Não	Não	Não	Não	Não

Observação: * O MediaMax não possui sistema de busca, para realizar o compartilhamento o usuário fornece o endereço do seu repositório virtual de arquivos.

Observando os dados mostrados no Quadro 5, estabelecemos que os nossos protótipos implementem todos os tipos de consultas listadas, além de permitir a consultas por: licença, padrão de compressão, *sample rate*, *bit rate*, duração e tamanho de armazenamento do áudio.

3.1.3 Serviços de Vídeo Digital

Nesta seção do trabalho iremos analisar sete dos serviços de manipulação de vídeos largamente utilizados na Internet: YouTube [YOUTUBE, 2007], Google Video [GOOGLE, 2007], Metacafe [METACAFE, 2007], Broadcaster [BROADCASTER, 2007], Dailymotion [DAILYMOTION, 2007], vMix [VMIX, 2007] e Gofish [GOFISH, 2007]. No Quadro 6, apresentamos alguns resultados obtidos com uma comparação, na etapa de inserção de vídeos, dos serviços escolhidos.

Quadro 6: Comparação na etapa de inserção de vídeos.

Serviço	Padrões suportados	Tamanho máximo	Inserção por software externo
YouTube	Sem limitação*	100MB	Não
Google Video	Sem limitação*	100MB	Sim
Metacafe	Sem limitação*	100MB	Sim
Broadcaster	Sem limitação*	150MB	Sim
Dailymotion	Sem limitação*	150MB	Não
vMix	AVI, família MPEG, WMV	350MB	Sim
Gofish	Sem limitação*	100MB	Não

Observação: * Todos os sistemas, após a inserção do vídeo, internamente, fazem a conversão para o padrão de compressão FLV e mostram o vídeo para o usuário através do plugin Flash Player [ADOBE, 2007] que exibe o vídeo diretamente no navegador.

Analisando os dados mostrados no Quadro 6, consideramos a importância de construir os protótipos de forma que seja uma tarefa simples estender a lista de padrões de compressão com suporte. Dada a limitação de tempo, decidimos para a atual versão do protótipo permitir os padrões: família MPEG, WMV, AVI

e OGG. Os três primeiros são os padrões proprietários mais utilizados e o último é o padrão livre que está a cada dia ganhando mais espaço.

E mais uma vez não limitamos o tamanho dos arquivos de vídeo, pelos mesmos motivos de imagens e áudios, como também novamente não julgamos relevante criar softwares externos ao sistema para realizar inserções de vídeos. No Quadro 7, apresentamos alguns resultados obtidos com uma comparação, na etapa de consulta de vídeos, dos serviços escolhidos.

Quadro 7: Comparação na etapa de consulta de vídeos.

Serviço	Palavras-Chave	Autor	Título	Descrição	Categoria	Duração	Ranking Popular
YouTube	Sim	Sim	Sim	Sim	Sim	Não	Sim
Google Video	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Metacafe	Sim	Não	Sim	Sim	Sim	Não	Sim
Broadcaster	Sim	Não	Sim	Sim	Sim	Não	Sim
Dailymotion	Sim	Sim	Sim	Sim	Sim	Não	Sim
vMix	Sim	Sim	Sim	Sim	Sim	Não	Sim
Gofish	Sim	Sim	Sim	Sim	Sim	Não	Sim

Observando os dados mostrados no Quadro 7, decidimos que os nossos protótipos devem implementar todos os tipos de consultas listadas, além de permitir consultas por: licença, padrão de compressão, *frame rate*, *bit rate*, duração, resolução e tamanho de armazenamento do vídeo. Ressaltamos que, nessa pesquisa, o ranking popular não será implementado, por restrições temporais.

3.2 Considerações Finais

Não apresentamos tabelas de comparação com atualização e remoção já que todos os serviços apresentam as mesmas opções de apenas atualizar as informações textuais e permitir a remoção dos objetos.

Após a realização da pesquisa comparativa, percebemos que de modo geral, todos os serviços de multimídia utilizam em suas consultas comparações de informações textuais passadas na etapa de inserção do objeto multimídia.

Também notamos uma carência, em todos os serviços analisados, de consultas avançadas através de características fundamentais dos objetos multimídia, tais como: resolução na busca de imagens e *bit rate* na busca de áudio e vídeos. Para sanar essa limitação, nos nossos protótipos propomos a opção de uma consulta avançada contemplando as características dos objetos.

Esta pesquisa foi de suma-importância para a definição do escopo dos protótipos, visto que pudemos analisar detalhadamente os serviços de manipulação de objetos multimídias mais usados na Internet. Além disso, tivemos a oportunidade de extrair as melhores características de cada serviço, bem como observar os seus pontos-fracos.

4. Análise dos Protótipos

A prototipagem de sistemas de software pretende ser usada, principalmente, para animar e demonstrar os requisitos de um sistema. Pretende-se, assim, ajudar os clientes e os responsáveis pelo desenvolvimento do projeto a testar e a melhorar o sistema antes mesmo de este estar finalizado [ELKOUTBI et al., 1999].

Lembramos que um dos objetivos dessa pesquisa é indicar uma possível arquitetura de manipulação de dados multimídia, do sistema de gerenciamento de banco de dados (SGBD) até a visualização do objeto multimídia no navegador do usuário.

Usando as informações obtidas durante a pesquisa realizada no capítulo anterior, neste capítulo iremos definir o ambiente de desenvolvimento, o escopo e estrutura dos protótipos que representam três abordagens tradicionais de manipulação de objetos multimídia no PostgreSQL [POSTGRESQL, 2007], SGBD escolhido, por apresentar a robustez necessária para manipulação de objetos multimídia, por ser o sistema utilizado no AMADEUS_MM e, principalmente, por condizer com a filosofia de software livre [FSF, 1985] assumida pelo projeto AMADEUS_MM.

E para finalizar proporemos um quarto protótipo que representa uma solução híbrida entre diferentes abordagens tradicionais que, de acordo com os testes realizados, mostrou ser uma possível abordagem a ser adotada para a criação de um sistema de manipulação de objetos multimídia para o projeto AMADEUS_MM.

4.1 Ambiente de Desenvolvimento

O ambiente de desenvolvimento foi o NetBeans [NETBEANS, 2007] por ser um software de uso gratuito e fornecer inúmeros recursos que aumentam a produtividade durante o processo de codificação de sistemas.

Como linguagem de programação escolhemos JAVA 5 [JAVA, 2007] com a tecnologia para desenvolvimento WEB JSP (Java Server Pages) [JSP, 2007], utilizando o servidor WEB: Apache Tomcat [TOMCAT, 2007] e, como já foi dito, o SGBD escolhido foi o PostgreSQL [POSTGRESQL, 2007].

As escolhas de JAVA 5 [JAVA, 2007], JSP [JSP, 2007], Apache Tomcat [TOMCAT, 2007] e PostgreSQL [POSTGRESQL, 2007] se deram pelo fato do AMADEUS_MM utilizar essas tecnologias e nós termos o propósito de futuramente integrar o protótipo de manipulação de objetos multimídia final com esse ambiente de ensino a distância.

Como ferramentas de apoio à documentação dos protótipos utilizamos o JUDE Commuty [JUDE, 2007] e o brModelo [BRMODELO, 2007].

4.2 Escopo dos Protótipos

Todos os serviços de manipulação de objetos multimídia estudados possuíam cinco funcionalidades básicas: inserção, recuperação, atualização, remoção e visualização do objeto diretamente no navegador.

Em todos os serviços pesquisados, a inserção se dá pelo preenchimento de algumas informações textuais que descrevem o objeto seguido da tarefa de enviar o objeto para o servidor responsável pelo armazenamento, e de propriedade do serviço de manipulação de dados multimídia. Esta funcionalidade foi escolhida para os protótipos semelhantemente aos tradicionais serviços.

A recuperação dos objetos em todos os serviços pesquisados se dá através das informações textuais que descrevem cada objeto. Para os nossos protótipos também escolhemos este tipo de recuperação por restrições de tempo, para testarmos e desenvolvermos algoritmos que realizem consultas através do conteúdo dos objetos.

Em todos os serviços pesquisados, a atualização se dá apenas nas informações textuais do vídeo, não sendo permitido atualizar o objeto multimídia em si. Resolvemos seguir este padrão nos protótipos, porque não achamos que a relação custo benefício propicia a eliminação desta restrição, visto que esta tarefa aumentaria o grau de complexidade dos protótipos e não agregaria valor aos usuários que já estão acostumados com essa restrição.

A remoção de objetos multimídia é permitida em todos os tradicionais serviços e não poderia deixar de existir nos protótipos, contudo a fim de facilitar a tarefa do usuário decidimos que o mesmo pudesse realizar uma remoção de vários objetos ao mesmo tempo.

Quanto à visualização dos objetos multimídia, esta funcionalidade foi a que os nossos protótipos apresentam uma diferença fundamental dos serviços pesquisados. Esta diferença não se dá na visualização de imagem, já que nos protótipos essa visualização é similar aos serviços estudados, uma vez que todos os navegadores, sem precisar de *plugin* adicional, apresentam para os usuários as imagens nos formatos JPEG [JPEG, 2007], GIF [GIF, 2007] e PNG [PNG, 2007].

No entanto, a diferença radical citada surge na reprodução de áudios e vídeos dos serviços pesquisados com os protótipos, uma vez que todos os serviços realizam uma etapa de conversão dos áudios e vídeos para o padrão FLV [ADOBE, 2007] e os apresentam aos usuários através do *plugin* Adobe Flash Player [ADOBE, 2007].

Decidimos, para diminuir a carga de processamento do servidor de dados multimídia, eliminar esta etapa de conversão para o padrão FLV e através do navegador invocarmos o *Player default* do usuário para exibir o áudio ou o vídeo, cabendo a esse *Player* realizar o Streaming (tecnologia que permite o envio de

informação multimídia através de pacotes, utilizando redes de computadores, sobretudo a Internet [WIKIPEDIA, 2007]) do objeto.

O benefício de se remover essa etapa de conversão para FLV é diminuir o *overhead* do servidor que realiza essa tarefa. É imperativo o uso de algum software adicional para que o usuário reproduza o vídeo ou áudio diretamente no seu navegador. Então decidimos trocar o *Adobe Flash Player* pelo *Player Default* do usuário e remover esta etapa que causava essa carga de processamento.

Desta forma, os nossos protótipos realizam as cinco funcionalidades principais dos serviços de manipulação de dados multimídia pesquisados, no entanto com algumas customizações que julgamos importantes.

