

UNIVERSIDADE DE CAXIAS DO SUL
Centro de Computação e Tecnologia da Informação
Curso de Bacharelado em Ciência da Computação

Elton Poloni da Cunha

**DEFINIÇÃO E IMPLEMENTAÇÃO DA CAMADA DE PERSISTÊNCIA
DE DADOS PARA A FERRAMENTA INDEXVIDEO .**

Caxias do Sul – RS.

2010

Elton Poloni da Cunha

**DEFINIÇÃO E IMPLEMENTAÇÃO DA CAMADA DE PERSISTÊNCIA
DE DADOS PARA A FERRAMENTA INDEXVIDEO.**

Trabalho de Conclusão de Curso
para obtenção do Grau de
Bacharel em Ciência da
Computação da Universidade de
Caxias do Sul.

**Prof. Daniel Luis Notari
Orientador**

**Prof.^a Maria de Fátima Webber do Prado Lima
co-orientadora**

**Caxias do Sul - RS
2010**

**“A Mente que se abre à
uma nova ideia jamais
voltará ao seu tamanho
original”.**
Albert Einstein.

AGRADECIMENTOS

Agradeço a todos que de uma forma direta ou indireta me auxiliaram, e tiveram a destreza de compreender e auxiliar nas tarefas desempenhadas para conclusão deste trabalho, e em toda a minha graduação, em especial a minha família, pois sem ela não seria o que sou hoje e não teria forças para enfrentar esta batalha.

“TKS PX Maior” Obrigado Deus, por ter me inspirado e me dado a sabedoria de pensar e raciocinar e força para trabalhar e lutar pelos meus ideais.

RESUMO

IndexVÍdeo é uma aplicação desenvolvida para fins didáticos que auxilia no aprendizado através de conteúdos multimídias. A mesma possui diretivas de busca de conteúdos e recuperação de trechos de vídeos selecionados. Prevê também a possibilidade de postar novos conteúdos multimídias e até mesmo acrescentar informações nos conteúdos já existentes. Os conteúdos multimídia da aplicação seguem o padrão da tecnologia MPEG-7. Para isso se desenvolveu uma estrutura cliente-servidor para a aplicação indexVÍdeo. Atualmente suas informações são armazenadas em uma estrutura de dados baseadas na tecnologia XML. Optou-se pela utilização do XML devido à compatibilidade entre as tecnologias XML e MPEG-7, o qual necessita de um arquivo XML correspondente ao vídeo para conseguir reproduzir conteúdos multimídias associado a anotações relativas ao vídeo. A tecnologia MPEG-7 é utilizada nesta aplicação para reprodução do conteúdo multimídia. Visando a melhoria no armazenamento dos dados e a confiabilidade dos usuários perante a aplicação, foi analisado e viabilizado a alteração da camada de persistência de dados da aplicação indexVÍdeo, a qual passará a armazenar os seus dados em um SGBD (sistema gerenciador de banco de dados) diferentemente da situação atual, que é armazenada em arquivos XML.

Palavras-Chave: XML, MPEG-7, SGBD, Persistência de Dados.

LISTA DE FIGURA

Figura 1 – Intreface com Usuário – Manipulação de Vídeos	14
Figura 2 – Funcionalidades da aplicação indexVideo	15
Figura 3 – Diagrama esquemático da arquitetura cliente-servidor.....	16
Figura 4 – Interface Gráfica – Aplicação Servidor.....	17
Figura 5 – Interface com Usuário – Consulta Anotações.....	18
Figura 6 – Comunicação entre pacotes da aplicação.....	20
Figura 7 – Estrutura de pacotes da aplicação.....	21
Figura 8 – Estrutura do XML que armazena os dados de um vídeo.....	23
Figura 9 – XML Schema 2001 do MPEG-7.....	26
Figura 10 – Esquema de SGBD.....	30
Figura 11 – Diagrama esquemático da arquitetura cliente-servidor proposto.....	34
Figura 12 – Fluxo de consulta proposto.....	35
Figura 13 – Requisitos de adequação da camada de persistência.....	36
Figura 14 – Modelo Relacional Proposto.....	37
Figura 15 – Fluxo da Aplicação indexVideo.....	41
Figura 16 – Modelo Relacional Implementado.....	43
Figura 17 – Pacotes da aplicação indexVideo.....	45
Figura 18 – Pacotes adicionados à aplicação indexVideo.....	45
Figura 19 – Classes do pacote “DAO”.....	46
Figura 20 – Classes do Pacote “SCHEMA”.....	47
Figura 21 – Diagrama de Classes – Pacote “DAO”.....	47
Figura 22 – Diagrama de Classes – Pacote “SCHEMA”.....	48
Figura 23 – Classes Alteradas – Pacote “IO”.....	50
Figura 24 – Configuração Aplicação Servidor.....	53
Figura 25 – Configuração Aplicação Cliente.....	54
Figura 26 – Teste Cenário A.....	56
Figura 27 – Teste Cenário B.....	56
Figura 28 – Seleção de Idioma.....	65
Figura 29 – Seleção do Caminho de Instalação.....	66
Figura 30 – Seleção do Menu a ser disponibilizado.....	66
Figura 31 – Start da Instalação e copia dos arquivos.....	67
Figura 32 – Finalização do processo de instalação.....	67
Figura 33 – Instalação Concluída.....	68
Figura 34 – Configuração do Cliente.....	69
Figura 35 – Configuração do Servidor.....	70

LISTA DE TABELAS

Tabela 01 – Limitadores SGBD Postgress.....	72
---	----

LISTA DE ABREVIATURAS E SIGLAS

Sigla	Descrição em Inglês	Descrição em Português
API	<i>Application Programming Interface</i>	Interface de Programação de Aplicativos
BD		Banco de Dados
DB	<i>Data Base</i>	Base de dados
DDL	<i>Description Definition Language</i>	Definição da Linguagem de Descrição
GML	<i>General Markup Language</i>	Linguagem de Marcação Genérica
MPEG-7	<i>Multimídia Content Description Interface</i>	Interface de descrição de conteúdo multimídia
SGBD		Sistema Gerenciador de Banco de Dados
XML	<i>Extended Markup Language</i>	Linguagem Padronizada de Marcação Genérica

SUMÁRIO

1	Introdução.....	11
1.1	Objetivos.....	12
1.2	Estrutura do Texto.....	12
2	IndexVideo.....	14
2.1	Principais Funcionalidades.....	15
2.2	Estrutura da Aplicação.....	16
2.3	Características da Implementação.....	19
2.4	Camada de persistência da aplicação indexVideo.....	22
2.4.1	XML – Vantagens e desvantagens na aplicação.....	24
2.4.2	MPEG-7.....	25
2.4.3	Padrão MPEG-7 versus XML Schema.....	26
2.5	Considerações Finais.....	27
3	Alterações Propostas para Camada de Armazenamento do indexVideo.....	28
3.1	Análise de uso de um SGBD.....	28
3.1.1	SGBD.....	28
3.1.2	Componentes do SGBD.....	30
3.1.3	Objetivos de SGBD.....	31
3.1.4	Vantagens e desvantagens do SGBD.....	32
3.2	Comparação entre XML e SGBD.....	33
3.3	Mudanças Propostas.....	34
3.3.1	Funcionalidades Novas.....	35
3.3.2	Camada de Persistência em SGBD.....	37
3.3.3	Mudanças no código fonte.....	38
3.3.4	Impactos na Aplicação.....	39
3.3.5	Vantagens e Desvantagens.....	39
3.4	Considerações Finais.....	40
4	Alterações Realizadas na Aplicação.....	41
4.1	Fluxo de Informação.....	41
4.2	Modelo Relacional Adotado.....	43
4.3	Pacotes da Aplicação.....	44
4.4	Camada de persistência.....	45
4.5	Camada IO.....	49
4.6	Alterações de Integração - JVLC.....	52
4.7	Configuração da Aplicação.....	52
4.7.1	Configuração da Aplicação Servidor.....	53
4.7.2	Configuração da Aplicação Cliente.....	54
4.8	Melhorias Executadas na aplicação.....	54
4.9	Considerações Finais.....	55
5	Testes da Aplicação.....	56
5.1	Acesso a Base de Dados.....	57
5.2	Geração de XML.....	57
5.3	Reprodução de Vídeo.....	58
5.4	Integração entre XML e SGBD.....	59
5.5	Consultas.....	59
5.6	Considerações Finais.....	59
6	Conclusão.....	61
7	Referências.....	63

Anexo A.....	65
Anexo B.....	71

1 INTRODUÇÃO

A aplicação indexVideo foi desenvolvida para fins didáticos, que por meio de sistema computacional, auxilia no aprendizado através de conteúdos multimídia, na qual prevê mecanismo de busca e recuperação das informações armazenadas. Permitindo ao usuário fazer *upload* de seus próprios conteúdos e adicionar anotações em trechos de vídeos previamente selecionados, assim acrescentando informações pertinentes a seleção feita pelo próprio usuário, aumentando a base de conhecimento da aplicação (CARBONERA, 2007).

Para a aplicação indexVideo possuir interatividade com o usuário foi utilizado recursos do padrão MPEG-7 integrado com XML para armazenamento das informações pertinentes a cada vídeo salvo na aplicação. Concebido em uma estrutura de cliente - servidor, fez-se necessário a utilização de *Sockets*¹ para as aplicações comunicarem entre si e trocar mensagens para seu devido gerenciamento (CARBONERA, 2007).

O MPEG-7 padronizado pela MPEG é responsável pelo desenvolvimento de padrões de representações codificadas de áudio e vídeo digitais, a qual fornece várias ferramentas de mesmo padrão, facilitando na integração entre ferramentas; sendo que as descrições multimídia possuem uma forma de pesquisa muito similar ao mecanismo de busca na internet, permitindo ao usuário fazê-las de forma fácil, ágil e simplificada, assim retornando o conteúdo necessário para satisfazer a sua pesquisa (MPEG-7, 2009), (CHELLA, 2003). Devido a esta facilidade e padronização das estruturas, foi adotada na aplicação indexVideo padrão MPEG-7, a qual utiliza estruturas XML para armazenar os dados e as anotações de um vídeo (CARBONERA, 2007).

A cada vídeo salvo na aplicação existe um arquivo XML correspondente, contendo os dados gerais do vídeo e as suas anotações, onde cada anotação contém informações específicas de um trecho de vídeo selecionado. A cada unidade temporal de um trecho de vídeo dos trechos selecionados pelo usuário existe uma anotação, contendo dados de personagem, local da cena e etc (CARBONERA, 2007). Estas informações auxiliam na busca dos trechos dos vídeos relacionados com assuntos especificados pelo usuário.

Para auxiliar na educação a aplicação indexVideo utiliza o mecanismo das anotações, para facilitar a busca de conteúdos específicos em trechos dos vídeos armazenados, assim demonstrando ao aluno exatamente o que ele está procurando. Adotando a ideia de um sumário de livro onde se faz uma busca rápida através do mesmo, localizando o assunto

¹ <http://java.sun.com/j2se/1.4.2/docs/api/java/net/Socket.html>

desejado em menor tempo do que se tivesse que procurar no livro inteiro. Seguindo esta ideia implementou-se na aplicação um sumário digital, que abrevia a procura de um assunto especificado pelo aluno em um determinado trecho de vídeo, ao invés de ver o vídeo inteiro ele só irá ver o trecho de interesse (CARBONERA, 2007).

A aplicação indexVideo possui uma interface amigável com o usuário, permitindo buscas rápidas sobre os conteúdos desejados, devido a uma gama de filtros que podem ser aplicados a uma única consulta e também permite ao usuário fazer *uploads* de vídeos para compartilhamento com os demais usuários da aplicação. Este compartilhamento permite que os usuários da aplicação indexVideo façam anotações nos vídeos já existentes, assim expandindo a biblioteca de conteúdos da aplicação e detalhando os conteúdos existentes na aplicação indexVideo.

A aplicação indexVideo utiliza uma estrutura XML para persistir os dados armazenando-os em arquivo XML correspondente. Com este cenário existe uma perda de performance na manipulação destes arquivos, acarretando ao usuário final uma demora no tempo de resposta de suas solicitações à aplicação (CARBONERA, 2007).

1.1 OBJETIVOS

Com o objetivo de aprimorar a aplicação indexVideo e solucionar o problema da persistência de dados, sugere-se alterar a estrutura de armazenamento dos dados, passando de XML para uma estrutura pré definida de compreensão de um SGBD, Isto garantirá uma melhor segurança para aplicação trazendo uma série de vantagens que já estão desenvolvidas em qualquer SGBD que se adote.

1.2 ESTRUTURA DO TEXTO

No Capítulo dois, têm-se a explicação completa da aplicação indexVideo abrangendo as principais funcionalidades e estrutura da aplicação, as principais características da implementação, a estrutura e as funções da camada de persistência, vantagens e desvantagens da utilização da estrutura XML, e a integração do MPEG-7 com XML, permitindo conhecer melhor a aplicação para fazermos uma análise adequada dos requisitos.

O Capítulo três consiste na elaboração da análise da alteração da camada de persistência de dados, analisando a utilização do SGBD para aplicação com seus

componentes, comparação das estruturas XML e SGBD, validando as novas funcionalidades desenvolvidas, e as mudanças necessárias no código fonte, avaliação dos principais impactos que sofreria a aplicação, as vantagens e desvantagens desta alteração, visando aprimorar a aplicação.

O Capítulo quatro abrangerá todas as alterações feitas na aplicação, para viabilizar a entrada em produção com o sistema de SGBD, discriminando as funcionalidades adicionadas e alteradas, e as pendências que ficaram para serem desenvolvidas.

No Capítulo cinco, será tratada a elaboração e execução dos testes executados na aplicação indexVideo, afim de validar a consistência dos dados e integridade das operações.

O Capítulo seis, possui os resultados obtidos com este trabalho e as possíveis melhorias para a aplicação indexVideo.

2 INDEXVIDEO

A aplicação indexVideo é um sistema gerenciador de conteúdos multimídia. Tem por objetivo armazenar vídeos e anotações correspondentes aos mesmos, e permitir ao usuário adicionar novos conteúdos e as suas anotações. Também é permitido adicionar anotações em vídeos já postados na aplicação indexVideo por outros usuários. As anotações auxiliam na busca de conteúdos multimídia, além de conter dados sobre um trecho específico previamente selecionado, por exemplo, a unidade temporal que ocorre a anotação. Esta busca é executada por meio de expressões textuais, informadas pelo usuário. A aplicação processa esta expressão e retorna os conteúdos multimídia associadas, seguindo a especificação elaborada pelo usuário (CARBONERA, 2007).

Anotação é um conjunto de metadados que contém informações temporais e dados sobre o trecho do vídeo selecionado. Os metadados compreendem as seguintes informações: tema, autor, local, etc, e estas informações fazem inferência a um trecho de vídeo específico (CARBONERA, 2007).