4.3 Diagrama de Classes

Como padrão de projeto decidimos utilizar uma adaptação do MVC, padrão de projeto que define a separação dos interesses em um programa onde o modelo (model) define a estrutura interna dos dados do programa, a visão (view) define como o modelo é renderizado para o usuário, e o controlador (controller) executa as ações no programa que afetam o modelo (model) [GAMMA et al., 1994]. Isto porque ao invés de utilizarmos um controlador central, decidimos utilizar uma camada com vários controladores, já que os mesmos são poucos e isso irá diminuir um nível de delegação de métodos e melhorar a performance. Contudo, sabemos que na versão final do sistema teremos vários controladores, daí teremos que realizar testes e definir definitivamente a necessidade ou não de um controlador central.

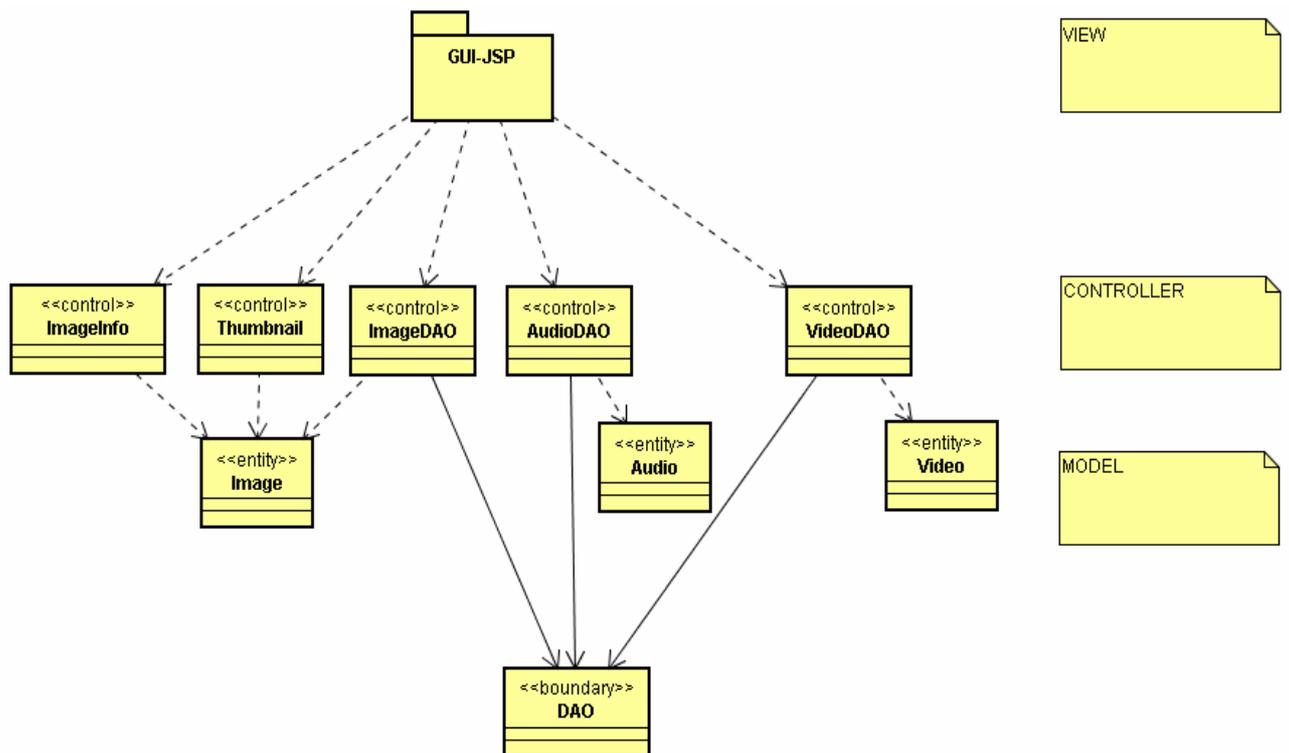


Figura 14: Diagrama de Classes dos Protótipos.

Todos os protótipos desenvolvidos apresentam a mesma estrutura de classes, representada na Figura 14, com algumas diferenças nos tipos dos atributos que representam o objeto multimídia nas classes de entidade, já que cada protótipo manipula os objetos multimídia de forma diferente no SGBD.

Na camada de MODEL temos as classes: Image, Audio e Video que são classes persistentes (são mapeamentos da tabelas do banco de dados) e a classe DAO responsável por gerir a conexão com o banco e gerenciar a execução de operações no SGBD.

Na camada de CONTROLLER temos as classes: ImageDAO, AudioDAO e VideoDAO que realizam respectivamente as operações de manipulação de imagens, áudios e vídeos. Ainda nessa camada temos as classes Thumbnail (gera um *thumbnail* de uma imagem antes dela ser inserida no banco) e a classe ImageInfo que extrai o padrão de compressão da imagem e a sua resolução.

Na camada de VIEW temos arquivos JSP que são responsáveis por criar a interface gráfica do sistema com o usuário.

A seguir iremos detalhar cada protótipo, uma vez que cada um utiliza uma abordagem de armazenamento e consulta de objetos multimídia diferente.

4.4 Protótipo 1

Protótipo 1 consiste na implementação do diagrama de classes, apresentado na Figura 14, juntamente com a opção de armazenamento de objetos multimídia presente no PostgreSQL [POSTGRESQL, 2007] chamada *Large Object*. Para usar esta opção do SGBD, o usuário define o campo que irá receber o objeto com o tipo “*lo*”.

O conteúdo de um campo de tipo “*lo*” é armazenado em uma tabela interna do PostgreSQL [POSTGRESQL, 2007]. Na tabela do usuário apenas é guardado um inteiro que é um apontador para o objeto armazenado na tabela interna citada anteriormente.

O carregamento do objeto multimídia se dá por uma função interna do SGBD, “*lo_import*”, passando-se o caminho do arquivo. Para se recuperar um objeto multimídia armazenado, com essa abordagem, é necessário invocar a função interna do SGBD, “*lo_export*”, que faz o caminho inverso da primeira. Para tanto, passamos o nome do campo, de tipo “*lo*”, e o local para onde o arquivo deve ser exportado.

Com a Figura 15, podemos visualizar a arquitetura do Protótipo 1.

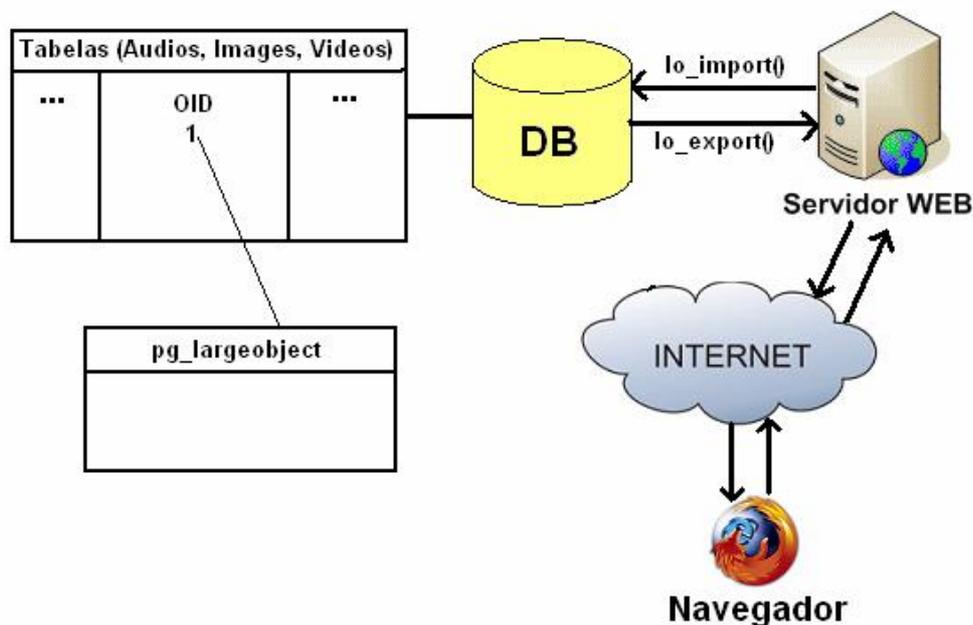


Figura 15: Arquitetura do Protótipo 1.

Essa abordagem acarreta na necessidade do **upload** do arquivo para um servidor WEB antes de realizar a inserção no SGBD. Bem como, também implica na necessidade de que antes que o usuário possa visualizar o objeto multimídia, deva acontecer à exportação do objeto do SGBD para uma pasta temporária no servidor WEB.

Após esta exportação, o Protótipo 1, através do uso das **tags** “*img*” e “*embed*” do HTML (HyperText Markup Language) [W3C, 2007] permite que o usuário final da aplicação possa visualizar os objetos multimídia diretamente no seu navegador, para o caso de imagens, ou com o auxílio do seu **Player Default** que fica embutido no navegador para os casos dos objetos multimídia serem áudio ou vídeo.

Não podemos deixar de citar que essa abordagem possibilita ao sistema como um todo um controle sobre os objetos multimídia, visto que ao se armazenar o objeto no banco, tem-se todas as vantagens providas por um SGBD como: integridade, restrição de acesso e controle de concorrência.

4.4.1 Estrutura do Banco de dados

Images	Audios	Videos
* id: int	* id: int	* id: int
description: text	description: text	description: text
name: text	name: text	name: text
author: text	author: text	author: text
tags: text	tags: text	tags: text
image: lo	audio: lo	video: lo
thumbnail: lo		thumbnail: lo
license: char	license: char	license: char
width: int	length: int	width: int
height: int	dateinsertion: date	height: text
length: int	extension: text	length: int
dateinsertion: date	samplerate: int	dateinsertion: date
extension: text	bitrate: int	extension: text
	duration: int	bitrate: int
		framerate: float
		duration: int
		standard: text

Figura 16: Tabelas do Protótipo 1.

A Figura 15 nos mostra as tabelas que compõem a base de dados do Protótipo 1. Podemos notar que os campos *image*, *thumbnail* (tabela *Images*, representa a imagem em uma resolução de 160 x 120 pixels), *audio*, *video* e *thumbnail* (tabela *Videos*, representa o primeiro frame do vídeo em uma resolução de 160 x 120 pixels) são do tipo “lo”.

4.5 Protótipo 2

Similarmente ao Protótipo 1, o Protótipo 2 consiste também na implementação do diagrama de classes, mostrado na Figura 14, juntamente com uma abordagem de armazenamento e manipulação de objetos multimídia conhecida por armazenamento por referência (esta abordagem apenas armazena no SGBD o caminho do objeto multimídia no sistema operacional) [DA SILVA, 2006].