A indexação e recuperação semântica das informações sobre o vídeo é uma tecnologia desenvolvida para viabilizar a utilização de suporte multimídia em atividades de aprendizagem em sala de aula. A ideia principal é a navegação entre as anotações e recuperação para a reprodução do trecho do vídeo em questão, como se navega em um sumário de um livro, porém, em um meio eletrônico (MPEG-7, 2009).

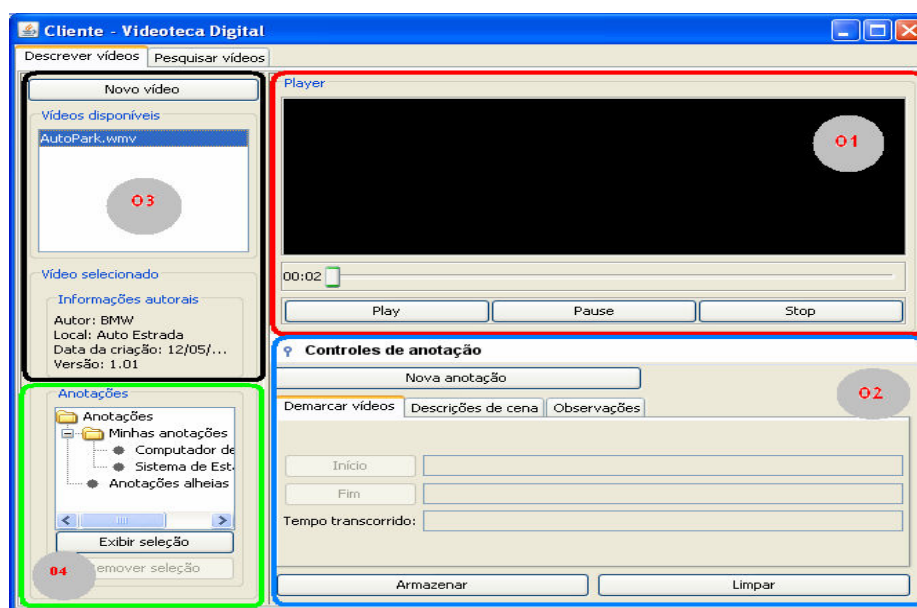


Figura 1 – Interface com Usuário – Manipulação de Vídeos

A Figura 1 mostra a interface gráfica da aplicação indexVideo. A mesma foi dividida em quatro áreas, sendo que a área 01 corresponde a um *player* multimídia que possui as funcionalidades básicas tradicionais. Já na área 02 encontra-se a anotação do vídeo selecionado, que compreende o mecanismo de delimitação temporal e um formulário para redigir os metadados relacionados ao trecho em questão. A área 03 possui os mecanismos de *upload* de vídeos para aplicação indexVideo e, também possibilita ao usuário selecionar os vídeos já salvos na aplicação. Na área 04 pode-se visualizar todas as anotações feitas sobre o vídeo selecionado (CARBONERA, 2007).

2.1 PRINCIPAIS FUNCIONALIDADES

A aplicação indexVideo proporciona ao usuário um ensino de qualidade através da utilização de conteúdos multimídia. A Figura 2 corresponde às funcionalidades desenvolvidas na aplicação atualmente estruturada.

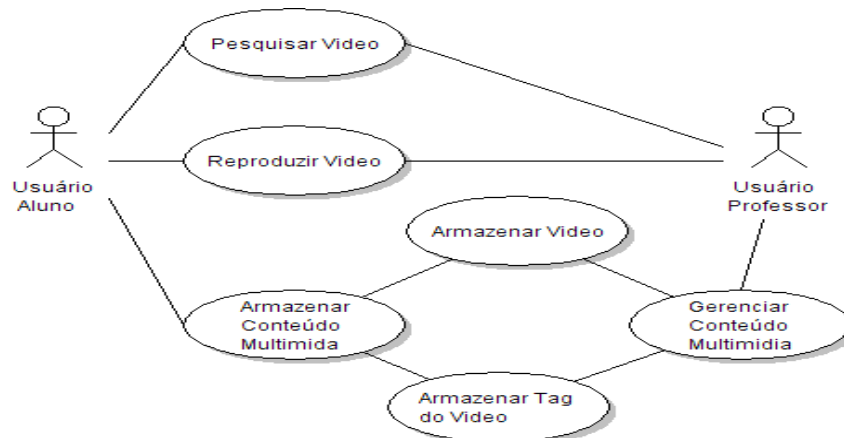


Figura 2 – Funcionalidades da aplicação indexVideo

O diagrama da Figura 2 mostra a estrutura geral do sistema, onde existem tarefas específicas do usuário aluno e específicas do usuário professor e as tarefas comuns aos dois tipos de usuários. A pesquisa de conteúdo multimídia pode ser realizada por qualquer usuário, informando os conteúdos a serem selecionados. Os registros localizados nas *tags* dos arquivos XML se dão por meio do mecanismo de busca semântica.

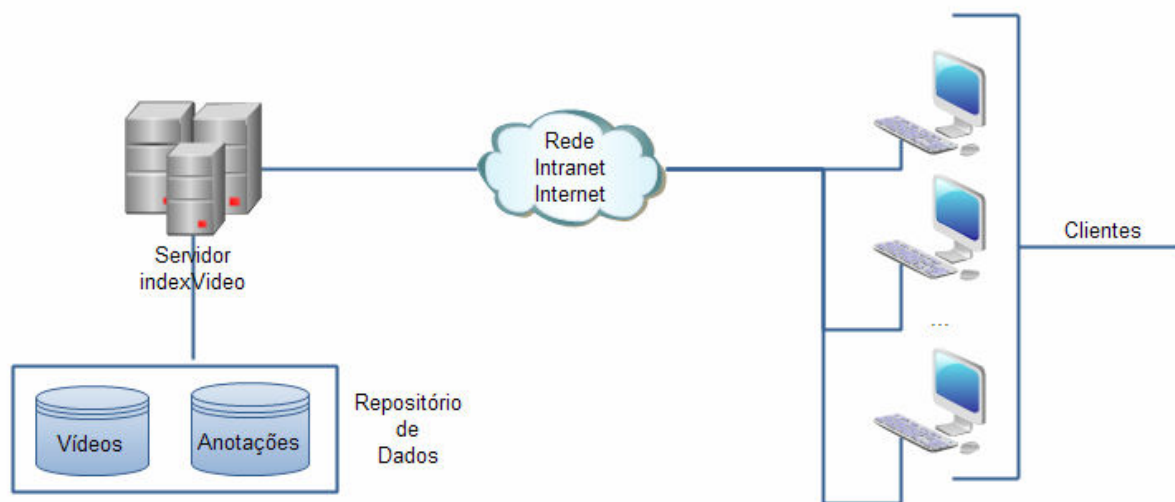
A reprodução de vídeo se dá ao selecionar o conteúdo multimídia retornado na pesquisa, e a aplicação indexVideo irá reproduzir o trecho selecionado pelo usuário. O

armazenamento de conteúdos multimídia ocorre através da postagem de vídeos pelos usuários e inclusão de tags. Estas tags são os comentários e seleções de trecho dos vídeos salvos. Caberá ao usuário professor gerenciar este conteúdo multimídia (CARBONERA, 2007).

2.2 ESTRUTURA DA APLICAÇÃO

A aplicação foi desenvolvida seguindo o modelo cliente-servidor demonstrado na Figura 3 e possui as seguintes funcionalidades (CARBONERA, 2007):

- Armazenamento de vídeos e suas respectivas anotações em repositório remoto: é o local disponibilizado à aplicação para salvar todas as anotações e os vídeos geridos pela mesma, para posteriormente ser recuperado e até mesmo adicionar novas anotações.
- Visualização de conteúdos multimídia com as anotações associadas: permite ao usuário verificar os vídeos relacionados a uma anotação ou as anotações dos vídeos.
- Fazer anotações de trechos de vídeos selecionados pelo usuário.
- Localizar, recuperar e exibir trechos de vídeo e das informações associadas através de busca semântica nas anotações.



*Figura 3 – Diagrama esquemático da arquitetura cliente-servidor.
(CARBONERA, 2007)*

O módulo servidor possui as seguintes tarefas (CARBONERA, 2007):

- Gerenciamento do repositório de vídeos e as devidas anotações: análise dos conteúdos salvos, sendo que o momento de salvar os vídeos o mesmo verifica a disponibilidade do serviço e do repositório. Através deste mecanismo é feita a recuperação

das informações solicitadas pelo usuário através dos parâmetros informados na interface gráfica disposta ao usuário.

□ Gerenciamento das conexões com os clientes remotos: é um módulo que fica analisando as informações trafegadas pela aplicação, garantido a entrega das solicitações para os usuários corretos. No momento que se abre uma conexão entre um cliente e o servidor é necessário este monitoramento das mensagens trafegadas entre as partes identificando a origem e o destino, eliminando a possibilidade de haver entrega de informações para o cliente errado.

□ Tratar as requisições dos clientes: está embutido neste processo, receber e fornecer vídeos e as respectivas anotações solicitadas pelo cliente, além de efetuar as buscas no repositório enviando o resultado ao cliente que as solicitou.

A aplicação servidor monitora as conexões e as solicitações dos clientes e possui uma interface gráfica de visualização do que está sendo solicitado pelos usuários e o que está trafegando pela aplicação conforme demonstrado na Figura 4.

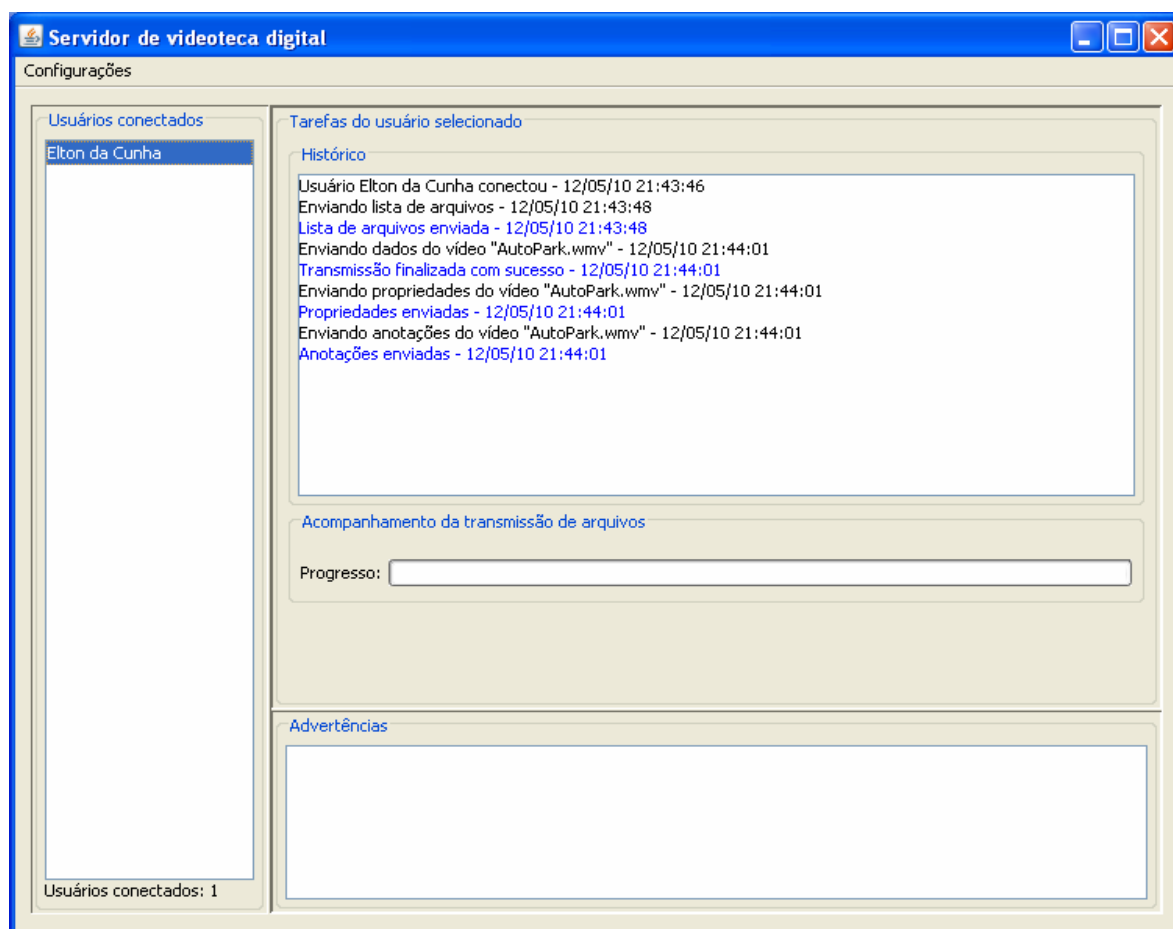


Figura 4 – Interface Gráfica – Aplicação Servidor

Na Aplicação cliente existe uma divisão, a qual possui dois sub-módulos: anotação e recuperação de vídeo. O módulo de anotação possui as seguintes funcionalidades (CARBONERA, 2007):

- Realizar *upload* de vídeos para o servidor remoto: adicionar um vídeo que está na máquina do usuário para a aplicação indexVideo, assim compartilhando o vídeo com os demais usuários, e possibilitando que todos vejam o vídeo salvo e possam fazer anotações no mesmo;
- Reproduzir vídeos armazenados: permite que quem esteja visualizando o vídeo possa fazer anotações;
- Anotar trechos de vídeos em reprodução: pode auxiliar na busca do conteúdo multimídia;
- Recuperar as anotações armazenadas: mostra o relacionamento com os vídeos a que se refere anotação mencionada, trazendo todas as informações da mesma.

O módulo cliente possui uma interface gráfica, que permite a consulta e a reprodução de conteúdos multimídia, conforme demonstrado na Figura 5.

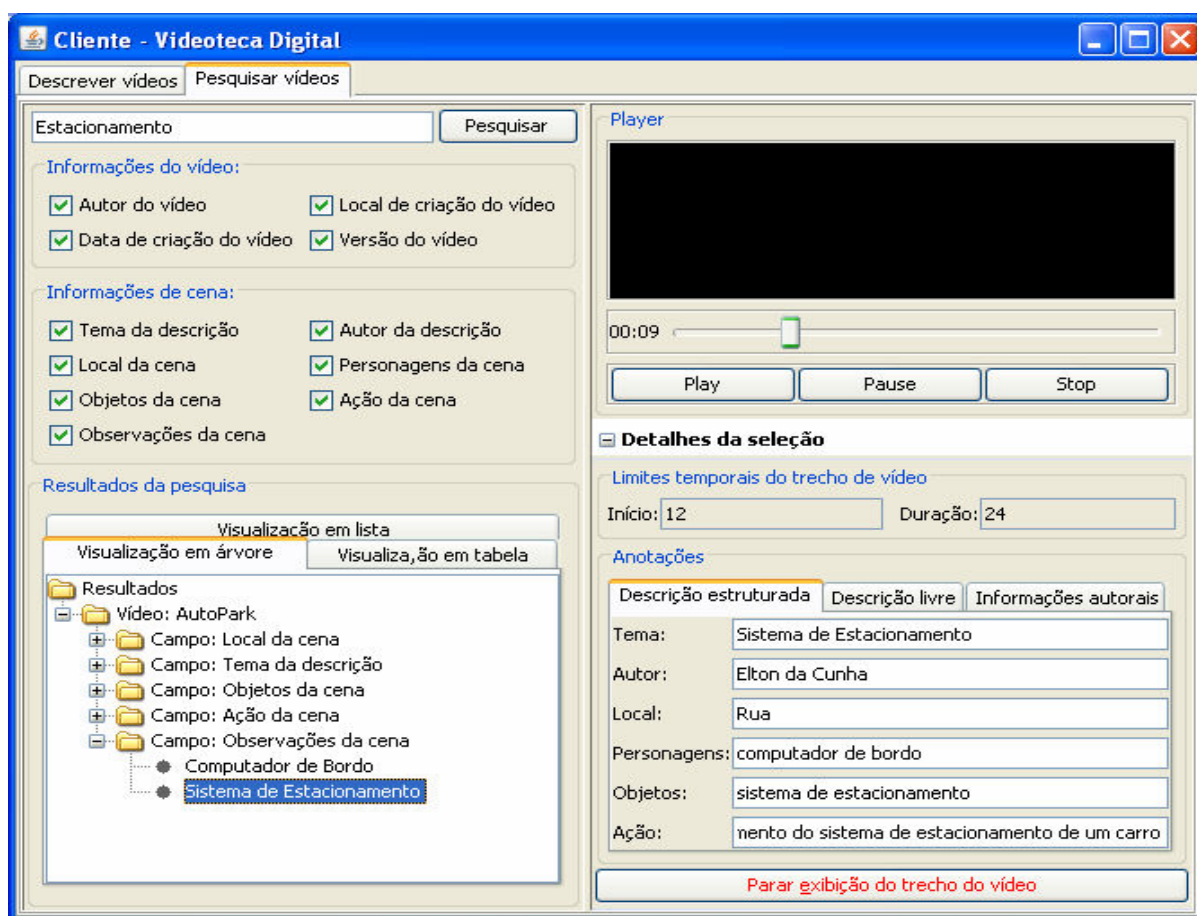


Figura 5 – Interface com Usuário – Consulta Anotações

A aplicação indexVideo possui dados sobre a anotação que segue o padrão MPEG-7. As informações utilizadas são explicadas a seguir:

- Tema da descrição: é o rótulo da anotação do que se trata o tema.
- Autor da descrição: identificação do autor da descrição, o responsável por aquela anotação.
- Local da cena: é a descrição do local registrado no trecho de vídeo.
- Personagem da cena: autor e/ou agente de relevância presentes na cena selecionada.
- Objetos da cena: objetos e/ou artefatos de relevância presentes da cena.
- Ação da cena: ações e efeitos registrados na cena.
- Observações da cena: campo textual destinado a descrições referente a detalhes do trecho do vídeo selecionado.
- Guia descrição livre: contém um campo o qual foi concebido para colocar informações diversas, observações sobre o vídeo.
- Guia informações autorais: possui dados do autor e personagens contidos no vídeo associado.

O módulo de recuperação de anotações faz busca semântica nos arquivos XML contida no repositório do sistema, solicitando ao servidor as anotações conforme especificação informada pelo usuário. A aplicação servidor irá retornar a aplicação cliente com os conteúdos localizados para posterior reprodução do conteúdo multimídia pela aplicação cliente.

2.3 CARACTERÍSTICAS DA IMPLEMENTAÇÃO

A aplicação indexVideo foi desenvolvida em linguagem de programação *Java*², utilizando bibliotecas para execução de vídeo. A biblioteca *JVLCPlayer*³ implementa o padrão MPEG-7. Para a comunicação entre cliente e servidor foi adotada a *API Socket*. Para o desenvolvimento das interfaces gráficas de interação com o usuário final utilizou-se a biblioteca *Swing*⁴ do *Java*. Devido à necessidade de a aplicação rodar sob um navegador de

² <http://java.sun.com/docs/books/tutorial/java/concepts/>

³ <http://javaplayer.sourceforge.net/doc/vlc/JVLC.html>

⁴ <http://java.sun.com/docs/books/tutorial/ui/index.html>

Internet utilizou-se *Applet*⁵, com a finalidade de embutir o programa em um navegador *web* (CARBONERA, 2007).

A aplicação está distribuída em pacotes entre cliente e servidor. A comunicação entre os pacotes ocorre através de objetos, com os seus métodos, conforme padrão de programação orientada a objetos, demonstrado na Figura 6.

A aplicação utiliza o padrão MPEG-7 integrado com a tecnologia XML. O XML é usado para armazenar as informações necessárias para a reprodução dos vídeos. Para manter a compatibilidade dos dados entre as tecnologias MPEG-7 e XML segue-se o padrão descrito pela MPEG, visando à reprodução do conteúdo multimídia atrelado com as anotações pertinentes.

Para as anotações trafegarem sem perda de dados e serem entregues aos clientes que solicitaram de maneira correta, utiliza-se *sockets* que prevê toda a estrutura de comunicação de dados entre cliente-servidor.

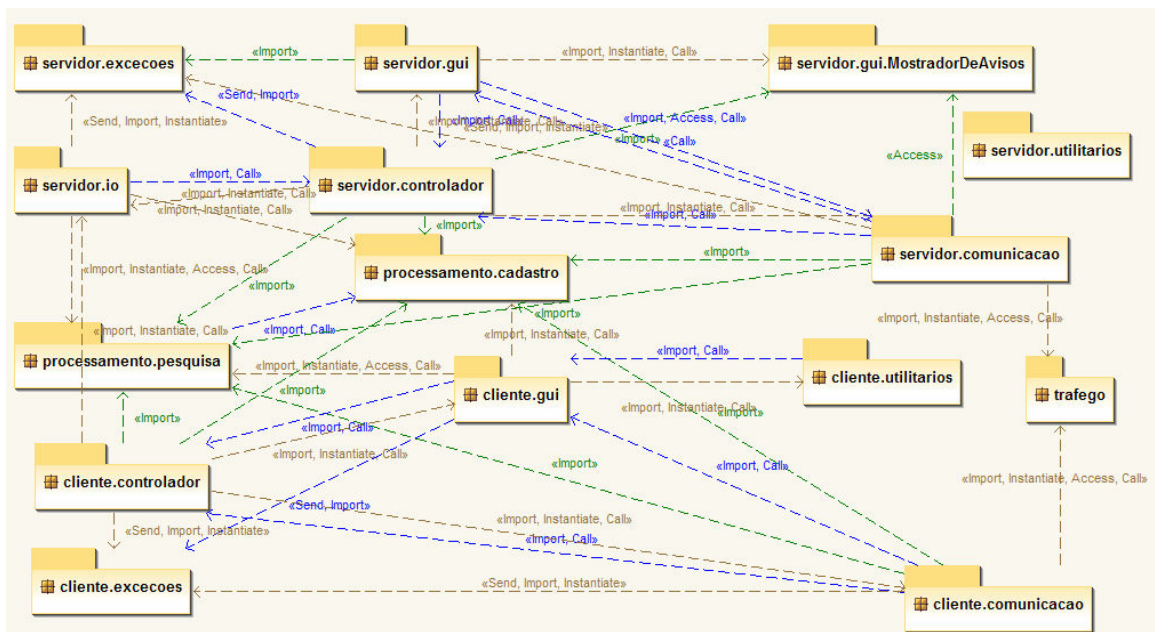


Figura 6 – Comunicação entre pacotes da aplicação.

A aplicação IndexVideo está distribuída em pacotes, onde existe pacotes de interface, persistência de dados e comunicação. Esta distribuição ocorre tanto para o servidor quanto ao cliente, conforme demonstrado na Figura 7, assim facilitando à manutenção dos programas, pois se segue um padrão, deixando toda a estrutura organizada e limpa.

⁵ <http://java.sun.com/docs/books/tutorial/deployment/applet/index.html>

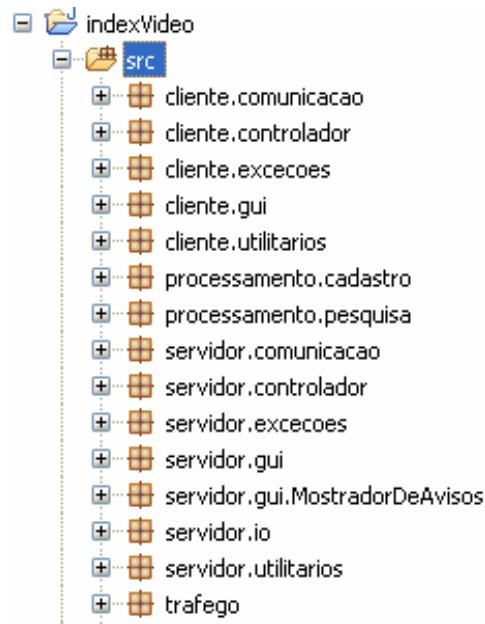


Figura 7 – Estrutura de pacotes da aplicação.

A aplicação indexVideo, no que se refere ao pacote servidor, foi distribuída entre as seguintes camadas:

- servidor.comunicacao: responsável pela comunicação entre cliente servidor e o tráfego entre as partes e os tipos de dados que trafegam entre a aplicação.
- servidor.controlador: tem como função criar e controlar a interface gráfica da aplicação servidor, a qual serve para comunicar-se com o usuário final, auxiliando-o no gerenciamento da aplicação, e inicializando funções vitais para a aplicação tais como abertura de porta de comunicação e configurações iniciais necessárias.
- servidor.excecoes: gerenciamento das exceções geradas pela aplicação servidor.
- servidor.gui: criação de toda a interface gráfica de comunicação com o usuário final.
- servidor.io: tem por finalidade gerenciar os dados trafegados entre o cliente e o servidor e/ou vice-versa, possuindo tratativas de acesso aos dados que a aplicação necessita gerenciar (vídeos e anotações), e a manipulação dos arquivos XML contendo as tag's do vídeo e os vídeos .
- servidor.utilitarios: possui métodos para gerenciar o tempo, data, hora, minuto e segundo.

A aplicação cliente possui as seguintes divisões de camadas:

- cliente.comunicacao: gerencia os dados da aplicação, envia e recebe as informações e responsabiliza-se pelas informações trafegadas entre cliente servidor.

- cliente.controlador: responsável pela criação e execução da interface gráfica e inicialização de métodos necessários para tornar a aplicação cliente operante, tais como abertura de comunicação com a aplicação servidor.
- cliente.excecoes: gerenciamento das exceções geradas pela aplicação cliente;
- cliente.gui: contém todas as interfaces gráficas de comunicação visual com o usuário final.
- cliente.utilitarios: possui métodos para controlar data, hora, minuto, segundo, controle de execução e reprodução de vídeo e processos de transmissão de vídeo.

Entre os pacotes de processamento de cadastro e pesquisa estão as seguintes funções :

- processamento.cadastro: contém os objetos que a aplicação utiliza para trafegar entre cliente servidor e serve também quando o mesmo irá salvar, alterar ou excluir as informações da base de dados.
- processamento.pesquisa: possui instruções de localização de dados conforme informações passadas pela interface do usuário para esta camada, que por sua vez irá pesquisar na base de dados e, num momento posterior, retornar o resultado ao usuário final .

O pacote tráfego contém informações das mensagens que serão trocadas entre cliente servidor e o formato que as mesmas terão que trabalhar para se comunicar, assim tendo um padrão para saber os resultados.

2.4 CAMADA DE PERSISTÊNCIA DA APLICAÇÃO INDEXVIDEO

Na aplicação indexVideo, o XML foi associado à tecnologia MPEG-7 devido a compatibilidade entre as mesmas. O XML tem a função de armazenamento de dados sobre os vídeos geridos por esta aplicação (CARBONERA, 2007).

O indexVideo possui um arquivo XML contendo os descritores para cada vídeo armazenado na biblioteca, independente se o XML terá descrição ou não. O arquivo XML correspondente terá que ser criado pela aplicação, caso isso não ocorra a estrutura para reprodução do vídeo não está completa e íntegra. A Figura 8 ilustra um modelo de armazenamento dos dados.

O XML possui uma estrutura em que constam todos os dados necessários para reprodução e execução do vídeo. Estrutura definida e padronizada pela MPEG, a criadora da tecnologia MPEG-7, conforme apresentado na seção 2.4.2 (MPEG-7, 2009).

A MPEG prevê a necessidade da utilização do XML para fazer a integração entre os dados do vídeo e descritor, porém, não menciona que estes dados tenham de ser armazenados em XML, mas faz-se necessária para a sua reprodução. O XML seria uma espécie de comunicador entre tecnologias, os dados poderiam estar armazenados em qualquer outro tipo de estrutura e utilizar o XML para integrar com o MPEG-7 (MPEG-7, 2009) (CARBONERA, 2007).

A estrutura do arquivo XML correspondente de um vídeo contempla informações do *schema* do XML, versão, autor, data, hora, caminho do repositório e *tags* marcadas no vídeo pelos usuários da aplicação. A Figura 8 ilustra uma demonstração da estrutura do XML.