Esta abordagem notoriamente tem melhor performance que abordagens que guardem o objeto multimídia no SGBD. No entanto, pode facilmente

apresentar sérios problemas de integridade, uma vez que a restrição do acesso aos objetos multimídia é controlada pelo sistema operacional e não pelo SGBD, sendo possível que algum usuário do servidor possa acidentalmente ou de forma mal intencionada apagar os objetos multimídia, gerando assim uma grave inconsistência na base de dados.

Com a Figura 17, podemos visualizar a arquitetura do Protótipo 2.

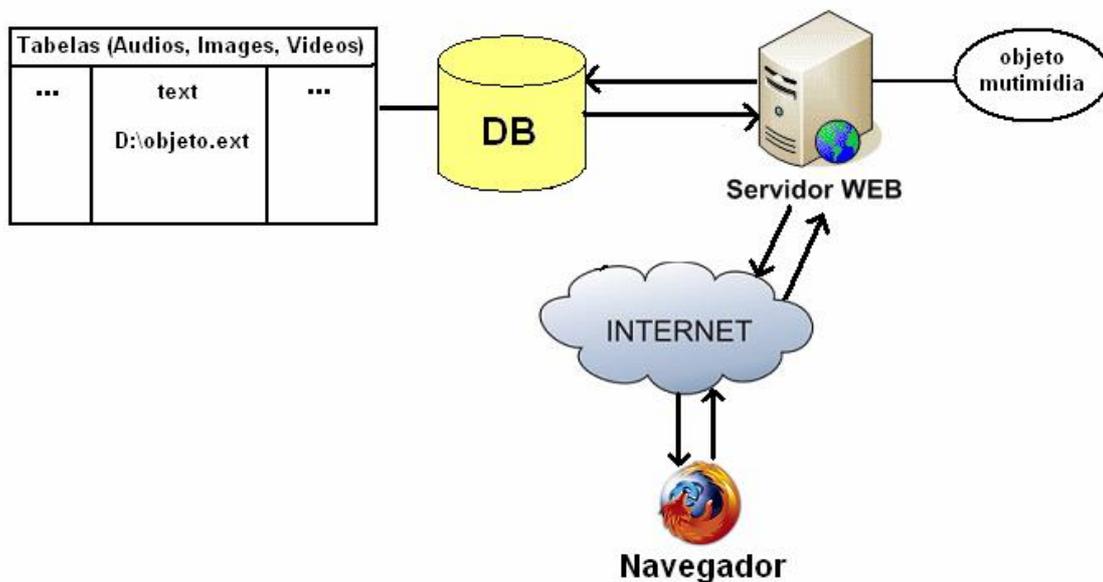


Figura 17: Arquitetura do Protótipo 2.

Com essa abordagem, a visualização dos objetos multimídia é feita de forma similar ao Protótipo 1, usando as tags "*img*" e "*embed*" do HTML.

Outro fato, que não podemos deixar de mencionar, é que com esta abordagem a permissão de inserção de objetos multimídia fica restrita aos usuários locais do servidor, uma vez que, apenas estes usuários por questão de segurança devem estar habilitados para escrever na pasta que será responsável por guardar os objetos multimídia.

4.5.1 Estrutura do Banco de dados

Images	Audios	Videos
* id: int	* id: int	* id: int
description: text	description: text	description: text
name: text	name: text	name: text
author: text	author: text	author: text
tags: text	tags: text	tags: text
image: text	audio: text	video: text
thumbnail: text		thumbnail: text
license: char	license: char	license: char
width: int	length: int	width: int
height: int	dateinsertion: date	height: int
length: int	extension: text	length: int
dateinsertion: date	sample rate: int	dateinsertion: date
extension: text	bitrate: int	extension: text
	duration: int	bitrate: int
		framerate: float
		duration: int
		standard: text

Figura 18: Tabelas do Protótipo 2.

A Figura 16 nos mostra as tabelas que compõem a base de dados do Protótipo 2. Podemos notar que diferentemente do Protótipo 1, os campos *image*, *thumbnail* (tabela *Images*), *audio*, *video* e *thumbnail* (tabela *Videos*) são do tipo “text”, já que nestes campos são guardados os caminhos dos objetos no sistema operacional.

4.6 Protótipo 3

Da mesma forma que os protótipos anteriores, o Protótipo 3 consiste na implementação do diagrama de classes, apresentado na Figura 14, juntamente com uma outra forma de armazenamento de objetos multimídia. Neste caso, a abordagem escolhida é o armazenamento de arquivos binários.

O PostgreSQL [POSTGRESQL, 2007] permite o armazenamento da seqüência binária de um objeto multimídia. Para isso, o SGBD fornece o tipo “bytea”.

Esta forma de armazenamento de objetos multimídia permite que a abordagem do Protótipo 3 se diferencie da abordagem do Protótipo 1 por não exigir a necessidade de se reconstruir o objeto, em uma pasta temporária no servidor, para que o usuário possa visualizar o objeto multimídia recuperado.

Com a Figura 19, podemos visualizar a arquitetura do Protótipo 3.

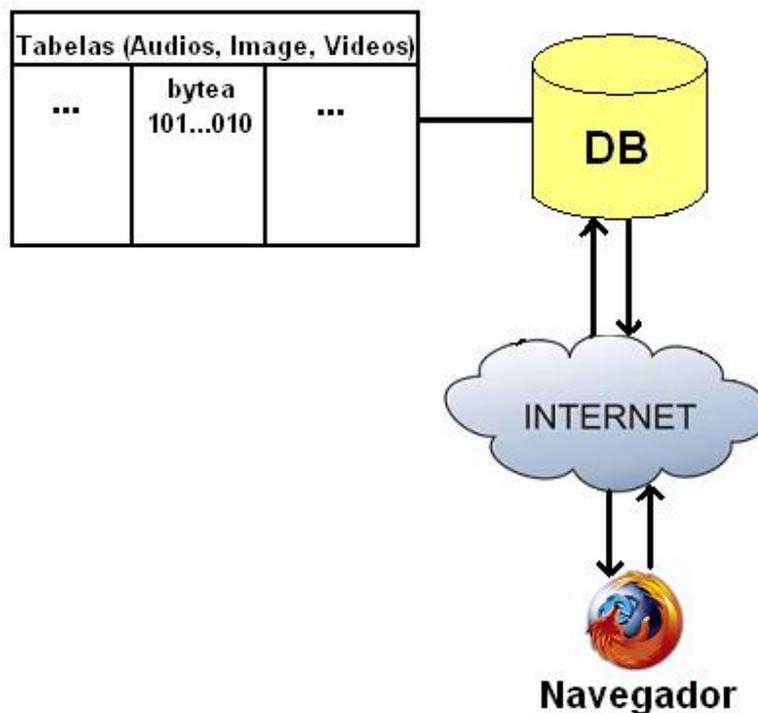


Figura 19: Arquitetura do Protótipo 3.

O Protótipo 3 envia para o navegador do usuário final a seqüência de bytes e algumas informações sobre o objeto para que o navegador possa interpretar essa seqüência e exibir, caso a seqüência de bytes seja uma imagem, ou simplesmente chamar o *Player default* capaz de tocar o áudio ou o vídeo recuperado.

Similarmente à abordagem do Protótipo 1, a abordagem do Protótipo 3 também provê as mesmas vantagens já citadas de se guardar um objeto multimídia no SGBD.

4.6.1 Estrutura do Banco de dados

Images	Audios	Videos
* id: int	* id: int	* id: int
description: text	description: text	description: text
name: text	name: text	name: text
author: text	author: text	author: text
tags: text	tags: text	tags: text
image: bytea	audio: bytea	video: bytea
thumbnail: bytea		thumbnail: bytea
license: char	license: char	license: char
width: int	length: int	width: int
height: int	dateinsertion: date	height: text
length: int	extension: text	length: int
dateinsertion: date	samplerate: int	dateinsertion: date
extension: text	bitrate: int	extension: text
	duration: int	bitrate: int
		framerate: float
		duration: int
		standard: text

Figura 20: Tabelas do Protótipo 3.

A Figura 17 nos mostra as tabelas que compõem a base de dados do Protótipo 3. Diferentemente dos protótipos já citados, os campos *image*, *thumbnail* (tabela *Images*), *audio*, *video* e *thumbnail* (tabela *Videos*) são do tipo “*bytea*”, uma vez que nestes campos são guardadas as seqüências binárias dos objetos multimídia.

4.7 Protótipo 4

Assim como as demais abordagens, esta abordagem também implementa o diagrama de classes, apresentado na Figura 14. Contudo, esta última abordagem surgiu como uma proposta nossa de se obter a vantagem da performance do armazenamento por referência com as vantagens já citadas de se guardar o objeto multimídia no SGBD.

Esta abordagem híbrida funciona da seguinte forma: o objeto multimídia é inserido no SGBD usando a abordagem *Large Object*, no entanto, a sua recuperação não ocorre similarmente ao Protótipo 1. Como visamos o menor

tempo de recuperação dos objetos, apenas invocamos a função “*lo_export*” caso o objeto já não esteja disponível no servidor WEB.

A primeira vez que um objeto é consultado, ele é exportado para o servidor WEB e ao contrário da abordagem do Protótipo 1, a pasta onde o objeto é exportado não tem o seu conteúdo apagado esporadicamente, já que essa pasta não será uma pasta temporária.

Nas próximas vezes que o objeto for consultado, o Protótipo 4 faz algumas verificações, que no próximo capítulo iremos perceber o impacto para a performance das consultas, para decidir se terá que exportar o arquivo para o servidor WEB. Dentre elas, o protótipo verifica a existência do objeto na pasta pré-determinada para checar se há necessidade da recuperação dos objetos. Caso o objeto não esteja lá, ele exporta. No caso contrário, ele verifica se o tamanho do objeto e a data de modificação do mesmo conferem com o tamanho e data de modificação que são guardados nas suas respectivas tabelas na hora da inserção do objeto multimídia no SGBD. Em caso de alguma diferença (alteração indevida), o SGBD exporta o objeto. Se nenhuma das duas situações ocorre, não é feita a recuperação do objeto.

Dessa forma, evitamos que caso o objeto já tenha sido exportado para o servidor WEB, o SGBD o exporte novamente. Esta solução utiliza, assim, a performance da abordagem por referência externa, garantindo a integridade do banco, visto que caso ocorra alguma exclusão do objeto multimídia do servidor WEB ou alguma modificação indevida no arquivo, o SGBD irá exportar novamente o objeto para o servidor WEB.

Com a Figura 21, podemos visualizar a arquitetura do Protótipo 4.