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
- <Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"
  xsi:schemaLocation="urn:mpeg:mpeg7:schema:2001 Mpeg7-2001.xsd">
- <DescriptionMetadata>
  <Version>1.01</Version>
- <Creator>
  - <Role href="creatorCS">
    <Name>BMW</Name>
  </Role>
  </Creator>
- <CreationLocation>
  <AdministrativeUnit>Auto Estrada</AdministrativeUnit>
  </CreationLocation>
  <CreationTime>2010-05-12T00:00:00-03:00</CreationTime>
</DescriptionMetadata>
- <Description xsi:type="ContentEntityType">
- <MultimediaContent xsi:type="MultimediaCollectionType">
  - <Collection xsi:type="ContentCollectionType">
    - <Content xsi:type="VideoType">
      - <Video>
        - <MediaLocator xsi:type="TemporalSegmentLocatorType">
          <MediaUri>D:\Projetos\Java\indexVideoRepositorio\videos\AutoPark.wmv</MediaUri>
        </MediaLocator>
        - <TemporalDecomposition>
          - <VideoSegment id="3">
            - <TextAnnotation>
              <FreeTextAnnotation xml:lang="pt">sistema de estacionamento</FreeTextAnnotation>
            </TextAnnotation>
            - <StructuredAnnotation>
              - <WhatTheme>
                <Name xml:lang="pt">Computador de Bordo</Name>
              </WhatTheme>
            </Author>
              <Name xml:lang="pt">Elton da Cunha</Name>
              <UserID xml:lang="pt">epcunha</UserID>
            </Author>
            - <Where>
              <Name xml:lang="pt">Rua / Estacionamento</Name>
            </Where>
            - <Who>

```

Figura 8 – Estrutura do XML que armazena os dados de um vídeo.

O arquivo XML contém as seguintes *tags* essenciais para o funcionamento da aplicação:

- DescriptionMetadata: tag que contém os dados do vídeo, tais como versão, nome, data e hora.
- MediaLocator: contém o caminho do repositório do vídeo, juntamente com o nome do vídeo.

□ TemporalDecomposition: contém as informações dos trechos selecionados nos vídeos, como personagens, unidade temporal, usuário e local.

A estrutura demonstrada na Figura 8 é criada no momento em que se faz o *upload* do vídeo para aplicação, podendo vir a receber manutenção das informações, conforme as instruções passadas pelo usuário final da aplicação. A criação das tags de anotação pode ser feita em qualquer momento da reprodução do vídeo, e todas estas alterações são salvas dentro de um único arquivo XML, correspondente ao vídeo.

2.4.1 XML – VANTAGENS E DESVANTAGENS NA APLICAÇÃO

A tecnologia MPEG-7 adotou como padrão de armazenamento de dados a estrutura XML. As informações que serão geridas por esta tecnologia são dispostas dentro de arquivos XML com *tags* para facilitar a recuperação dos dados necessários para a reprodução dos conteúdos audiovisuais (CHELLA, 2003).

As duas tecnologias se fundem entre si de forma simples e transparente, pois o XML é um padrão nativo de armazenamento de dados para a plataforma do MPEG-7. Nesta estrutura estão previstas todas as informações necessárias para gerir os conteúdos audiovisuais selecionados (CHELLA, 2003).

A pesquisa de conteúdos em arquivos XML se torna difícil, pois no momento que o usuário final irá buscar a informação, a aplicação *indexVideo* terá que abrir todos os arquivos XML varrendo o conteúdo de cada um para conseguir localizar as informações solicitadas pelo usuário, as quais estão contidas dentro das tags de vários arquivos XML. Esta recuperação é demorada e não confiável, podendo haver falhas de sistema na localização da informação.

Ao executar a leitura de um arquivo em um momento “X” e o mesmo está sendo alterado por outro usuário, significa que a informação recuperada no momento “X-1” já não existe mais. No formato atual da aplicação não temos como realizar o bloqueio da informação para execução de um procedimento de leitura ou escrita, pois a tecnologia XML não possui estas tratativas integradas a sua estrutura, ficando tudo a cargo da aplicação final.

Ao armazenar dados em XML não é possível fazer persistência de dados, bloqueio de informações no momento das atualizações, criações e deleções. Sem poder prever este tipo de operação, a consistência dos dados pode estar comprometida, podendo haver falhas no

armazenamento das informações. Caso haja dois usuários ao mesmo tempo fazendo alteração no mesmo dado, e um dos usuários verifica que a sua operação não foi executada, isso significa que uma das operações de um dos usuários não persistiu na base de dados, ocorrendo a subscrição de dados, ou seja, uma operação foi executada sobre a outra, parecendo que a primeira operação não existiu (GRAVES, 2003).

Não é possível gerir uma estrutura de *back-up* eficiente com arquivos XML, devido à atualização dos dados no momento que estará sendo executada a rotina de *back-up*. Neste formato empregado na aplicação indexVideo, não se consegue abrir uma janela de tempo no sistema, para tirar uma “foto” dos dados íntegros e consistidos. Se mesmo assim for feito o back-up dos arquivos XML, poderá vir a acarretar a inconsistência do *back-up*. E numa possível restauração não se terá os dados íntegros, ou seja, parte das informações atualizadas e outra parte desatualizada. A tratativa eficiente de *back-up* se torna possível em uma estrutura adequada, com ferramentas apropriadas para desempenhar esta tarefa, que dentro de um SGBD já está previsto em suas diretivas de trabalho.

2.4.2 MPEG-7

A tecnologia MPEG-7 foi desenvolvida e padronizada pela MPEG, sob os padrões ISO/IEC, com o intuito de codificar e reproduzir conteúdos áudiovisuais, a qual possui ferramentas para descrição multimídia. Ao desenvolverem esta tecnologia visou-se à usabilidade perante os usuários, de forma que os mesmos possam pesquisar recuperar e alterar estes conteúdos áudiovisuais (MARTÍNEZ, 2002).

Segundo (MARTÍNEZ, 10/2004) os principais elementos do MPEG-7 são:

- Descritores: descrição de características tais como cor, textura, tempo e etc.
- Esquema de descrição: descrição dos conteúdos audiovisuais.
- Linguagem de definição de descrição: é uma extensão dos esquemas de descrição já criados, onde pode conter especificações permitindo a criação de novos esquemas de descrição.
- Ferramentas de descrição: são elementos que dão suporte a multiplexação e sincronização de descrições com seus devidos conteúdos.

As ferramentas desta tecnologia permitem a criação de descritores de vídeo analógicos e digitais, dados multimídia, os quais são implicados no contexto da aplicação indexVideo e

são chamados de *Description Tools* representada por metadados, que possui descritores e seus esquemas são definidos no *Description Schemas* (CHELLA, 2003).

Ao tratarmos de armazenamento de dados da estrutura necessária para o MPEG-7, acaba-se utilizando a tecnologia XML como padrão para integrar os dados de especificações do conteúdo audiovisual, dando total suporte entre as tecnologias XML com MPEG-7 para desempenhar a tarefa de execução e descrição destes conteúdos audiovisuais (CHELLA, 2003).

O XML Schema é a base para a DDL (*Description Definition Language*), a qual é utilizada para a definição sintática dos descritores contidos no MPEG-7. Devido às facilidades que se possui entre as duas tecnologias MPEG-7 e XML, é que se torna uma compatibilidade transparente e de fácil utilização para contexto da aplicação indexVideo(CHELLA, 2003).

2.4.3 PADRÃO MPEG-7 VERSUS XML SCHEMA

A biblioteca áudio visual MPEG-7, utiliza uma estrutura padrão para suprir as informações necessárias para a reprodução de conteúdos multimídia, prevendo as anotações de vídeos e de dados, como a localização do arquivo em seu repositório, autor, e informações complementares. Estas informações estão em uma estrutura de dados sob a tecnologia XML, que segue um *schema* adotado pela W3C. A versão mais atual do XML *Schema* disponível até o momento é a versão de 2001 que utiliza o mesmo padrão do XML *Schema* 1999, porém com algumas melhorias de estrutura, abaixo na Figura 9 do XML *Schema* 2001, adotado pela W3C, órgão normatizador desta estrutura (MARTÍNEZ, 2004).

```
<?xml version="1.0" ?>
<!DOCTYPE xs:schema (View Source for full doctype...)>
- <xs:schema targetNamespace="http://www.w3.org/2001/XMLSchema-instance" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.w3.org/1999/xhtml"
  finalDefault="*" blockDefault="*" elementFormDefault="unqualified" attributeFormDefault="unqualified">
- <xs:annotation>
- <xs:documentation>
  <h1>XML Schema instance namespace</h1>
- <p>
  See
  <a href="http://www.w3.org/TR/xmlschema-1/">the XML Schema Recommendation</a>
  for an introduction
</p>
<hr />
$Date: 2001/03/16 20:25:57 $
<br />
$Id: XMLSchema-instance.xsd,v 1.4 2001/03/16 20:25:57 ht Exp $
</xs:documentation>
</xs:annotation>
+ <xs:annotation>
  <xs:attribute name="nil" />
  <xs:attribute name="type" />
  <xs:attribute name="schemaLocation" />
  <xs:attribute name="noNamespaceSchemaLocation" />
</xs:schema>
```

Figura 9 – XML Schema 2001 do MPEG-7.
(W3C, 2001).

Sob este padrão funciona a plataforma MPEG-7 demonstrada na Figura 9, a qual necessita de um XML de anotação para poder executar a multiplexação, e sincronizar a descrição com o conteúdo áudiovisuais. As descrições desenvolvidas com estas ferramentas não dependem de como o seu conteúdo está codificado ou armazenado e sim como está estruturado. A plataforma MPEG-7 necessita desta estrutura para poder interpretar as informações e conseguir sincronizar os dados, assim disponibilizando ao usuário final, o vídeo associado com sua descrição e unidade temporal (MARTÍNEZ, 2004).

2.5 CONSIDERAÇÕES FINAIS

Sendo o XML uma estrutura que pode ser empregada como banco de dados na ideia de armazenar informações diversas e/ou integrar/interagir com outras aplicações que a utilizem como estrutura, neste contexto têm-se desde um arquivo para comunicação entre aplicações até um repositório de dados (banco de dados) (GRAVES, 2003).

O XML é versátil, podendo ser utilizado em várias aplicações, desde um simples arquivo de comunicação até como um sistema de armazenamento de dados em massa. A definição de utilizar esta tecnologia depende de cada caso, cabendo a equipe de desenvolvimento de um sistema avaliar qual é a melhor alternativa que se adeque ao problema.

3 ALTERAÇÕES PROPOSTAS PARA CAMADA DE ARMAZENAMENTO DO INDEXVIDEO

Analisando a aplicação indexVideo e a estrutura dos dados necessárias para sua execução, verificou-se a necessidade de alterar a forma de armazenamento dos dados. Para isso, será realizada alterações na camada de persistência de dados, e adequação na geração do arquivo XML para integrar com o MPEG-7. Serão analisadas os impactos sofridos pela alteração, testes de execução, e possíveis ajustes.

Também se faz necessário analisar o melhor SGBD que se adeque a estrutura de dados necessárias para armazenar os dados da aplicação indexVideo, os ganhos e as perdas que a aplicação teria com a alteração sofrida. Verificar entre as formas de armazenamento de dados, XML e SGBD qual trará melhor custo benefício à aplicação indexVideo.

3.1 ANÁLISE DE USO DE UM SGBD

Para analisarmos o melhor SGBD que se adeque a realidade da aplicação indexVideo se faz necessário um mapeamento completo dos dados a serem geridos pela aplicação indexVideo e pelo SGBD identificando a estrutura através de um modelo relacional.

O modelo relacional deverá conter toda a estrutura de dados necessária para gerir a aplicação, contemplando o padrão MPEG-7 e prever a existência de todas as informações já existente no XML referente ao vídeo.

Os SGBD'S existentes no mercado, possuem várias funcionalidades e propriedades. Analisando as mesmas poderemos diagnosticar qual é a mais coerente à realidade do indexVideo, assim disponibilizando as funcionalidades do SGBD para sua perfeita integração.

3.1.1 SGBD

SGBD (Sistema Gerenciador de Banco de Dados) é um conjunto de programas associados a uma base de dados onde se faz uso dos programas para gerir as informações armazenadas nesta base; podendo ser geridas informações dos mais diversos assuntos, organizado em uma estrutura pré-definida, tendo por objetivo principal proporcionar um ambiente confiável e eficiente para o armazenamento e recuperação das informações nele dispostas. Garante-se a integridade das informações, não permitindo que problemas

adjacentes interfiram nos dados armazenados e muito menos que sofram alterações indesejadas (SOLOMON, 2004).

Os SGBDS são projetados para trabalhar com volumes grandes de dados e acesso simultâneo ou não. O gerenciamento destes dados implica na concepção da estrutura a ser utilizada para armazenamento das informações; na definição dos mecanismos de manipulação dos dados, tais como inclusão, exclusão, alteração e consulta das informações nele armazenada, também prevendo privilégios de acesso aos usuários, e as possíveis falhas de sistema causadas pelo meio o qual está se trabalhando esta base de dados, independente da plataforma a qual está instalada o SGBD (SOLOMON, 2004).

No momento que ocorrer uma falha de sistema, o SGBD em questão deverá usar os seus mecanismos de defesas para proteger os dados nele armazenados, não permitindo violação da informação, inconsistência da base de dados, prevendo possíveis falhas de transmissão. Só permitir salvar os dados se a informação nele disposta está completa e íntegra, assim terá uma estrutura ideal para *back-up*. Compete ao SGBD prover a cópia e a restauração da informação íntegra e sem falhas ou perdas de dados, deixando a base de dados consistida em um momento “X” no tempo (NAVATHE, 2005).

A base de dados gerida por um SGBD deve estar sempre íntegra e completamente disponível em tempo real para utilização destes dados pelos sistemas adjacentes ou a usuários que possuam permissão de acesso a esta base. Sendo um dos principais objetivos destes sistemas, evitar operações que resultem em anomalias nos dados nele concebidos, eliminando a inconsistência de dados, ou a tentativa de salvar dados incompletos ou com falhas (NAVATHE, 2005).

A inconsistência da informação, ou informação não íntegra, refere-se a uma possível falha do SGBD que não conseguiu atualizar o dado totalmente em sua base de dados ou resultou em uma alteração parcial ou incompleta, remetendo a uma falha de dados, deixando uma lacuna nas informações, assim prejudicando o funcionamento dos sistemas adjacentes. Por estes motivos, se faz necessário, a implementação de mecanismos de consistência de dados, sempre visando a integridade das informações dispostas na base de dados (NAVATHE, 2005).

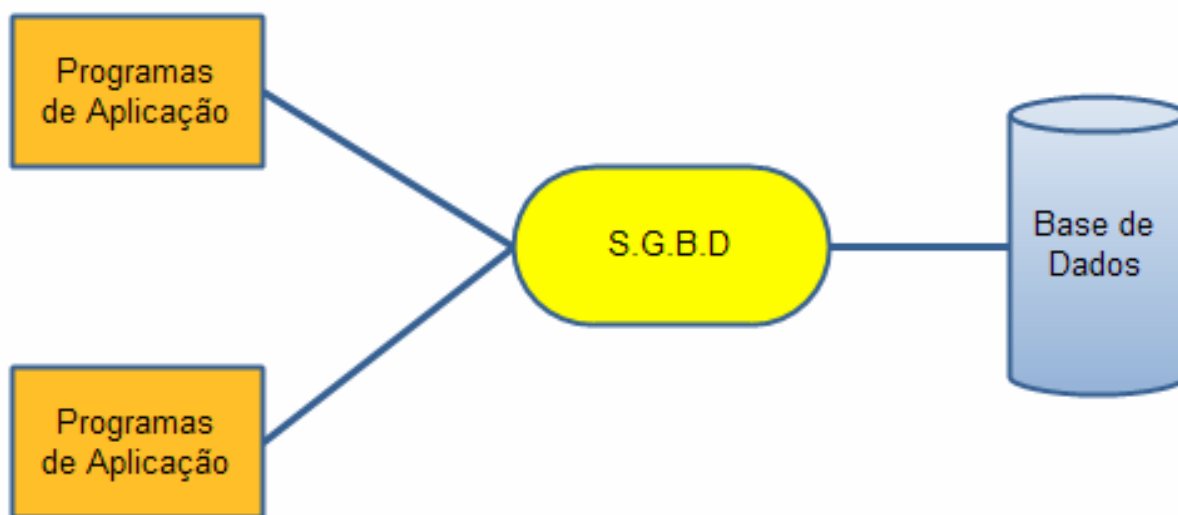
Os mecanismos de consistência de dados preveem políticas de gravação e recuperação da informação, basicamente atuando no momento da gravação dos dados para poder verificar a integridade antes de salvar o dado na base de dados (NAVATHE, 2005).

Além do mecanismo de consistência um SGBD possui outras funções necessárias para sua operação, como recuperação de falhas, recuperação de base de dados, recuperação de

índices, criação de visões, dando permissão parcial ou total à visualização de um dado para determinado usuário, liberando e bloqueando funções por usuário e atividade a ser realizada e possibilitando com isso a melhor segurança e privacidade das informações nele contidas (NAVATHE, 2005).

3.1.2 COMPONENTES DO SGBD

Um SGBD possui três componentes principais: programas de aplicação, SGBD e base de dados, demonstrados na Figura 10 (GONÇALO, 2006).



*Figura 10 – Esquema de SGBD.
(GONÇALO, 2006).*

Cada um dos componentes possui uma funcionalidade específica. Os programas de aplicação fazem a parte de comunicação entre usuário e sistema, prevendo a forma de utilizar as informações e a sua disposição. O SGBD é o sistema que gerencia os dados, faz a comunicação entre a aplicação e o BD, podendo receber a requisição de vários usuários e de várias aplicações ao mesmo tempo; todas as validações, persistência, consistência, instruções de back-up, posicionamento de dados no BD é feita sobre o SGBD. O SGBD é responsável pela integridade das informações armazenadas em sua estrutura. O BD é o local onde ficam armazenados os dados ordenados e combinados, conforme instruções passadas pelo SGBD, sendo a única ferramenta que possui permissão para fazer a interação direta aos dados. O SGBD é a ferramenta que deixa tudo organizado na estrutura previamente definida, e

deixando disponível para sua utilização em qualquer momento, que seja solicitado (GONÇALO, 2006), (POSTGRESSQL, 2010).

3.1.3 OBJETIVOS DE SGBD

O SGBD possui objetivos que serão abaixo descritos e explicados. Eles servem para nortear o desenvolvimento de um SGBD e definir as estruturas a serem aplicadas no propósito de gerir as informações de uma base de dados.

□ Inconsistência e redundância de dados: em uma base de dados se faz necessário a redundância de dados para haver a ligação das informações entre vários arranjos que possam ser montados, porém, se a redundância da informação não for controlada de maneira clara e objetiva pode se tornar um problema, que acarretará em uma inconsistência nos arranjos, criando uma falha de dados e deixando a informação incompleta. Esta falha ocorre quando há divergência entre os dados de um mesmo fato, assim desperdiçando espaço de armazenagem e falhas nas informações, não aproveitando o recurso adequadamente (POSTGRESSQL, 2010).

□ Flexibilidade, integridade, segurança e independência: a flexibilidade ao acessar as informações baseado no critério determinado pelo usuário (aplicação de usuário final), tornando simples a recuperação e acesso aos dados (POSTGRESSQL, 2010).

Integridade e segurança, cabendo ao administrador definir privilégios de acesso e programar mecanismos de integridade da base de dados visando à confiabilidade e a preservação da mesma, não permitindo situações que possam levar ao descontrole dos dados, ou seja, a perda da confiabilidade das informações (POSTGRESSQL, 2010).

O SGBD é um intermediador entre a base de dados e os usuários ou programas de aplicação, tornando os dados totalmente independentes do SGBD e de programas e usuários finais, podendo transportá-los entre SGBD's sem haver inconsistência na base de dados. Para isso ocorrer é necessário adequações dos dados conforme padrão do SGBD que irá operar, pois cada SGBD possui um padrão de conversação com a base de dados, que é especificado pelo fabricante ou detentor do SGBD a ser utilizado (NAVATHE, 2005), (POSTGRESSQL, 2010).