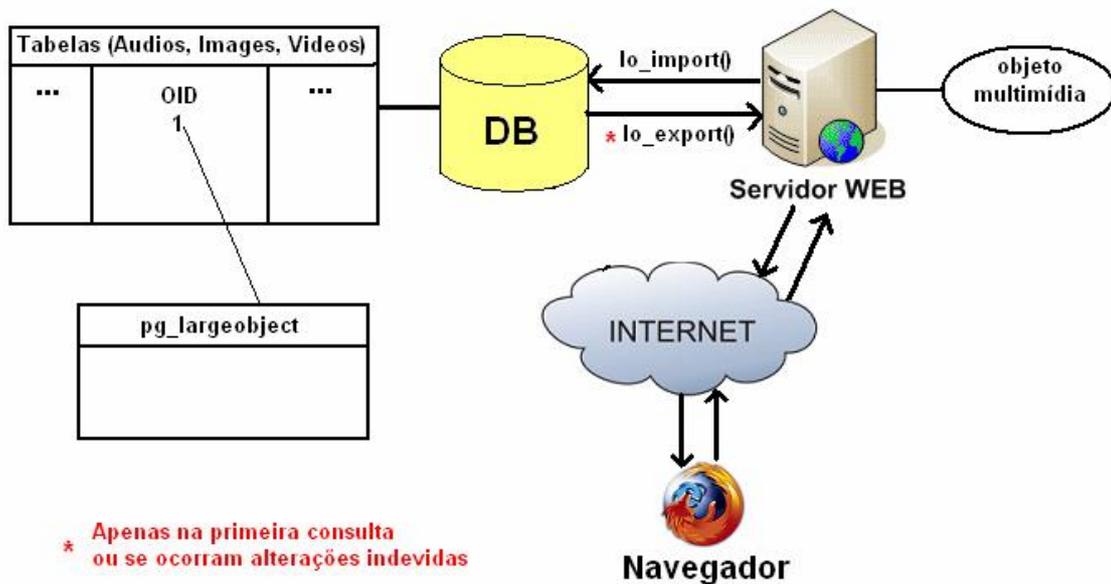


Figura 21: Arquitetura do Protótipo 4.

Esta abordagem é similar ao Protótipo 1 e ao Protótipo 2 no tocante à visualização do objeto por parte do usuário final.

4.7.1 Estrutura do Banco de dados

Images	Audios	Videos
* id: int	* id: int	* id: int
description: text	description: text	description: text
name: text	name: text	name: text
author: text	author: text	author: text
tags: text	tags: text	tags: text
image: lo	audio: lo	video: lo
thumbnail: lo	datemodification: long	thumbnail: lo
datemodification: long	license: char	datemodification: long
license: char	length: int	license: char
width: int	dateinsertion: date	width: int
height: int	extension: text	height: text
length: int	samplerate: int	length: int
dateinsertion: date	bitrate: int	dateinsertion: date
extension: text	duration: int	extension: text
		bitrate: int
		framerate: float
		duration: int
		standard: text

Figura 22: Tabelas do Protótipo 4.

A Figura 18 nos mostra as tabelas que compõem a base de dados do Protótipo 4. Podemos notar que a única diferença entre a base de dados do Protótipo 4 e a do Protótipo 1 é o aparecimento do campo *datemodification* (a este campo foi atribuído o tipo “*long*” já que a linguagem JAVA através da classe *Java.io.File* fornece um método de consulta da data de modificação de um arquivo e retorna esse valor com o tipo “*long*”). Este campo aliado ao campo *length* é de fundamental importância para o sucesso da abordagem proposta pelo Protótipo 4.

4.8 Considerações Finais

Como foi mostrado durante o capítulo, o conjunto de tabelas que compõem a base de dados de cada protótipo poderia apresentar uma tabela a mais, a qual teria todos os campos comuns aos três tipos de objetos multimídia estudados, enquanto que as tabelas *Images*, *Audios* e *Videos* conteriam apenas as informações particulares a cada tipo de objeto. Decidimos a priori a não criação dessa tabela genérica para evitar o custo com junções quando fôssemos recuperar os objetos. Contudo, nada impede, que futuramente, na versão final do sistema, se comprovadamente estas junções não impliquem em uma perda considerável de performance que apliquemos essa abordagem de normalizar a base de dados.

Enquanto neste capítulo apresentamos as arquiteturas de quatro protótipos com relação à forma de armazenamento / recuperação de dados multimídia em SGBD, no próximo capítulo iremos criar e aplicar um plano de testes em cada um e discutir os resultados obtidos.

5. Análise dos testes e resultados obtidos

Neste capítulo, definimos e aplicamos um plano de testes sobre os quatro protótipos obtidos com a implementação das quatro abordagens de manipulação de objetos multimídia estudadas. Na seqüência, apresentaremos uma análise dos resultados obtidos.

5.1 Plano de Testes

Utilizaremos a abordagem de testes denominada caixa preta, abordagem onde os testes se concentram apenas nas entradas e nas saídas geradas pelo componente avaliado [MOLINARI, 2006]. Como já foi dito anteriormente, utilizamos o JMeter [JMETER, 2007] para realizar um teste de carga e performance com os protótipos.

Outro ponto importante é: todos os testes foram realizados de forma “local”, isto significa que a mesma máquina que foi realizada a instalação do servidor de banco de dados e servidor WEB foi utilizada como máquina cliente das funcionalidades dos protótipos. A seguir, apresentaremos as informações sobre o servidor que foi utilizado para realizar os testes, bem como detalharemos as etapas dos testes.

5.1.1 Descrição do Servidor

O computador utilizado como servidor tem as seguintes características:

- Modelo da máquina - TravelMate 2434WLMi;
- Processador - Intel Celeron M Processor 380, 1.6 GHz, 400 MHz FSB, 1MB L2 Cache;
- Quantidade de memória RAM - 1GBytes;
- Capacidade do Disco Rígido - 60GBytes;

- Interface de Rede - Fast Ethernet 10/100 M e 802.11b/g wireless LAN;
- Sistema Operacional - Windows XP; e
- Processos ativos no momento do teste - Servidor de Banco de Dados, Servidor WEB, JMeter e processos internos do Sistema Operacional.

5.1.2 Etapas dos Testes

A priori tivemos a instalação do servidor de banco de dados, servidor WEB e *Framework* JMeter de forma *default*.

No segundo momento, estabelecemos doze cenários listados no Quadro 8 em que os quatro protótipos, descritos no capítulo 4, foram submetidos.

Quadro 8: Cenários de Testes.

Tipo do Objeto	Operação	Quantidade de usuários concorrentes	Tamanho do objeto	Padrão de Compressão
Imagem	Inserção	10	100 Kb	JPEG
Imagem	Consulta	10	100 Kb	JPEG
Imagem	Inserção	100	100 Kb	JPEG
Imagem	Consulta	100	100 Kb	JPEG
Áudio	Inserção	10	1 Mb	MP3
Áudio	Consulta	10	1 Mb	MP3
Áudio	Inserção	100	1 Mb	MP3
Áudio	Consulta	100	1 Mb	MP3
Vídeo	Inserção	10	1 Mb	MPEG-4
Vídeo	Consulta	10	1 Mb	MPEG-4
Vídeo	Inserção	100	1 Mb	MPEG-4
Vídeo	Consulta	100	1 Mb	MPEG-4

Cada cenário contido no Quadro 8 tem um tempo de um minuto para que todos os usuários completem a operação escolhida, além do que para obtermos mais dados cada cenário é executado por 10 iterações. Assim, em um cenário

onde tenhamos 10 usuários concorrentes obteremos 100 amostras de tempo (10 usuários X 10 iterações) e em outro cenário que tenhamos 100 usuários obteremos 1000 amostras de tempo (100 usuários X 10 iterações).

Os cenários descritos no Quadro 8 foram estabelecidos com o intuito de observarmos os comportamentos dos protótipos. Estabelecemos o tamanho da entrada em 100 Kb para uma imagem de teste e 1 Mb para áudios e vídeos de testes, visto que como os rodamos os testes em um servidor com pouco poder de processamento e espaço de disco, para termos noção do tempo que seria gasto em um ambiente real, escolhemos então tamanhos de arquivos em média 10 vezes menores que o tamanho dos arquivos que tem suporte em sistemas virtuais de ensino para imagens e áudios 100 vezes menores para vídeo.

E quanto à quantidade de usuários escolhida para os cenários, escolhemos 10 usuários simultâneos realizando uma mesma operação por ser um valor próximo do encontrado em sistemas virtuais de ensino, como também escolhemos criar cenários com 100 usuários para observamos o comportamento de cada protótipo em condições de estresse.

Todos os usuários que testarão um cenário, realizarão as operações de inserção e consulta utilizando os mesmos arquivos de imagem, áudio e vídeo. Esta observação é de fundamental importância para entendermos os resultados obtidos pela abordagem híbrida implementada pelo Protótipo 4. Na seção seguinte, apresentaremos os resultados dos testes aplicados nos protótipos utilizando os cenários descritos.

5.2 Resultados

Como já foi dito, nesta seção mostraremos os resultados obtidos com a aplicação dos cenários, descritos no Quadro 8, nos quatro protótipos descritos no capítulo 4. Para um melhor entendimento, decidimos agrupar os resultados por objetos multimídia: imagens, áudios e vídeos.

5.2.1 Testes nos Cenários Utilizando Arquivos de Imagens

Nesta Seção mostraremos os gráficos obtidos pelas séries temporais, criadas com os valores de tempo (em milissegundos) de resposta que cada operação de imagem testada demorou em um dado cenário e discutiremos os resultados obtidos.

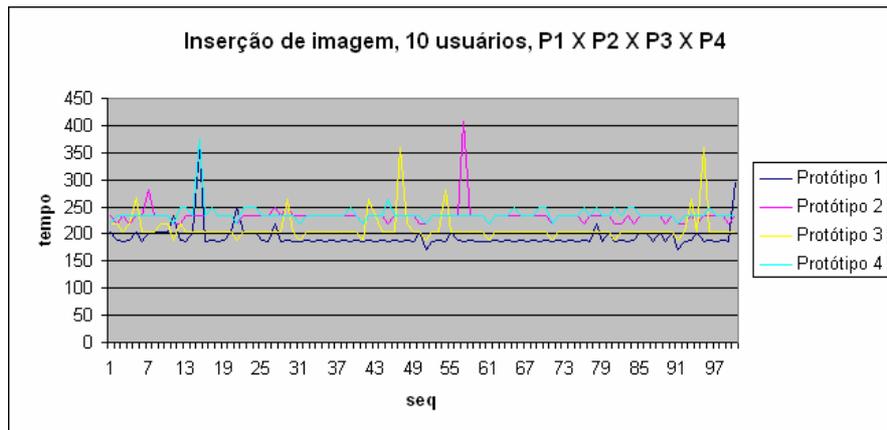


Figura 23: Inserção de Imagem (10 usuários).

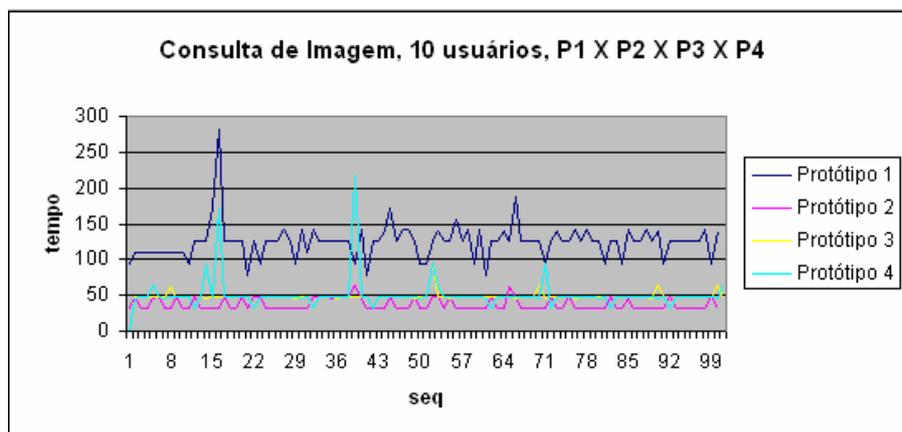


Figura 24: Consulta de Imagem (10 usuários).

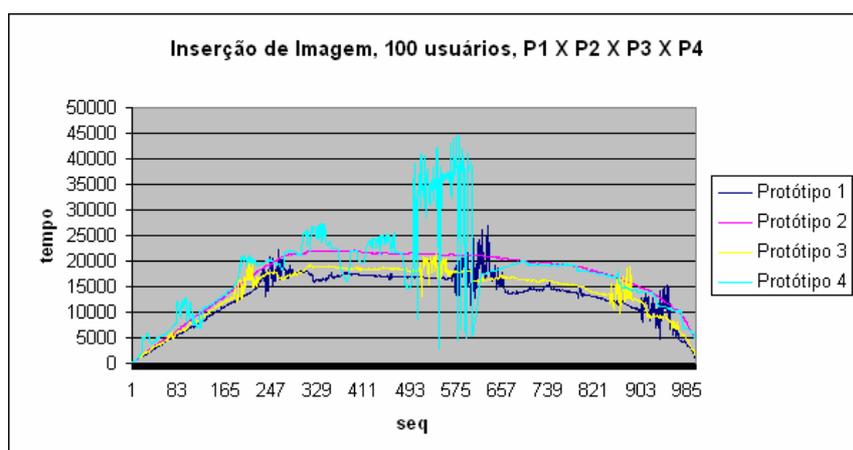


Figura 25: Inserção de Imagem (100 usuários).

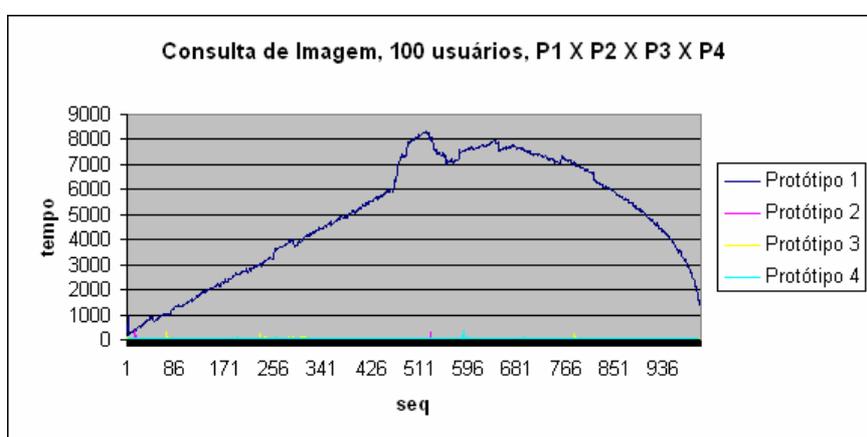


Figura 26: Consulta de Imagem (100 usuários).

Com base nos resultados apresentados com os gráficos (Figura 19, Figura 20, Figura 21 e Figura 22) montamos o Quadro 9, o qual apresenta uma comparação da média de tempo que cada protótipo levou para completar cada operação em um dado cenário de teste de imagem.

Quadro 9: Comparação do Desempenho do Protótipos.

Operação	Protótipo 1	Protótipo 2	Protótipo 3	Protótipo 4
Inserção 10 Usuários	195 ms	234 ms	210 ms	237 ms
Consulta 10 Usuários	124 ms	37 ms	47 ms	50 ms
Inserção 100 Usuários	13232 ms	16855 ms	14139 ms	18088 ms
Consulta 100 Usuários	4922 ms	46 ms	47 ms	46 ms

Com base nos resultados apresentados no Quadro 9, segue a análise do comportamento de cada protótipo.

O Protótipo 1 apresentou os melhores tempos para os dois cenários de inserção. Contudo, nos cenários de consulta apresentou tempos consideravelmente piores que os demais protótipos. O Protótipo 2 obteve tempos de inserção próximos aos dos protótipos 3 e 4, e como já era esperado, apresentou ótimos tempos para a recuperação das imagens. O Protótipo 3 apresentou os segundos melhores tempos de inserção e de consultas. Por fim, como também prevíamos, o Protótipo 4 apresentou tempos de inserções próximos, porém piores, que o Protótipo 1, por apresentar uma arquitetura de inserção semelhante a do Protótipo 1, adicionada de mais um campo: *datemodification*, e também como esperávamos o Protótipo 4 obteve tempos de consultas próximos ao Protótipo 2.

5.2.2 Testes nos Cenários Utilizando Arquivos de Áudios

Nesta Seção, mostraremos os gráficos obtidos pelas séries temporais, criadas com os valores de tempo (em milisegundos) de resposta que cada operação de áudio testada demorou em um dado cenário e discutiremos os resultados obtidos.

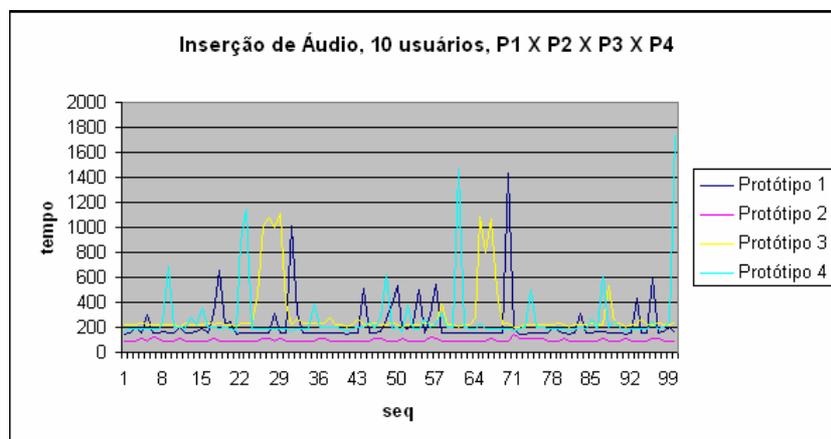


Figura 27: Inserção de Áudio (10 usuários).

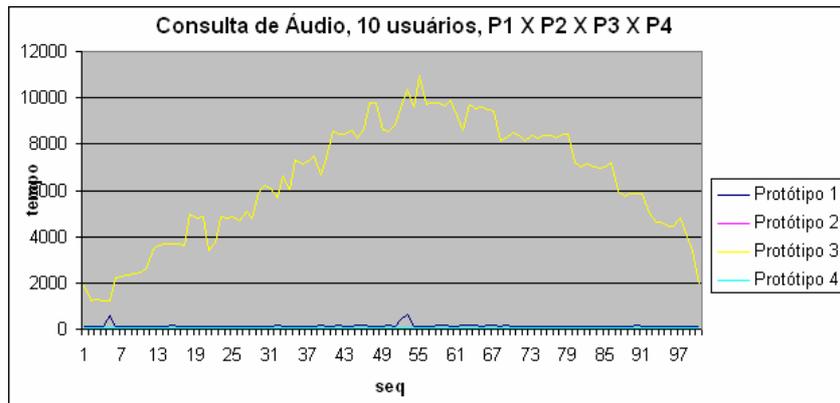


Figura 28: Consulta de Áudio (10 usuários).

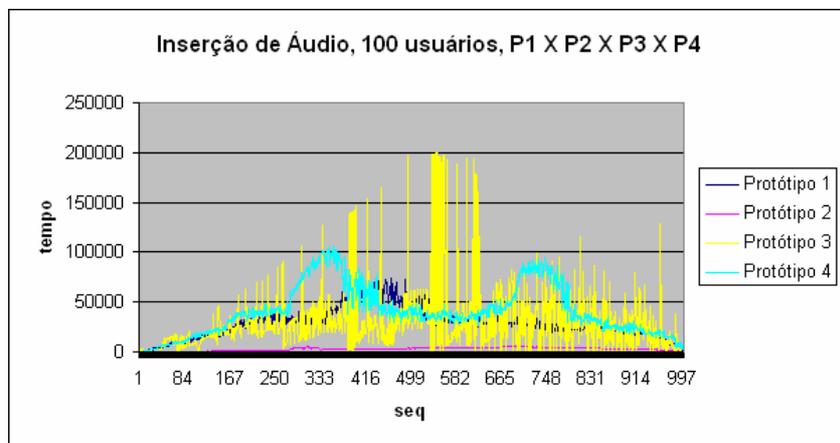


Figura 29: Inserção de Áudio (100 usuários).

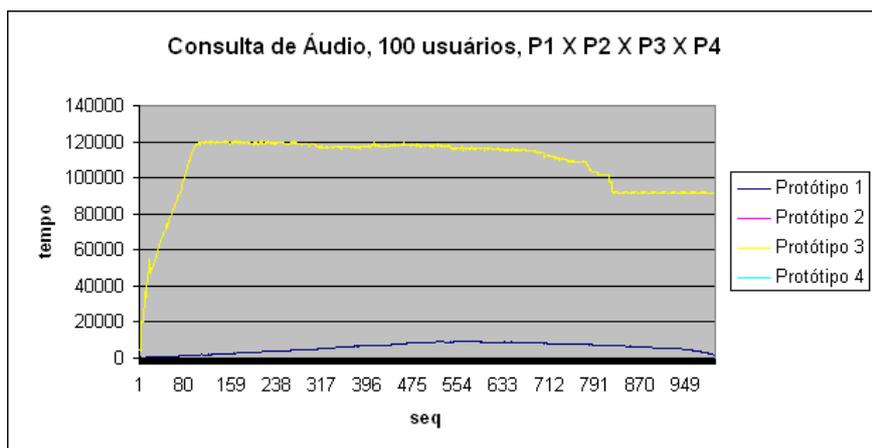


Figura 30: Consulta de Áudio (100 usuários).