Com a finalidade de prover informações e informações verídicas os SGBD's que existem hoje em utilização pelo mercado de software possuem objetivos parecidos, porém, as implementações sofrem grandes alterações de fabricante para fabricante até mesmo em versões do SGBD do mesmo fabricante. Mas o intuito principal é gerir informações da melhor

forma possível e adequada para cada situação, cabendo fazer uma análise o qual melhor se enquadra para resolução de um problema proposto (NAVATHE, 2005).

3.1.4 VANTAGENS E DESVANTAGENS DO SGBD

O SGBD possui uma série de ferramentas que preveem a integridade, preservação e a perpetuidade dos dados, garantindo que os dados nele armazenado não sofram alterações do meio de uma forma indevida sem ser pela estrutura pré-definida (GARCIA; MENEZES; BOSCARIOLI, 2008).

O armazenamento das informações em uma base de dados através de SGBD's são garantidas por procedimentos pré-definidos pelo SGBD, com isso terá a garantia que toda a manipulação dos dados será feita de forma correta, e caso hajam falhas de sistema, o SGBD possui mecanismo de recuperação de dados, ou se identificado antes de salvar a inconsistência, o mesmo não deixará salvar na base (POSTGRESSQL, 2010).

A recuperação de dados se dá através da validação e verificação de inconsistência. O dado será gravado na estrutura quando o SGBD verificar que o dado a ser implantado está completo, íntegro e em condições de salvar em toda a estrutura, e esta manipulação do dado já está definida no momento da implantação do sistema SGBD (POSTGRESSQL, 2010).

A manipulação dos dados tais como recuperação, disponibilização ao usuário, alteração e exclusão, ocorre através de comandos solicitados pelo usuário através de parâmetros definidos. Assim, o SGBD interpretará este comando, e selecionando na base de dados as informações solicitadas, devolve ao usuário as informações que ele deseja. Visa a garantia que a operação solicitada seja executada por completo, não permitindo inconsistências de dados causadas por operações indevidas. No momento que há um dano a um dado em específico, o SGBD prevê rotinas de recuperação e em último caso, a restauração de um back-up (POSTGRESSQL, 2010).

O *back-up* da base de dados é controlado pelo SGBD, tendo como compromisso fazer um *download* completo da base para um repositório pré-especificado, garantido que essas informações possam ser restauradas na estrutura padrão definida pelo gestor do sistema. Este *back-up* contém as informações completas e íntegras da base de dados (POSTGRESSQL, 2010).

A estrutura de *back-up* é específica de cada SGBD, mas seguem uma padronização orientada por normativas definidas por órgãos que gerenciam estes sistemas, tais como a própria W3C em conjunto com o fabricante do SGBD (POSTGRESSQL, 2010).

3.2 COMPARAÇÃO ENTRE XML E SGBD

Um dado em XML é semi-estruturado, podendo a cada ocorrência possuir um esquema em particular tornando-o auto-descritivo. Já com SGBD, teremos uma estrutura previamente definida, rígida. Na estrutura de XML podemos identificar duas categorias: (GARCIA; MENEZES; BOSCARIOLI, 2008).

- Documento XML centrado em dados: sua semi-estruturação é regular e repetitiva, basicamente utilizada em transferência de dados entre aplicações, na *internet* ou até mesmo em geração de relatórios (GARCIA; MENEZES; BOSCARIOLI, 2008).

- Documentos XML centrado em documentos: Sua semi-estrutura é irregular e particular, muito utilizado em relato de linguagem natural, como dados delimitados dentro do seu próprio conteúdo (GARCIA; MENEZES; BOSCARIOLI, 2008).

Os SGBD's tratam os documentos XML baseados nas duas categorias acima mencionadas. As estruturas XML centradas em dados, são relativamente fáceis de serem mapeadas em uma estrutura de SGBD, pois possui uma estrutura repetitiva e regular, e alguns SGBD's já possuem suporte nativo para estruturas XML incluso em suas diretivas (GARCIA; MENEZES; BOSCARIOLI, 2008).

Quanto às estruturas XML centradas em documentos são mais complexas de serem mapeadas em uma estrutura de SGBD, devido a não regularidade contida nas informações. Para persistir estes documentos, existem SGBD's específicos para esta finalidade, chamados de SGBD XML nativos, como por exemplo, o SGBD *Tamino*, tornando mais efetiva esta interação, entre XML e SGBD (GARCIA; MENEZES; BOSCARIOLI, 2008).

Para a manipulação da informação, se faz necessário as ferramentas previstas no SGBD, isso pode deixar as movimentações de dados mais lentas. Analisando sobre a aplicação *indexVideo*, teremos que criar uma camada de conversão entre os dados armazenados e o reprodutor de vídeo, pois o mesmo só aceita os dados no formato XML, devido a padronização da tecnologia MPEG-7 ter como modo nativo de armazenamento de dados o formato XML. Para extrair dados armazenados em um SGBD e ordenar estas informações no padrão que a tecnologia MPEG-7 exige, acarretará em montar uma estrutura

de XML temporária para suprir os dados que alimentaram o reprodutor de vídeo. Esta conversão terá que ser feita em tempo real de execução, impactando em um tempo maior de preparação da massa de dados antes de disponibilizar a utilização pela aplicação.

O tempo de execução das pesquisas será diminuído, pois é mais rápido pesquisar em tabelas as informações solicitadas pelo usuário do que abrir cada arquivo XML. Verificar se o que está sendo buscado está ali armazenado, este seria um grande ganho para a aplicação indexVideo, economizando tempo de resposta do usuário, porém, para executar o vídeo com suas anotações irá perder performance e aumentar o tempo de resposta, devido a geração do arquivo XML em tempo de execução.

3.3 MUDANÇAS PROPOSTAS

Analisando a aplicação e o funcionamento de um SGBD, verifica-se que uma alternativa para melhorar o desempenho da aplicação seria alterar a forma de armazenamento dos dados, eliminando o armazenamento dos arquivos XML e transpondo todos os dados para um SGBD. Para isso, deve-se fazer alterações no nível de aplicação indexVideo.

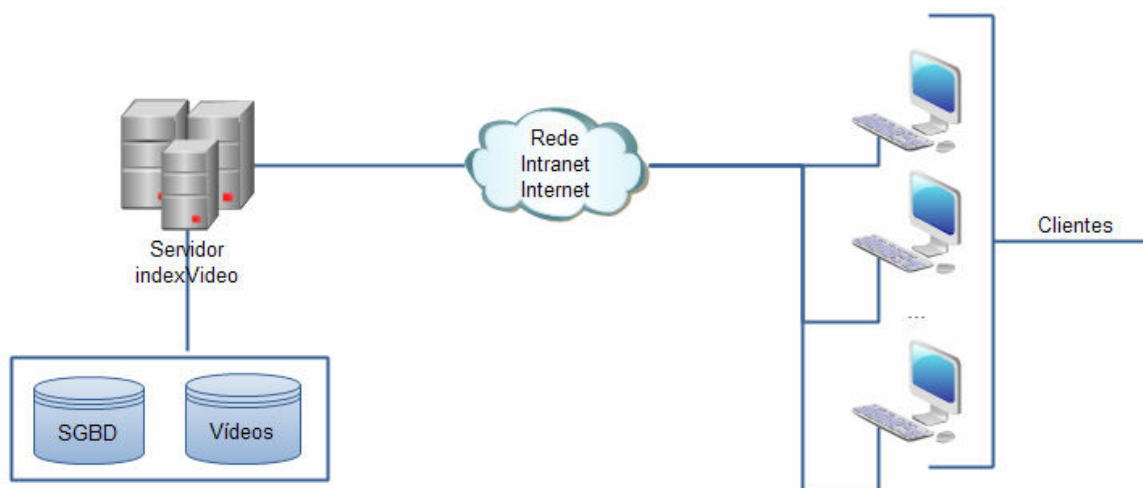


Figura 11 – Diagrama esquemático da arquitetura cliente-servidor proposto.

Na aplicação indexVideo a camada “IO” existente hoje, estruturada para manipular dados em arquivos XML, será alterada para uma camada de persistência de dados, assim evitando que as anotações sejam armazenadas em arquivos XML e passem a ser armazenados em uma estrutura específica de um SGBD, conforme demonstrado na Figura 11. As anotações

serão armazenadas em um SGBD e os vídeos continuarão sendo salvos em um repositório disposto no servidor que está rodando a aplicação indexVideo servidor.

O fluxo proposto da aplicação indexVideo consiste que toda a solicitação de dados que aplicação vier a necessitar, será solicitado diretamente ao SGBD, após o retorno dos dados fornecido, a aplicação irá verificar a operação a ser feita. Caso a solicitação seja a reprodução de um trecho de uma anotação, a própria aplicação irá montar o arquivo XML necessário para a reprodução e enviará a JVLCPayer, o vídeo e o XML. Conforme Figura 12, temos a exemplificação do fluxo proposto.

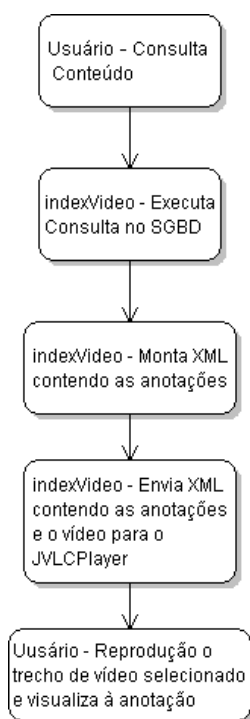


Figura 12 – Fluxo de consulta proposto.

3.3.1 FUNCIONALIDADES NOVAS

Está sendo criada na aplicação uma camada de comunicação com sistema gerenciador de banco de dados, a qual irá fazer toda a manipulação dos dados: criação, exclusão e alteração, responsabilizando-se pela ligação entre a camada de manipulação de estruturas XML com o SGBD.

A camada de manipulação de estruturas XML irá criar os arquivos XML no padrão do MPEG-7, e enviada juntamente com o vídeo para execução de sua reprodução, sendo alterada a forma de pesquisar os dados.

Toda a pesquisa de dados será direcionada a esta nova camada, a qual se encarregará de criar as diretivas de pesquisa mais adequadas e solicitar ao SGBD o dado que foi solicitado pelo usuário, conforme parâmetro que o mesmo solicitou através de sua interface gráfica. Esta camada irá fazer a comunicação com o SGBD, que irá buscar na base de dados as informações, e no momento posterior, com a massa de dados montada irá retornar a essa nova camada os dados que, por sua vez, irá devolver a resposta ao usuário.

Com isso foi elaborado um diagrama de requisitos para adequar a camada de persistência de dados, conforme Figura 13.

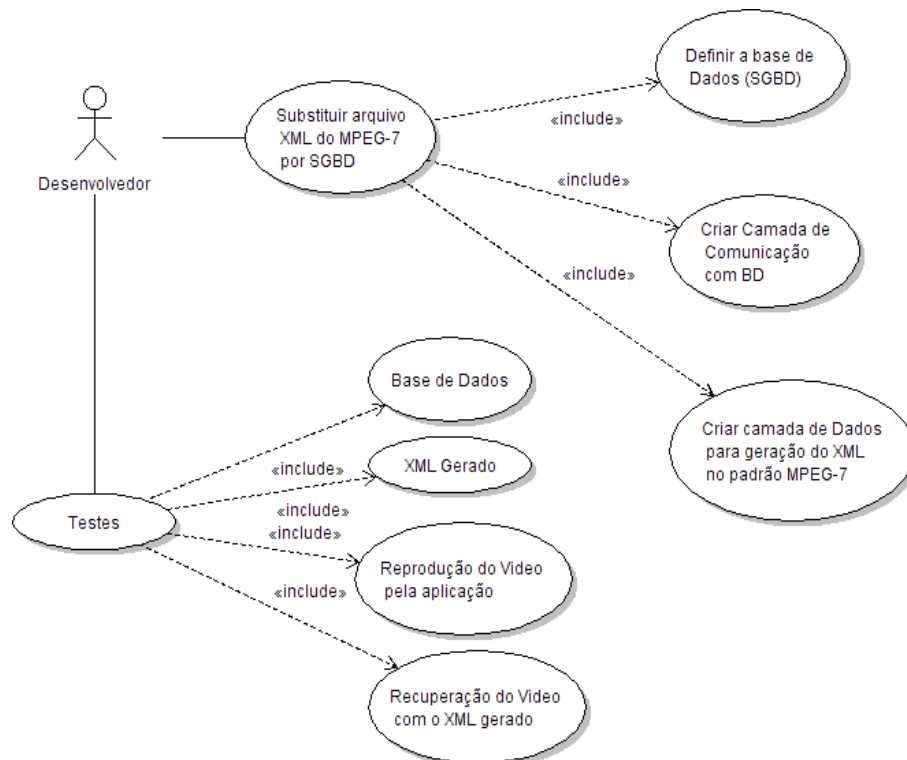


Figura 13 – Requisitos de adequação da camada de persistência.

Conforme o diagrama da Figura 13, o mesmo está dividido em duas etapas:

□ Substituição do arquivo XML por SGBD: esta etapa contém a definição da estrutura da base de dados, a criação da camada de comunicação entre a aplicação e o SGBD utilizado, e a criação de uma camada de transformações dos dados recuperados do SGBD para a estrutura XML compatível com MPEG-7.

□ Testes: nesta etapa, está previsto teste de acesso e consistência na base de dados, geração do XML para suprir a aplicação indexVideo, a reprodução de vídeo pela indexVideo, e a integração entre vídeo, XML e SGBD.

3.3.2 CAMADA DE PERSISTÊNCIA EM SGBD

Para poder analisar a utilização de um SGBD é importante mapear as informações necessárias para gerir a aplicação. Conforme a Figura 14 o modelo relacional provê o mapeamento completo dos dados contidos em um arquivo XML necessários para aplicação indexVideo funcionar corretamente.

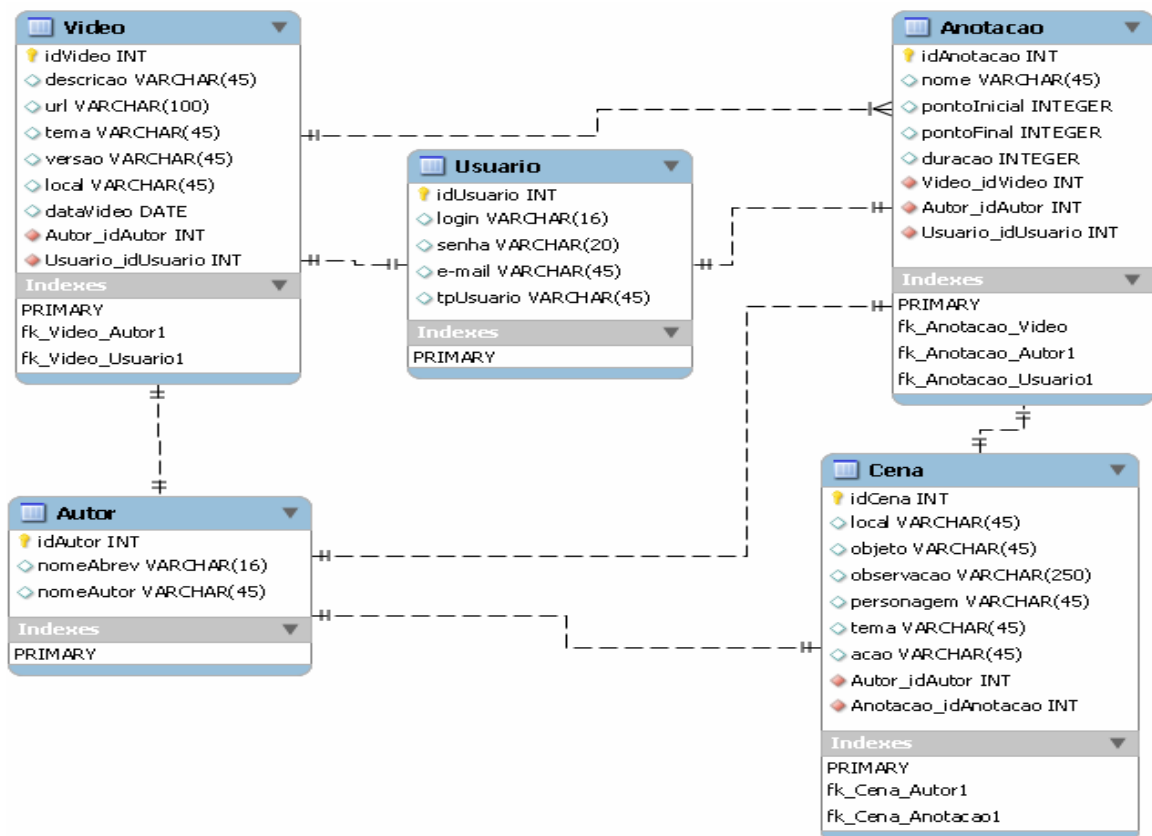


Figura 14 – Modelo Relacional Proposto.

Os dados a serem geridos pelo SGBD, consistem em:

- Vídeo: será armazenado dado do vídeo, tais como, identificação, descrição, tema. Estes dados são necessários para facilitar o mecanismo de busca da aplicação indexVideo.
- Autor: dados do autor do vídeo, da anotação e da cena .

□ Usuário: informações do usuário que realizou a operação, de *upload* do vídeo, da criação de uma anotação, e posteriormente para *login* da aplicação.

□ Anotação: armazena os dados das anotações, o que seria as tags do arquivo XML. Contém a informação temporal referente a um vídeo, sendo utilizada na recuperação do mesmo, pois a principal busca do *indexVideo* é por trechos dos vídeos, referente a um assunto especificado pelo usuário.