Com base nos resultados apresentados com os gráficos (Figura 23, Figura 24, Figura 25 e Figura 26) montamos o Quadro 10, que apresenta uma

comparação da média de tempo que cada protótipo levou para completar cada operação em um dado cenário de teste de áudio.

Quadro 10: Comparação do Desempenho do Protótipos.

Operação	Protótipo 1	Protótipo 2	Protótipo 3	Protótipo 4
Inserção 10 Usuários	223 ms	99 ms	291 ms	266 ms
Consulta 10 Usuários	145 ms	37 ms	6410 ms	45,5 ms
Inserção 100 Usuários	28645 ms	2789 ms	30018 ms	40912 ms
Consulta 100 Usuários	5791 ms	40 ms	107372 ms	49 ms

Com base nos resultados apresentados no Quadro 10, analisaremos o comportamento de cada protótipo.

O Protótipo 1 apresentou uma performance aceitável com um ambiente de 10 usuários realizando uma dada operação em um minuto. Contudo, quando foi submetido a um ambiente estressante de 100 usuários em um minuto teve uma degradação de performance bastante considerável em relação ao Protótipo 2, notoriamente o protótipo de melhor desempenho.

O Protótipo 3 obteve uma drástica perda de performance com o aumento do tamanho do objeto multimídia e apresentou os piores resultados dentre os protótipos. Para esta perda, apontamos como principal causa o aumento do tamanho do arquivo manipulado. Observamos que a partir de arquivos com 500 Kb o PostgreSQL demanda muitos recursos (processamento e memória) do servidor do banco de dados.

O Protótipo 4, conforme o esperado, obteve resultados nas operações de inserções similares, porém piores, que o Protótipo 1, pelos mesmos motivos que citamos na análise dos cenários de imagens (Seção 5.2.1). Os resultados apresentados pelo Protótipo 4 nos cenários de consulta, como também prevíamos, foram próximos aos resultados obtidos pelo Protótipo 2.

5.2.3 Testes nos Cenários Utilizando Arquivos de Vídeos

Nesta Seção mostraremos os gráficos obtidos pelas séries temporais, criadas com os valores de tempo (em milissegundos) de resposta que cada operação de vídeo testada demorou em um dado cenário e discutiremos os resultados obtidos.

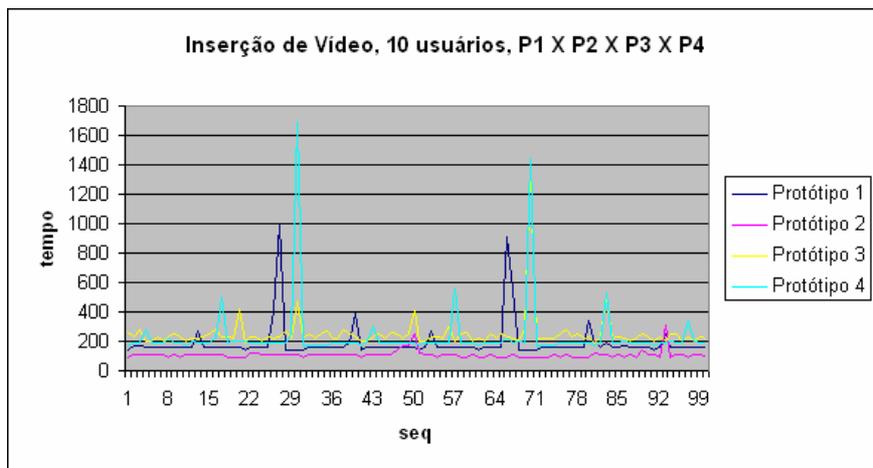


Figura 31: Inserção de Vídeo (10 usuários).

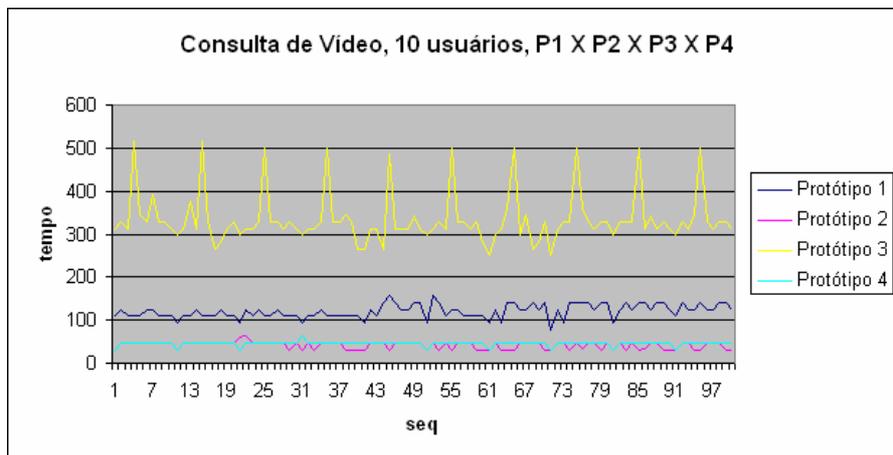


Figura 32: Consulta de Vídeo (10 usuários).

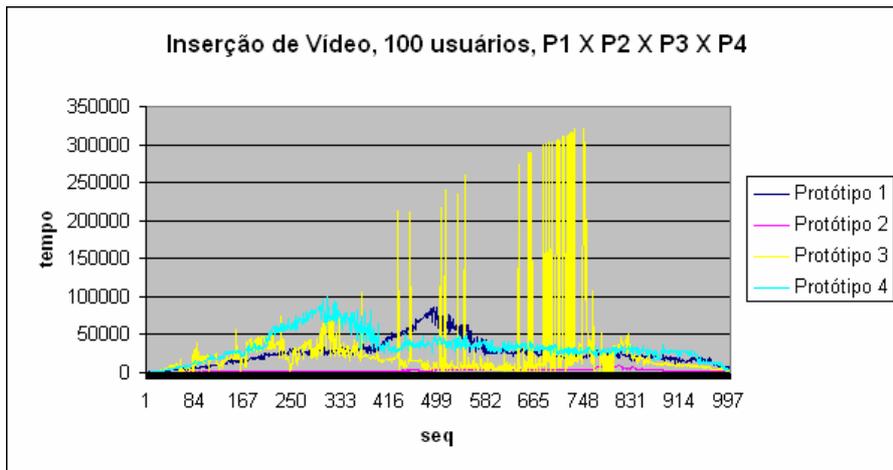


Figura 33: Inserção de Vídeo (100 usuários).

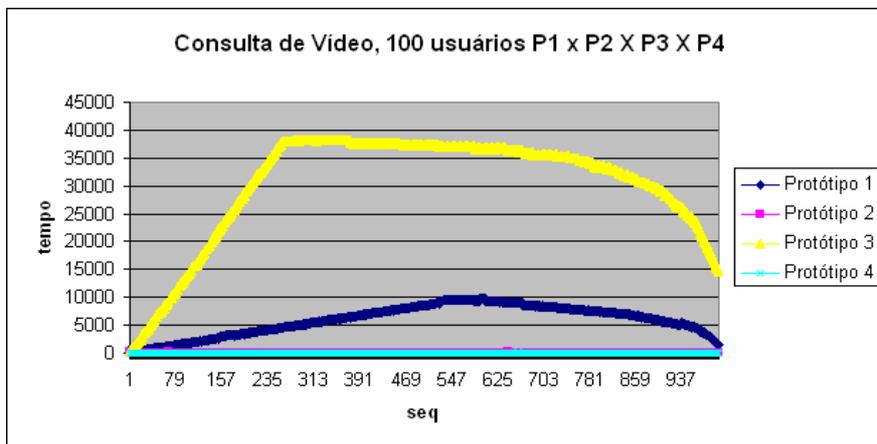


Figura 34: Consulta de Vídeo (100 usuários).

Com base nos resultados apresentados com os gráficos (Figura 27, Figura 28, Figura 29 e Figura 30) montamos o Quadro 11, que apresenta uma comparação da média de tempo que cada protótipo levou para completar cada operação em um dado cenário de teste de vídeo.

Quadro 11: Comparação do Desempenho do Protótipos.

Operação	Protótipo 1	Protótipo 2	Protótipo 3	Protótipo 4
Inserção 10 Usuários	185 ms	110 ms	250 ms	229 ms
Consulta 10 Usuários	121 ms	42 ms	334 ms	46 ms
Inserção 100 Usuários	26533 ms	2782 ms	27452 ms	34143 ms
Consulta 100 Usuários	5936 ms	38 ms	30322 ms	46 ms

Com base nos resultados apresentados no Quadro 11, analisaremos o comportamento de cada protótipo.

O Protótipo 1 novamente apresentou uma performance aceitável em uma ambiente não estressado. Contudo, em um ambiente de estresse teve uma grande degradação de performance em relação aos resultados em relação ao Protótipo 2, que como era esperado, manteve um ótimo desempenho independentemente do tamanho do objeto manipulado.

O Protótipo 3, assim como nos cenários de áudio, apresentou os piores resultados dentre os protótipos. E por fim, o Protótipo 4, novamente conforme estimado, apresentou resultados nas etapas de inserção próximos ao Protótipo 1, porém maiores, como também obteve resultados nos cenários de consultas próximos ao Protótipo 2.

5.3 Considerações Finais

Durante este capítulo, criamos um plano de testes com o intuito de entendermos os comportamentos dos protótipos em alguns ambientes controlados. Pudemos perceber com estes testes preliminares, vantagens e desvantagens de cada protótipo. No Quadro 12 apresentamos uma relação entre as virtudes, as limitações e a situação mais indicada para o uso de cada protótipo.

Quadro 12: Comparação Final dos Protótipos.