□ Cena: dados referentes a cena selecionada na anotação, sendo armazenados dados como local, objetos utilizados, observações, etc.

Com o modelo acima descrito podemos montar a estrutura da aplicação condizendo com a estrutura de dados a ser manipuladas e gerenciar melhor o SGBD a ser utilizado, assim dimensionando os recursos corretamente.

A camada de persistência de dados, que será adicionada na aplicação *indexVideo*, tem o objetivo de fazer a comunicação direta com SGBD. Nesta camada estarão previstas diretivas de consulta, inclusão, alteração e exclusão de dados contidos na base. Todos os dados necessários para suprir a aplicação *indexVideo* passarão por esta camada.

Conforme demonstrado na Figura 13, a concepção da camada de persistência dos dados em um SGBD será dividida em duas etapas:

□ Substituição do arquivo XML por SGBD: conforme descrito na seção 3.3.1. Nesta etapa será concedido a alteração da camada de persistência de dados e adaptado a estrutura já existente da aplicação *indexVideo*, prevendo a geração dos arquivos XML temporários para integrar com o MPEG-7, cuidando para não interferir no que já foi desenvolvido.

□ Testes: esta etapa será elaborada sobre uma massa de dados que possibilite avaliar o desempenho do SGBD e a aplicação *indexVideo*, verificando se não há nenhum tipo de falha na comunicação entre aplicação cliente e aplicação servidor, e validando a estrutura dos arquivos XML, verificando se está de acordo com o padrão especificado pela MPEG.

3.3.3 MUDANÇAS NO CÓDIGO FONTE

Será desenvolvida uma camada de persistência de dados que conterão os comandos de abertura de conexão com o SGBD, as instruções de inclusão, exclusão e alteração de dados na base. Também é responsável por entregar e recuperação dos dados para a camada de conversão que, quando necessário, montará o arquivo XML temporário, a fim de enviar a aplicação cliente para reprodução do vídeo juntamente com suas anotações e propriedades.

Na camada de conversão teremos os dados vindos da camada de persistência, com isso será montado o arquivo XML, compatível com a estrutura MPEG-7 e, depois de recuperado o vídeo em questão, será enviado à aplicação, para a sua reprodução.

Também será alterada a busca dos dados que passarão a ser feitas diretamente no SGBD e não mais em cada arquivo XML. Os arquivos XML passarão a ser documentos de integração entre plataforma e não mais uma estrutura que armazena as informações do indexVideo.

3.3.4 IMPACTOS NA APLICAÇÃO

Para o usuário final não haverá nenhum impacto visual, pois não se trata de alterações no nível de interface, porém, o mesmo terá mais segurança na informação armazenada nesta aplicação.

A aplicação sofrerá impactos nas diretivas de armazenamento e recuperação dos dados, os quais deixarão de ser armazenados em arquivos XML para serem armazenados em um SGBD, que possui uma série de ferramentas e políticas para garantir a integridade da informação nele depositada, visando à disponibilidade destes dados para pesquisa.

A pesquisa na base de dados será mais ágil, pois se está trabalhando com um SGBD que prevê estrutura de buscas de dados, agilizando o processo de recuperação de informação, deixando disponível ao usuário, em um tempo menor, tornando a aplicação mais eficiente, pois no momento da pesquisa destes dados não necessita recuperar um arquivo XML para suprir a aplicação. O arquivo XML só será necessário no momento da reprodução do vídeo, sendo que neste momento a nova camada adicionada na aplicação entrará em ação para montagem do arquivo XML, que seguirá juntamente com o vídeo para sua reprodução, e o usuário final terá o mesmo resultado desejado com mais eficiência e segurança, mesmo que isto esteja implícito para o usuário (POSTGRESSQL, 2010).

3.3.5 VANTAGENS E DESVANTAGENS

Analisando as alterações propostas em nível de armazenamento de informações, têm-se um grande ganho na qualidade dos dados, pois se está trabalhando com SGBD, com isso se

ganha em segurança na informação armazenada, melhoria na qualidade nas consultas realizadas pela aplicação.

No nível de aplicação, há uma melhora no desempenho na questão de extração das informações armazenadas no SGBD, e no tempo de resposta das requisições, quando submetido à solicitação de “N” usuários em “N” instâncias da aplicação indexVideo.

Já no nível de manutenção do sistema por parte da equipe técnica, ficará melhor organizado, pois o SGBD prevê rotinas de *back-up* da base de dados podendo ser agendados horários para esta execução, sem ter a intervenção humana, assim, garantindo um *back-up* íntegro da base de dados, caso um dia houvesse uma necessidade de uma restauração do *back-up*.

Com uma política de segurança e acesso a base de dados, a mesma estará segura, pois ela não ficará exposta a alterações indesejadas por pessoas não autorizadas ou que não possuam conhecimento sobre a ferramenta. Hoje, ter acesso ao repositório, fica fácil apagar um arquivo XML e perder todas as informações referentes a um ou vários vídeos se as informações estiverem em um SGBD não estarão dispostas a estas vulnerabilidades.

Com a utilização de um SGBD para armazenar as informações da aplicação indexVideo será necessário criar uma camada de conversão de dados para suprir a estrutura necessária para funcionar o MPEG-7, que seria extrair os dados do SGBD e transformá-los em arquivo XML, e enviar ao reproduzidor de vídeo, acarretando um atraso na entrega do vídeo para a reprodução devido a conversão necessária.

3.4 CONSIDERAÇÕES FINAIS

As alterações propostas, para a aplicação indexVideo tem como principal propósito um melhor desempenho do mesmo e a confiabilidade dos usuários junto a aplicação. Conforme manual de instalação contido no anexo “A” o mesmo auxiliará no momento de instalar a aplicação e fazer as devidas manutenções.

No nível de manutenção dos dados da aplicação indexVideo, sendo armazenado em SGBD temos maior segurança nas operações a serem executadas, pois existe procedimentos padronizados, para *back-up* e restauração.

A indexVideo será testada, e somente após a certificação de que as atividades propostas pela aplicação estejam perfeitamente funcionando, será liberada para utilização em massa.

4 ALTERAÇÕES REALIZADAS NA APLICAÇÃO

Para tornarmos a aplicação indexVideo operante com SGBD, faz-se necessário criar uma camada de persistência de dados. Como consequência a esta implementação foi alterado o fluxo original da aplicação, assim prevendo todas as alterações necessárias para manter a compatibilidade da aplicação com as tecnologias adotadas em seu projeto inicial.

Em seu projeto inicial previu-se a implementação da aplicação indexVideo sob o padrão MPEG-7 integrada com XML para armazenamento das informações do vídeo. Com isso criamos métodos para manipulação de arquivos XML temporários, gerando uma integração entre SGBD e MPEG-7

4.1 FLUXO DE INFORMAÇÃO

Substituindo a estrutura de armazenamento dos dados da aplicação IndexVideo, de arquivos XML para SGBD é necessário alterar o fluxo das informações percorridas entre as camadas da aplicação conforme demonstrado na Figura 15.

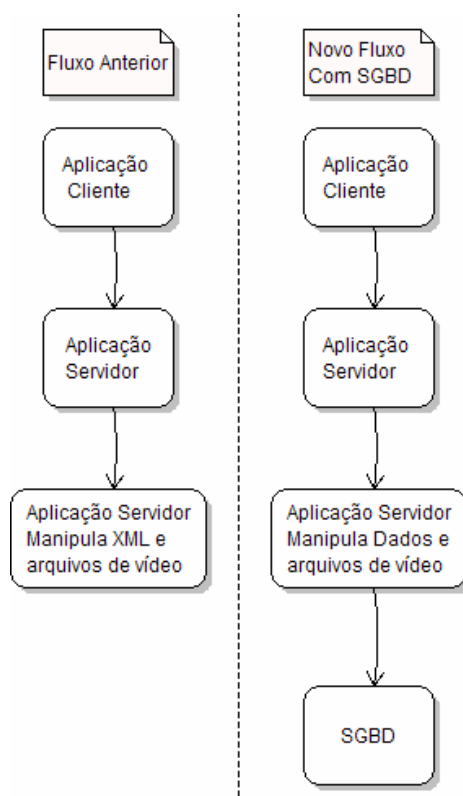


Figura 15 – Fluxo da Aplicação indexVideo.

O fluxo original da aplicação indexVideo é: aplicação cliente comunicando-se com a aplicação servidor, e a aplicação servidor responsável pelo armazenamento de dados em arquivos XML: salvar o arquivo de vídeo em um repositório especificado nas configurações da aplicação. Com a substituição da forma de armazenamento, acrescentou-se o SGBD, onde a aplicação cliente continuará comunicando diretamente com aplicação servidor, a qual suprirá as informações do cliente no mesmo padrão necessário para rodar sob o padrão MPEG-7, porém a aplicação servidora não gerenciará mais arquivos XML com as informações e sim, enviará ao SGBD as informações para salvar e habilitar para posteriores consultas.

As consultas feitas pela aplicação cliente serão solicitadas à aplicação servidor que não irá mais pesquisar nos arquivos XML e sim, em um SGBD, através de instruções SQL, agilizando o processo de pesquisa e recuperação dos dados.

A recuperação de um vídeo ou de uma anotação pela aplicação sofrerá uma consulta na base de dados e recuperará as informações necessárias para montar o arquivo XML temporário correspondente ao vídeo em questão, com as anotações e os dados do vídeo, montado em estrutura compatível com MPEG-7, que é enviado à aplicação cliente, juntamente com o vídeo e o arquivo XML temporários, para sua reprodução.

A reprodução do vídeo associado a anotação correspondente, na aplicação cliente, continua sendo executada na forma original que foi concebida; as alterações sofridas ocorrem no nível da aplicação servidor.

Na aplicação servidor foi adicionado uma camada de persistência de dados, a comunicação com SGBD, e alterado a manipulação das estruturas XML, a camada de "IO" dos vídeos e as suas anotações. Adicionado no nível de interface campos para instruções de comunicação com o SGBD e os repositórios de vídeo e anotações temporárias. As camadas adicionadas está seguindo o mesmo padrão de projeto adotado na concepção da aplicação indexVideo.

Para esta aplicação utilizamos o SGBD Postgressql, na versão 8.4.3.1, o qual possui boa performance para execução das instruções necessárias para manter os dados da aplicação indexVideo e possui todas as diretivas de back-up e proteção contra a falhas conforme descritos no anexo B.

4.2 MODELO RELACIONAL ADOTADO

Na seção 3.3.2 sugeriu-se um modelo relacional a ser adotado nesta aplicação, o qual foi montado com base na análise feita sob a estrutura do XML gerado pela aplicação, no formato padrão MPEG-7, porém não foi feita a análise do modelo relacional proposto na interface da aplicação indexVideo. Para seguirmos o modelo relacional inicialmente proposto teríamos que alterar a interface com o usuário na aplicação cliente, com isso verificamos a necessidade de alterarmos o modelo relacional adotado para aplicação indexVideo o qual está demonstrado da Figura 16.

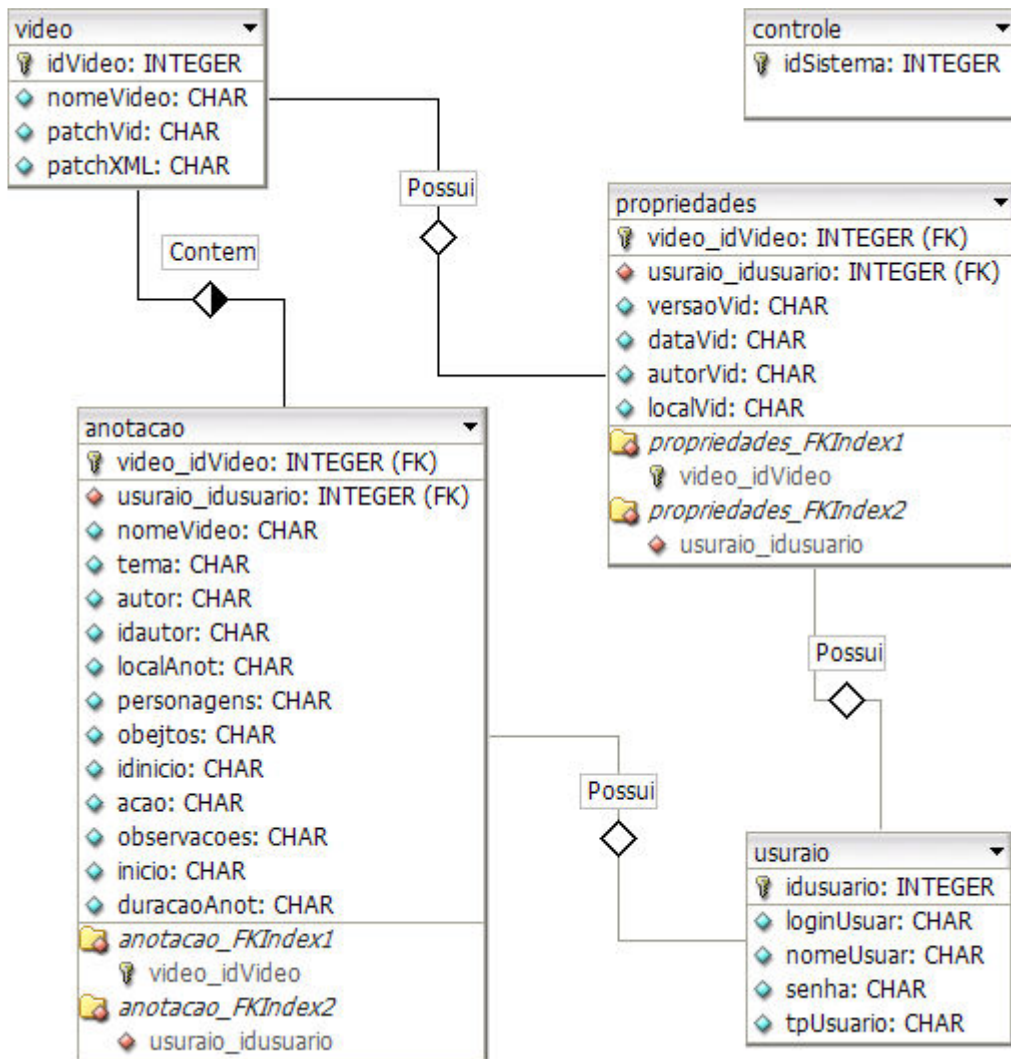


Figura 16 – Modelo Relacional Implementado.