Protótipo	Vantagens	Desvantagens	Cenário de Uso
Protótipo 1	Integridade, Restrição de Acesso e Isolamento dos dados	Performance e obrigação da exportação do objeto recuperado para um servidor WEB	Ambientes com poucos usuários simultâneos e que a performance não é o primordial
Protótipo 2	Alta Performance	Falta de controle do SGBD sobre os objetos multimídia e possíveis inconsistências causadas por alterações indevidas na pasta que armazena os objetos.	Ambientes com vários usuários simultâneos e que a performance é o único fator relevante
Protótipo 3	Integridade, Restrição de Acesso, Isolamento dos dados e Envio direto do objeto multimídia para a aplicação cliente sem necessidade de exportação para um servidor WEB.	Baixa Performance	Ambientes com poucos usuários simultâneos e que a performance não é o primordial e sistemas que necessitem enviar objetos multimídia diretamente para as aplicações dos usuários
Protótipo 4	Integridade, Controle do SGBD sobre os objetos multimídia e Alta Performance nas Consultas	Baixa Performance nas Inserções	Ambientes com poucos usuários simultâneos para realizar inserções e ambientes com vários usuários simultâneos para realizar consultas e com o foco na performance

Com base no Quadro 12, podemos inferir que os protótipos 1 e 3 apresentaram resultados inviáveis para o contexto de sistemas virtuais de ensino, visto que à medida que foi aumentado o tamanho do objeto e quantidade de usuários simultâneos estas duas abordagens tiveram graves degradações de desempenho em relação ao melhor protótipo.

O Protótipo 2, como já se era esperado, apresentou a melhor performance dentre todos os protótipos, contudo com esta abordagem o banco de dados não tem nenhum controle sobre os objetos multimídia e alterações indevidas nas pastas que armazenam estes objetos podem causar sérios problemas de consistência no banco de dados.

O Protótipo 4, também apresentou os resultados conforme esperávamos. Ele obteve tempos nas operações de inserção similares ao Protótipo 1 e tempos nas consultas próximos ao Protótipo 2. O referido Protótipo 4 utiliza uma abordagem híbrida descrita na seção 7 do Capítulo 4.

Mesmo o Protótipo 4 apresentando resultados na inserção de objetos normalmente distantes dos tempos do protótipo de melhor desempenho, ele foi escolhido como a melhor das quatro abordagens, visto que o mesmo apresentou, assim como o Protótipo 2, ótimos resultados nas consultas independentemente dos cenários escolhidos. É sabido que com servidores mais potentes e com ajustes nas configurações internas do PostgreSQL os tempos apresentados pelas inserções de objetos multimídia podem ser reduzidos. Outro fator que ratifica a escolha dessa abordagem como a mais indicada das quatro abordagens analisadas é o fato da operação de inserção ser executada em um grau muito menor de vezes que a operação de consulta em sistemas virtuais de ensino.

6. Conclusões

Ao longo da pesquisa, como planejamos, alcançamos todos os objetivos traçados. Conseguimos criar um material de estudo sobre os aspectos conceituais dos objetos multimídias escolhidos; realizamos uma pesquisa sobre serviços de manipulação de áudios, imagens e vídeos, de extrema importância para o sucesso do objetivo principal do trabalho; definimos, especificamos e implementamos protótipos baseados em diferentes abordagens de armazenamento e manipulação de objetos multimídia na WEB, de acordo com os serviços estudados; e, por fim, executamos um plano de teste sobre os protótipos que nos ajudaram a entender o funcionamento do PostgreSQL com objetos multimídia e indicar uma melhor abordagem de manipulação destes objetos em sistemas virtuais de ensino.

6.1 Contribuições

Dentre as várias contribuições obtidas com essa pesquisa podemos citar:

- Formação de um material de estudo sobre os aspectos conceituais de imagens, áudios e vídeos digitais;
- Criação de um material de estudo das funcionalidades dos principais serviços de manipulação de imagens, áudio e vídeos presentes na WEB;
- Formação de um material de estudo sobre a manipulação de dados multimídia no PostgreSQL;
- Mesmo não apresentando bons resultados com o PostgreSQL a construção de uma arquitetura (Protótipo 3) que possibilita que um objeto binário oriundo de um banco de dados qualquer, possa ser enviado diretamente para o navegador do usuário final, sem precisar de uma etapa intermediária, ou seja, cópia do arquivo do banco para uma pasta temporária em um servidor WEB é bastante interessante,

já que ao reduzir esta etapa intermediária deveria ser notório o ganho de performance, no entanto o PostgreSQL não reagiu bem com objetos multimídia maiores que 500 Kb inviabilizando assim que o protótipo 3 tivesse melhores resultados;

- Construção de uma arquitetura (todos Protótipos) para a visualização de vídeos e reprodução de áudios, que não precisa da etapa de conversão do vídeo ou áudio para o padrão de compressão FLV, comum em diversos serviços estudados no capítulo 3; e
- Por fim, propomos uma arquitetura híbrida (Protótipo 4) que de acordo com os nossos testes preliminares se mostrou uma opção interessante para adoção em sistemas virtuais de ensino.

6.2 Dificuldades Encontradas

Durante esta pesquisa conseguimos, utilizando uma biblioteca chamada ImageInfo [IMAGEINFO, 2007], extrair no momento da inserção de uma imagem as suas características. Tentamos implementar essa extração automática para áudio e vídeo utilizando as bibliotecas JMF [JMF, 2007] e FFMPEG [FFMPEG, 2007], contudo não tivemos sucesso. Com a primeira biblioteca ainda conseguimos extrair o padrão de compressão e a duração de um áudio ou vídeo; com a segunda nos deparamos com problemas de compatibilidade com o sistema operacional utilizado no ambiente de testes. Com isso, resolvemos, indicar esta tarefa para a seção de *Trabalhos Futuros* e nos dedicamos a outras questões da pesquisa.

Outra dificuldade a ser listada, foi o fato de não conseguirmos em tempo hábil rodar o plano de teste do capítulo 5 em um servidor mais potente. No entanto, indicamos para a seção de *Trabalhos Futuros* criar outros planos de testes e executá-los em servidores mais potentes, similares aos que normalmente sistemas virtuais de ensino são hospedados.

6.3 Trabalhos Futuros

Para esta seção, como já foi dito, indicamos a extração automática de características de áudio e vídeo, por ser uma tarefa importante tendo em vista a possibilidade de fornecer consultas sobre as características de um objeto multimídia. Na etapa de inserção, como o sistema se responsabilizaria por extrair os dados do objeto inserido, o usuário apenas precisaria passar as informações textuais.

No nosso trabalho, as buscas são realizadas nas informações que descrevem o objeto. Sabemos que buscas por conteúdos dos objetos multimídia agregariam valor a sistemas virtuais de ensino. Por isso, também indicamos esta tarefa como um trabalho futuro.

Também indicamos como trabalhos a serem realizados no futuro, refinamentos e estudos de usabilidade com as interfaces dos protótipos. Também, indicamos estender o sistema para tratar outros tipos objetos multimídia, tais como objetos 3D.

E como principal trabalho futuro indicamos a realização de testes com outros inúmeros cenários e com outras abordagens híbridas possíveis de gerar os meios para obter a melhor performance do PostgreSQL, sem perder as vantagens de se guardar o objeto multimídia no banco de dados.

Referências

- [3GPP, 2007] 3rd Generation Partnership Project. Disponível em: <http://www.3gpp.org/>. Acesso em: 01/05/2007.
- [4SHARED, 2007] 4Shared, 2007. Disponível em: <http://www.4shared.com>. Acesso em: 01/05/2007.
- [AMPAS, 2007] Academy of motion pictures arts and sciences – AMPAS, 2007. Disponível em: <http://www.oscars.org/>. Acesso em: 15/05/2007.
- [ADOBE, 2007] Adobe, 2007. Disponível em: <http://www.adobe.com>. Acesso em: 02/05/2007.
- [AMADEUS, 2006] AMADEUS_MM. Plataforma de Educação à distância. Disponível em: <http://amadeus.cin.ufpe.br>. Acesso em: 01/05/2007.
- [APPLE, 2006] APPLE, 2006. QuickTime and MPEG-4: Now Featuring H.264. Disponível em: http://images.apple.com/quicktime/pdf/H264_Technology_Brief.pdf. Acesso em: 02/06/2007.
- [APPLE, 2007] APPLE, 2007. QuickTime Technologies. Disponível em: <http://www.apple.com/quicktime/technologies/>. Acesso em: 02/05/2007.
- [AVARO et al., 2002] AVARO,O.; KOENEN,R.; PEREIRA, F. MPEG-4 Overview. In: PEREIRA, F.;EBRAHIMI, T. THE MPEG-4 Book. Upper Saddle River: Prentice Hall PTR,2002.
- [BATTISTA et al., 1999] BATTISTA, S.; CASALINO, F.; LANDE, C., 1999. MPEG-4: A Multimedia Standard for the Third Millennium - Part 1. IEEE MultiMedia.
- [BDA, 2007] Blu-ray Disc Association, 2007. Disponível em: <http://www.blu-raydisc.com/>. Acesso em: 01/05/2007.
- [BESSER & TRANT, 1995] Besser, H. and Trant, J., 1995. Introduction to imaging: Issues in constructing an image database. Getty Information Institute.

- [BROADCASTER, 2007] Broadcaster, 2007. Disponível em: <http://www.broadcaster.com>. Acesso em: 02/05/2007.
- [BRMODELO, 2007] brModelo, 2007. Disponível em: <http://chcandido.tripod.com/>. Acesso em: 10/07/2007.
- [CASTELLS, 2000] CASTELLS, Manuel. A Sociedade em Rede. 3.ed. São Paulo - SP: Paz e Terra, 2000.
- [CHIARIGLIONE, 1996] CHIARIGLIONE, L., 1996. Short MPEG-1 description. Disponível em: <http://www.chiariglione.org/mpeg/standards/mpeg-1/mpeg-1.htm>. Acesso em: 02/05/2007.
- [COMPUSERVE, 2007] CompuServe, 2007. Disponível em: <http://www.compuserve.com/>. Acesso em: 02/05/2007.
- [CORNELL, 2000] Cornell, 2000. Moving theory into practice: Digital imaging tutorial. Relatório Técnico, Cornell University Library.
- [DA SILVA, 2006] DA SILVA, R. C., 2006. Benchmark em banco de dados multimídia: Análise de desempenho em recuperação de objetos multimídia. Dissertação de Mestrado, Universidade Federal do Paraná. Disponível em: http://dspace.c3sl.ufpr.br/dspace/bitstream/1884/4683/1/disserta_ricardo.pdf. Acesso em: 10/06/2007.
- [DAB, 2007] Digital Audio Broadcasting – DAB, 2007. Disponível em: <http://www.worlddab.org/>. Acesso em: 02/05/2007.
- [DAILYMOTION, 2007] Dailymotion, 2007. Disponível em: <http://www.dailymotion.com/>. Acesso em: 02/05/2007.
- [DIVSHARE, 2007] DivShare, 2007. Disponível em: <http://www.divshare.com/>. Acesso em: 02/05/2007.
- [DVB, 2007] Digital Video Broadcasting – DVB, 2007. Disponível em: <http://www.dvb.org/>. Acesso em: 02/05/2007.
- [DVD, 2007] DVD Forum, 2007. Disponível em: <http://www.dvdforum.org/>. Acesso em: 02/05/2007.

- [ELKOUTBI et al., 1999] M. Elkoutbi, I. Khriess & R.K. Keller. Generating User Interface Prototypes from Scenarios, IEEE Int'l. Symp. On Requirements Eng., 1999.
- [FERNANDES, 2006] FERNANDES, J., 2006. TV digital interativa. Trabalho de conclusão de curso de pós-graduação, Escola Superior aberta do Brasil - ESAB, Vitória - ES. Disponível em: <http://www.esab.edu.br/site/biblioteca/monografias/TccEngSistJf.pdf>. Acesso em: 02/05/2007.
- [FFMPEG, 2007] FFmpeg, 2007. Disponível em: <http://ffmpeg.mplayerhq.hu/>. Acesso em: 18/06/2007.
- [FLICKR, 2007] Flickr, 2007. Disponível em: <http://www.flickr.com/>. Acesso em: 02/05/2007.
- [FRAUNHOFER, 2007] Instituto Fraunhofer, 2007. Disponível em: <http://www.iis.fraunhofer.de/>. Acesso em: 10/05/2007.
- [GAMMA et al. , 1994] GAMMA, Erich et al. Design patterns: elements of reusable object-oriented software. Reading: Addison-Wesley, 1994.
- [GATTASS, 2005] GATTASS, M., 2005. Imagem Digital. Disponível em: http://www.tecgraf.puc-rio.br/~mgattass/cg/pdf/03_ImagemDigital.pdf. Acesso em: 10/05/2007.
- [GIBSON, 2006] GIBSON, B., 2006. Sampling Rates from The AudioPro Home Recording Course. Disponível em: <http://digitalproducer.digitalmedianet.com/articles/viewarticle.jsp?id=7408>. Acesso em: 13/05/2007.
- [GOFISH, 2007] GoFish, 2007. Disponível em: <http://www.gofish.com/>. Acesso em: 02/05/2007.
- [GONZALEZ & WOODS, 1992] GONZALEZ, R. C. and WOODS, R. E., 1992. Digital Image Processing. Addison-Wesley Publishing Company.
- [GOOGLE VIDEO, 2007] Google Video, 2007. Disponível em: <http://www.googlevideo.com/>. Acesso em: 02/05/2007.

- [GOULART, 2003] GOULART, R., 2003. Personalização e adaptação de conteúdo baseadas em contexto para TV Interativa. Tese apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP. Disponível em: <http://www.mpeg.org.br/arquivos/rudinei.pdf>. Acesso em: 02/05/2007.
- [GRILL, 2006] GRILL, Bernhard. 2006. The MPEG-4 General Audio Coder. Disponível em: <http://sound.media.mit.edu/mpeg4/audio/general/aes17-43-mp4ga.pdf>. Acesso em: 02/05/2007.
- [H.263, 1998] H.263, 1998. ITU-T Recommendation H.263. Disponível em: <http://www.itu.int/rec/T-REC-H.263/recommendation.asp?lang=en&parent=T-REC-H.263-199802-S>. Acesso em: 02/05/2007.
- [IETF, 2007] Internet Engineering Task Force, 2007. Disponível em: <http://www.ietf.org/>. Acesso em: 02/05/2007.
- [IMAGEINFO, 2007] ImageInfo, 2007. Disponível em: <http://schmidt.devlib.org/image-info/>. Acesso em: 18/06/2007.
- [ISMA, 2007] Internet Streaming Media Alliance, 2007 Disponível em: <http://www.isma.tv/>. Acesso em: 02/05/2007.
- [ITU-T, 2007] International Telecommunication Union - ITU-T, 2007. Disponível em: <http://www.itu.int/ITU-T/>. Acesso em: 12/05/2007.
- [JAVA, 2007] Java Technology, 2007. Disponível em: <http://java.sun.com/>. Acesso em: 04/06/2007.
- [JMETER, 2006] JMeter, 2006. Disponível em: <http://www.apache.org/jakarta/jmeter>. Acesso em: 10/05/2007.
- [JMF, 2007] Java Media Player - JMF, 2007. Disponível em: <http://java.sun.com/products/java-media/jmf>. Acesso em: 10/05/2007.
- [JPEG, 2007] Joint Photographic Experts Group - JPEG, 2007. Disponível em: <http://www.jpeg.org>. Acesso em: 10/05/2007.

- [JVT, 2007] Joint Video Team – JVT, 2007. Disponível em: <http://www.itu.int/ITU-T/studygroups/com16/jvt/index.html>. Acesso em: 10/05/2007.
- [JUDE, 2007] Jude Community, 2007. Disponível em: <http://jude.change-vision.com/jude-web/index.html>. Acesso em: 10/06/2007.
- [JSP, 2007] Java Server Pages – JSP, 2007. Disponível em: <http://java.sun.com/products/jsp/>. Acesso em: 02/05/2007.
- [MEDIAMAX, 2007] MediaMax, 2007. Disponível em: <http://www.mediamax.com/>. Acesso em: 02/05/2007.
- [METACAFE, 2007] Metacafe, 2007. Disponível em: <http://www.metacafe.com/>. Acesso em: 02/05/2007.
- [MICROSOFT, 2007] Microsoft® Corporation, 2007. Disponível em: <http://www.microsoft.com/>. Acesso em: 02/05/2007.
- [MOLINARI, 2006] MOLINARI, Leonardo. Testes de Software – Produzindo sistemas melhores e mais confiáveis. Ed. Érica, 2006, 3ª edição;
- [MPEG, 2007] Moving Picture Experts Group – MPEG, 2007. Disponível em: <http://www.mpeg.org/>. Acesso em: 05/05/2007.
- [NUNES, 2005] Nunes, F. H. C., 2005. Desenvolvimento de Aplicações Hiperídia para Gerenciamento de Documentos Multimídia e Preservação de Acervos Digitais: Um Estudo de Caso no Cecor. Dissertação de Mestrado, Universidade Federal de Minas Gerais, Belo Horizonte. Disponível em: <http://www.npd.dcc.ufmg.br/orientacoes/mestrado/fhcunes/dissertacao.pdf>. Acesso em: 05/05/2007.
- [ON2, 2007] On2 Technologies, 2007. Disponível em: <http://www.on2.com/>. Acesso em: 03/05/2007.
- [PICASA, 2007] Picasa, 2007. Disponível em: <http://www.picasa.com/>. Acesso em: 02/05/2007.
- [POSTGRESQL, 2007] PostgreSQL, 2007. Disponível em: <http://www.postgresql.org/>. Acesso em: 10/06/2007.

- [REALNETWORKS, 2007] RealNetworks, 2007. Disponível em: <http://www.realnetworks.com/>. Acesso em: 04/05/2007.
- [RICARTE, 1997] Ricarte, I. L. M., 1997. Produção de Documentos para a World-WideWeb. UNICAMP, Campinas. Disponível em: <http://www.dca.fee.unicamp.br/courses/IA368F/Referencias/docwww.pdf>. Acesso em: 06/05/2007.
- [SMPTE, 2007] Society of Motion Picture and Television Engineers – SMPTE, 2007. Disponível em: <http://www.smpite.org/>. Acesso em: 07/05/2007.
- [SONY, 2007] Sony, 2007. Disponível em: <http://www.sony.com/>. Acesso em: 08/05/2007.
- [TERRA, 2007] Terra Fotolog, 2007. Disponível em: <http://fotolog.terra.com.br/>. Acesso em: 15/05/2007.
- [THOMSON, 2007] THOMSON SA, 2007. Disponível em: <http://www.thomson.net>. Acesso em: 11/05/2007.
- [TODESCO, 2000] TODESCO, G., 2000. Criação de Ambientes Virtuais Multiusuário através do Padrão MPEG-4. Programa de Pós-Graduação, Universidade Federal de São Carlos - SP. Disponível em: <http://www.mpeg.org.br/arquivos/todesco2000a.pdf>. Acesso em: 10/05/2007.
- [TOMCAT, 2007] Apache Tomcat, 2007. Disponível em: <http://tomcat.apache.org/>. Acesso em: 15/05/2007.
- [TUDOR, 1995] TUDOR, P.N., 1995. MPEG-2 Video Compression. Electronics & Communication Engineering Journal, December. Disponível em: http://www.bbc.co.uk/rd/pubs/papers/paper_14/paper_14.shtml. Acesso em: 12/05/2007.
- [UNISYS, 2007] Unisys, 2007. Disponível em: <http://www.unisys.com/>. Acesso em: 07/05/2007.
- [UOL, 2007] UOL Fotoblog, 2007. Disponível em: <http://fotoblog.uol.com.br/> Acesso em: 15/05/2007.

- [VIANELLO, 2006] VIANELLO, J. M., 2006. Métodos de Estimação de Movimento com Resolução em Subpixel no CODEC H264/AVC. Dissertação de mestrado, PUC-RIO. Disponível em: http://www.maxwell.lambda.ele.puc-rio.br/cgi-bin/PRG_0599.EXE/10168_3.PDF?NrOcoSis=32747&CdLinPr g=pt. Acesso em: 13/05/2007.
- [VMIX, 2007] vMix, 2007. Disponível em: <http://www.vmix.com/>. Acesso em: 10/05/2007.
- [WEBOPEDIA, 2007] Webopedia, 2007. Disponível em: <http://www.webopedia.com/>. Acesso em: 15/05/2007.
- [WIKIPEDIA, 2007] Wikipedia, 2007. Disponível em: <http://www.wikipedia.org/>. Acesso em: 10/05/2007.
- [W3C, 2007] World Wide Web Consortium – W3C, 2007. Disponível em: <http://www.w3.org/>. Acesso em: 04/05/2007.
- [WMA, 2007] Windows Media Audio, 2007. Disponível em: <http://www.microsoft.com/windows/windowsmedia>. Acesso em: 12/05/2007.
- [WMV, 2007] Windows Media Video, 2007. Disponível em: <http://www.microsoft.com/windows/windowsmedia>. Acesso em: 12/05/2007.
- [YOU TUBE, 2007] You Tube, 2007. Disponível em: <http://www.youtube.com/>. Acesso em: 15/05/2007.
- [XIPH, 2007] Xiph.org Foundation , 2007. Disponível em: <http://xiph.org/>. Acesso em: 15/05/2007.

Assinaturas

Fernando da Fonsêca de Souza
Orientador

Armando Soares Sousa
Co-orientador

Antonio Josivaldo do Nascimento Filho
Aluno