A alteração ocorreu devido a aplicação cliente não possuir telas de cadastro específicas, tais como cena e autor. Para ser separado em um cadastro de cenas e autor, a aplicação cliente terá que sofrer alterações na sua interface, e neste caso ficará como sugestão de melhoria de produto, juntamente com o cadastro de usuários e suas devidas validações.

No modelo relacional adotado estão previsto os dados dos vídeos e suas propriedades, as anotações correspondentes e o cadastro de usuários, porém suas permissões não estão previstas, estando distribuídas estas informações nas seguintes tabelas:

- vídeo: tabela responsável por armazenar dados como nome do vídeo, caminho que o vídeo está armazenado e o caminho que será utilizado para geração do arquivo XML temporário, correspondente ao vídeo selecionado, obedecendo ao padrão MPEG-7.

- propriedades: contém as especificações do vídeo, o usuário que o postou, versão, data, sendo armazenado no formato padrão do MPEG-7, autor do vídeo e o local em que foi realizado.

- anotação: possui todas as anotações do vídeo, associado através de uma chave de identificação com a tabela vídeo, contendo os seguintes dados: usuário que implantou a anotação, nome do vídeo, tema da anotação, autor e sua identificação, o local que ocorre o trecho do vídeo selecionado, ação, personagens, objetos e observações da cena, o identificador de início (campo necessário para gerar o XML temporário no padrão MPEG-7), unidade temporal de início da anotação no formato padrão MPEG-7 e a duração da anotação.

- usuário: tabela disposta a armazenar os dados dos usuários da aplicação, possuindo um identificador, login, nome do usuário, senha e tipo de usuário. Esta tabela está prevista, porém, atualmente a aplicação não a utiliza, pois é necessário implementar as políticas de segurança e acesso a aplicação indexVideo.

- controle: foi criada esta tabela com o intuito de validar a disponibilidade dos dados e versão da aplicação, também poderá ser utilizada em implementações futuras como tabela de parâmetros necessários para alguma funcionalidade desenvolvida a esta aplicação, a mesma também não influencia diretamente nos dados distribuídos na aplicação, somente utilizada no teste de disponibilidade do SGBD.

4.3 PACOTES DA APLICAÇÃO

Adicionados dois pacotes na aplicação indexVideo, os quais competem da camada de persistência de dados, contendo as classes de conexão com SGBD, da persistência de dados e da recuperação dos dados, e alterado a camada de IO da aplicação, conforme Figura 17.

As alterações feitas foram no nível de criar um montador de XML compatível com o padrão MPEG-7, e a inibição da geração dos arquivos XML no momento que era salvo o vídeo no repositório.

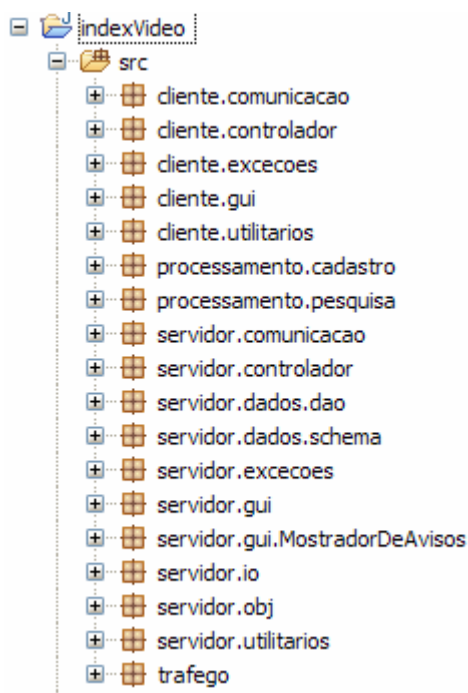


Figura 17 – Pacotes da aplicação indexVideo

4.4 CAMADA DE PERSISTÊNCIA

Para os dados da aplicação serem armazenados em um SGBD adicionou-se à aplicação servidor uma camada de persistência de dados e comunicação com o SGBD e foi distribuída em dois pacotes, conforme Figura 18:



Figura 18 – Pacotes adicionados à aplicação indexVideo

□ servidor.dados.dao: este pacote contém todas as classes necessárias para a conexão com o SGBD e execução das instruções SQL, as quais estão descritas no pacote schema, conforme demonstrado na Figura 19.

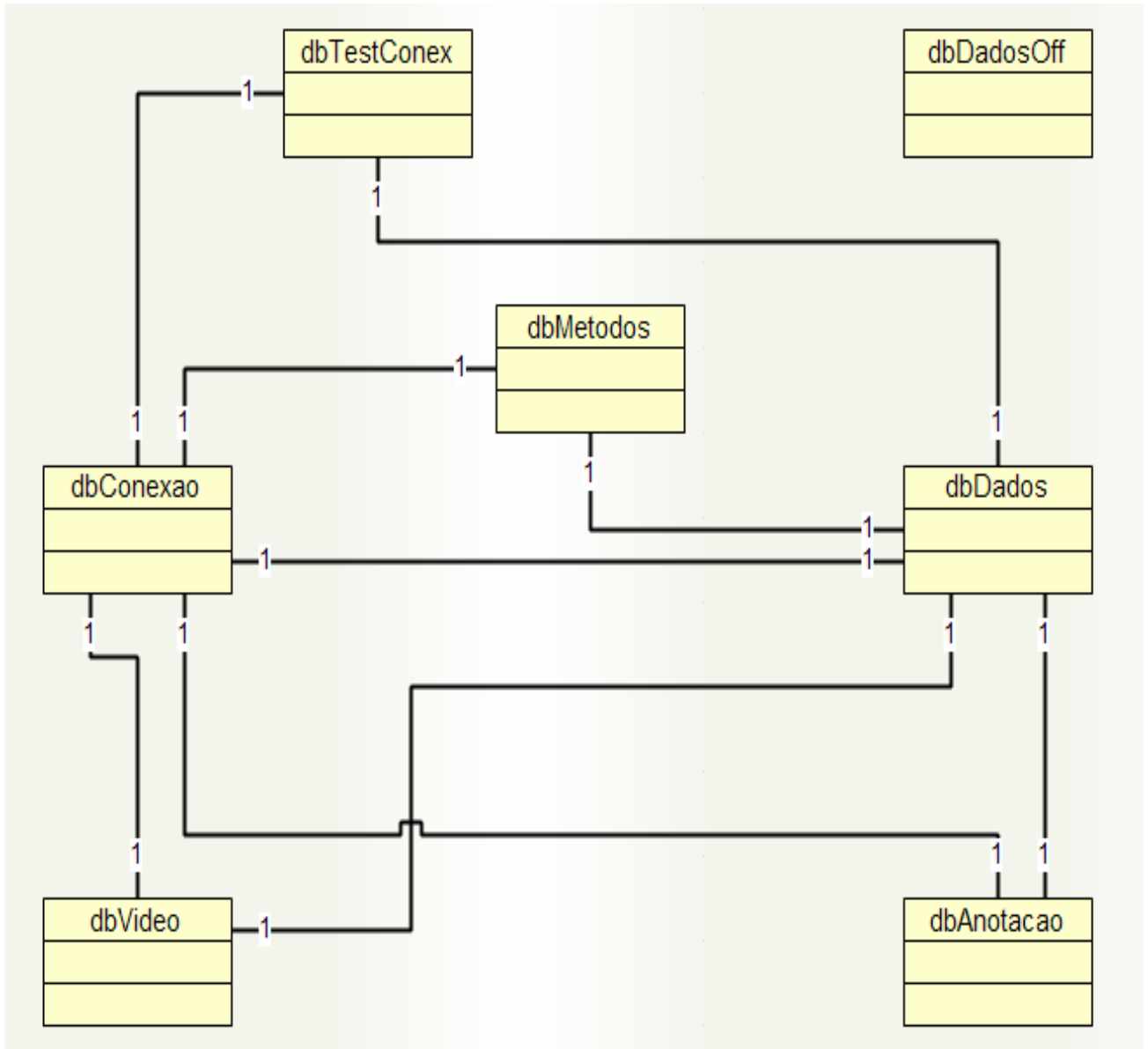


Figura 19 – Classes do pacote “DAO”

□ servidor.dados.schema: este pacote possui as instruções SQL, que são utilizadas para manipulação dos dados no SGBD, para inserção, extração e alteração dos dados da aplicação, contendo as classes conforme demonstrado na Figura 20 .

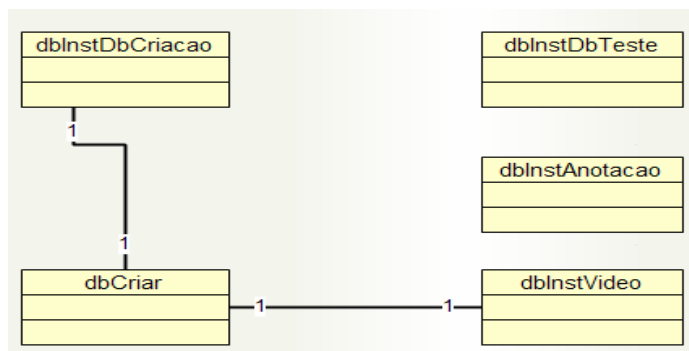


Figura 20 – Classes do Pacote “SCHEMA”

O pacote servidor.dados.dao, responsável pela execução das instruções SQL e comunicação com o SGBD, com as validações necessárias para rodar a aplicação, conforme dispostos nas classes abaixo é demonstrada no diagrama de classes da Figura 21 :

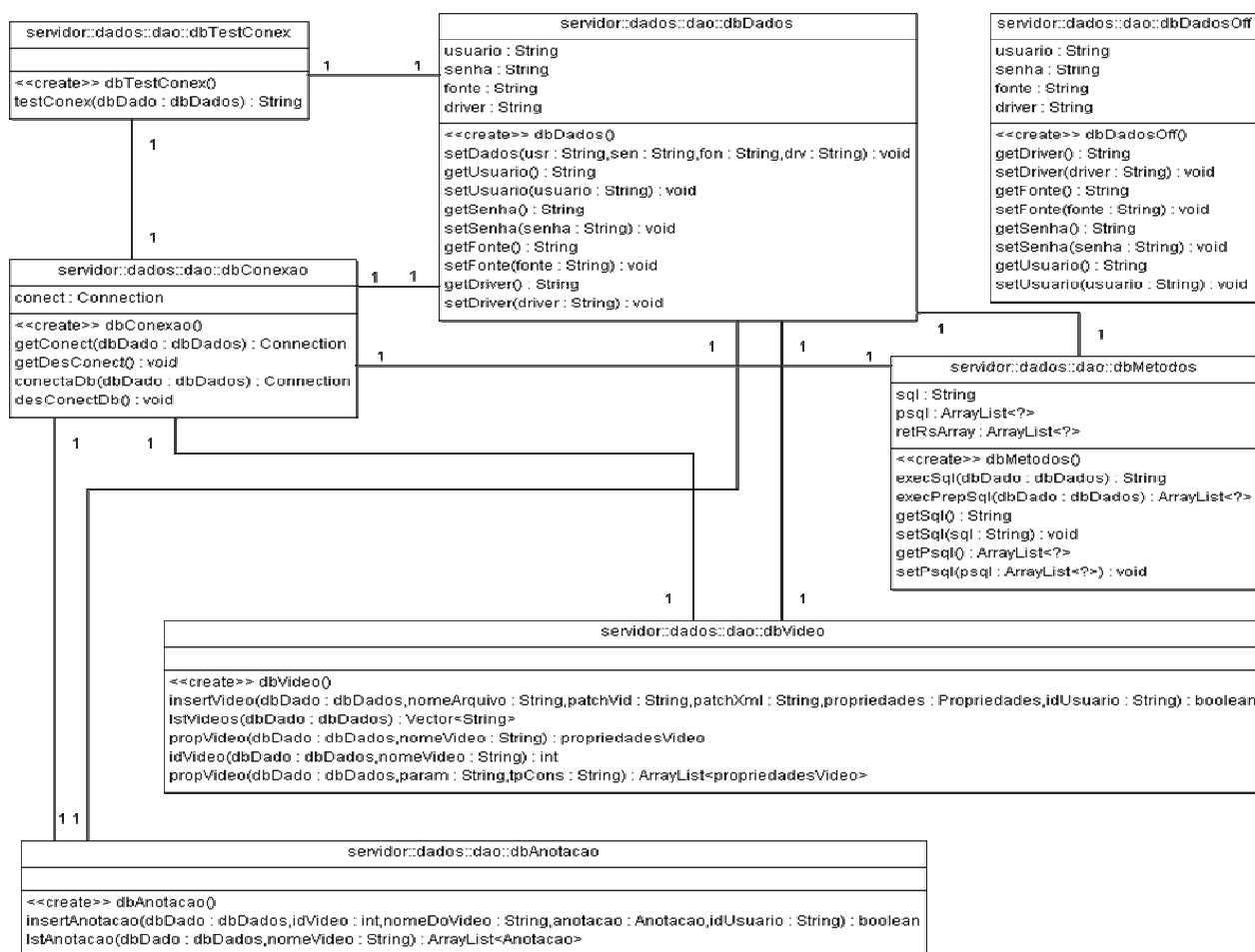


Figura 21 – Diagrama de Classes – Pacote “DAO”

- dbAntacao.java: manipulação dos dados na tabela de anotações dos vídeo.
- dbConexao.java: classe responsável por estabelecer a conexão com SGBD em

questão a ser utilizado .

- dbDados.java: objeto que contém as informações necessárias para conexão com o SGBD, as quais são carregadas na inicialização da aplicação servidor, estes dados vem do arquivo de configuração feitas pelo usuário.
- dbDadosOff.java: objeto que contém informações para conexão com o SGBD, utilizada pela classe dbCriar.java, responsável pela criação da estrutura de dados da aplicação indexVideo.
- dbMetodos.java: responsável pela preparação das instruções SQL, antes de ser executada, pela classe dbCriar.java.
- dbTestConex.java: responsável por testar se a comunicação com o SGBD está respondendo sem falhas e verificar a versão do programa.
- dbVideo.java: responsável pela manipulação dos dados da tabela de vídeo e propriedades do vídeo, criação, extração, alteração e exclusão das informações referente aos videos.

As classes adicionadas no pacote servidor.dados.schema, corresponde as instruções SQL para execução de procedimentos em um SGBD, conforme discriminado nas classes abaixo e demonstrado do diagrama de classes na Figura 22 :

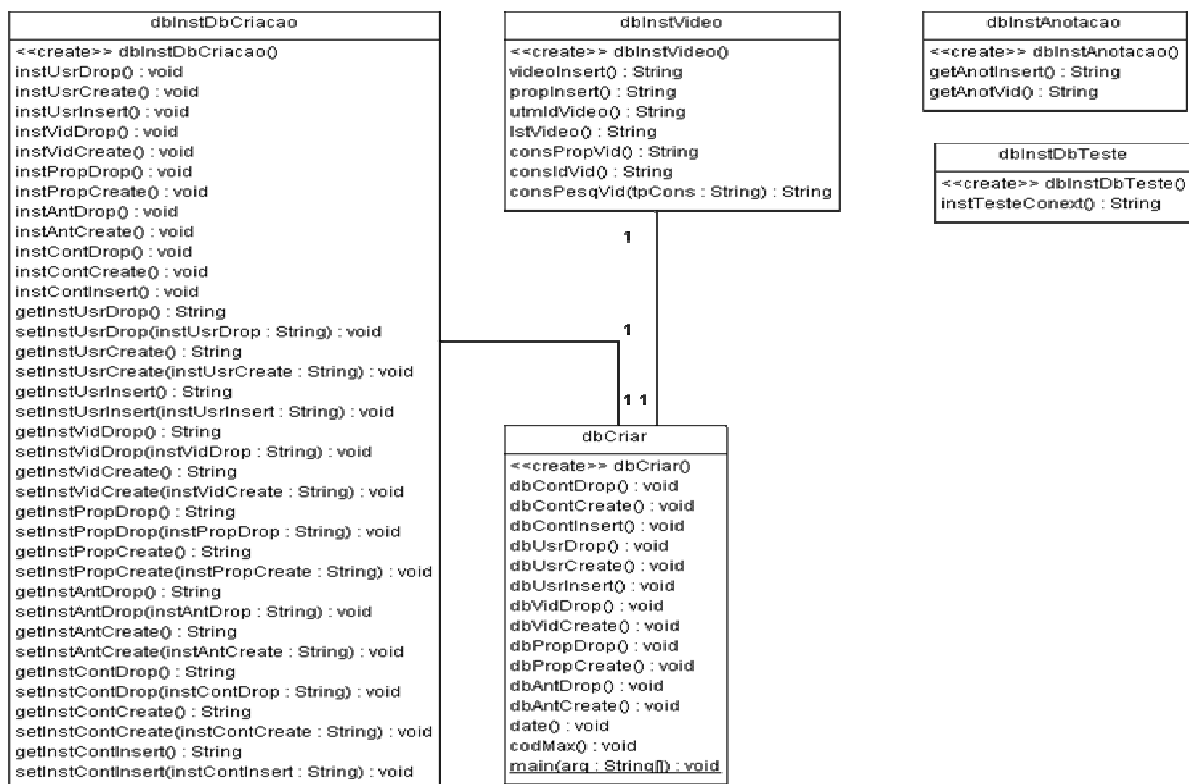


Figura 22 – Diagrama de Classes – Pacote “SCHEMA”

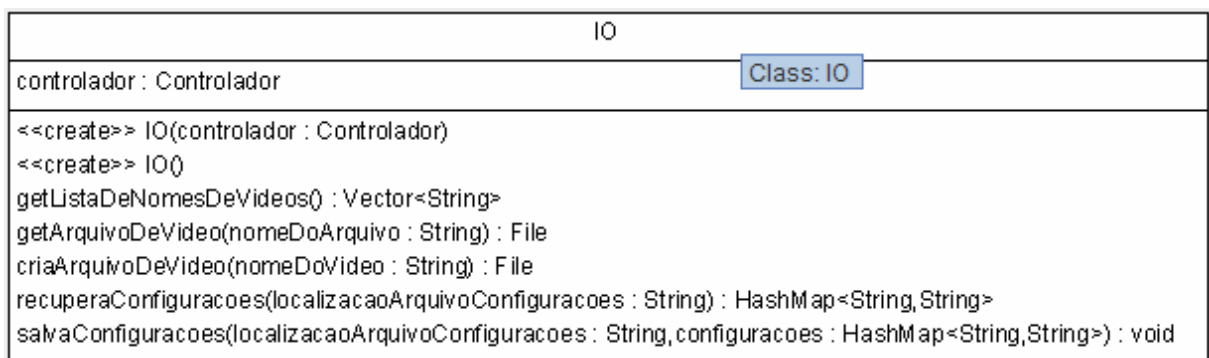
- dbCriar.java: classe responsável pela criação da estrutura de dados necessária para comportar as informações da aplicação indexVideo.
- dbInstAnotacao.java: possui as instruções SQL referente a tabela do SGBD correspondente a Anotações.
- dbInstDbCriacao.java: corresponde as instruções de criação da estrutura de dados necessários para suprir as informações da aplicação indexVideo.
- dbInstDbTeste.java: possui instrução de teste de comunicação com o SGBD.
- dbInstVideo.java: responsável por todas as instruções SQL de manipulação dos dados correspondente ao cabeçalho do arquivo XML, tais como autor, origem, data e etc.

Estes dois pacotes são responsáveis por inserir, alterar, excluir e consultar os dados da aplicação indexVideo, sendo utilizado uma camada de transporte de objetos das propriedades do vídeo, a qual está no pacote servidor.obj, que possui o objeto propriedadesVideo contendo todos os dados necessários para sua manipulação .

4.5 CAMADA IO

A fim de habilitar a camada de persistência de dados, se alterou a camada de “IO” da aplicação servidor, desabilitando os métodos de criação dos arquivos XML, consulta em arquivos XML e criado a montagem de estruturas XML temporárias no padrão MPEG-7.

As alterações sofridas no pacote seridor.io nas classes ManipulaXML e IO, são mostrados na Figura 23.



ManipulaXML
controlador: Controlador
<pre> <<create>> ManipulaXML(controlador: Controlador) removerArquivoDeDescricaoDeVideo(nomeDoVideo: String): void getPropriedades(MPEG7: Element): Propriedades getAnotacoesDeVideo(nomeDoVideo: String): ArrayList<Anotacao> getLocalizacaoVideo(arquivoXMLDescricaoDoVideo: File): File getLocalizacaoVideo(raiz: Element): File alteraLocalizacaoDeVideoNaDescricao(arquivoXMLDescricaoDoVideo: File, localizacaoVideo: File): void getArquivosRepositorio(): File[] removeAnotacoesEmArquivoDeDescricaoDeVideo(nomeDoVideo: String, anotacao: Anotacao): void converteData(data: String): String converteDataAbrir(dataHora: String): String converteDuracaoAbrir(duracao: String): String converteInicioAbrir(inicio: String): String converteDuracao(duracao: String): String converteInicio(inicio: String): String gravaArquivo(xml: String, arquivo: File): void getNomeDoArquivoDeVideo(arquivoXMLDeDescricao: File): String getElementoXMLRaiz(arquivoDescricao: File): Element pesquisaCampoPropriedadesEmArquivoDescricao(expressaoBuscada: String, campoABuscar: TipoDeCampo, MPEG7: Element, arquivoDescricao: File): ColecaoResultadosPorCampo pesquisaCampoDescricaoEmArquivoDescricao(expressaoBuscada: String, campoABuscar: TipoDeCampo, MPEG7: Element): ColecaoResultadosPorCampo getTodasAnotacoesDoArquivoDeDescricao(arquivoDescricao: File): ArrayList<Anotacao> getAnotacaoDeSegmento(segmento: Element): Anotacao pegaDescricaoTema(elemento: Element): String pegaDescricaoAutor(elemento: Element): String pegaDescricaoDusuario(elemento: Element): String pegaDescricaoLocal(elemento: Element): String pegaDescricaoPersonagens(elemento: Element): String pegaDescricaoObjetos(elemento: Element): String pegaDescricaoAcao(elemento: Element): String pegaDescricaoObservacoes(elemento: Element): String pegaDescricaoInicio(elemento: Element): String pegaDescricaoDuracao(elemento: Element): String getPropriedadesDeVideo(nomeDoVideo: String): Propriedades getArquivoDescricaoDeVideo(nomeDoVideo: String): File gravaCabecalhoNovoVideo(propriedades: Propriedades, arquivoDeVideo: File, idUsuario: String): void insereAnotacoesEmArquivoDeDescricaoDeVideo(nomeDoVideo: String, anotacao: Anotacao, idUsuario: String): void getTodasAnotacoesDoArquivoDeDescricao(nomeArquivoDeVideo: String): ArrayList<Anotacao> montCabXML(nomeVideo: String): String anoVideos(nomeVideo: String): ArrayList<Anotacao> montAnoXML(nomeVideo: String): String montRoadXML(): String converteAnotacaoEmXML(anotacao: Anotacao): String pesquisaEmRepositorio(expressaoBuscada: String, camposABuscar: TipoDeCampo[]): ResultadoPesquisa </pre>

Figura 23 – Classes Alteradas – Pacote “IO”

Na classe IO.java foi alterado os métodos abaixo descritos :

- salvaConfiguracoes e recuperaConfiguracoes: alterado para salvar as especificações dos dados para conexão com SGBD, e carregar o objeto dbDados que contém as informações de conexão com o SGBD.

- getListaDeNomesDeVideos: alterado para consultar na camada de persistência de dados a lista de vídeo disponíveis.

As alteração sofridas na classe ManipulaXML estão abaixo descritas, afim de eliminar a criação e manutenção de arquivos XML referente aos vídeos, e manipular estes dados juntos ao SGBD :

- gravaCabecalhoNovoVideo: no momento a inserção de um novo vídeo na aplicação serão criados os registros das informações nas tabelas e logo após será salvo o arquivo de vídeo no repositório especificado nas configurações da aplicação e não mais criado o arquivo XML.

- insereAnotacoesEmArquivoDeDescricaoDeVideo: alterado para transmitir os dados anteriormente salvo no arquivo XML correspondente para a camada de persistência de dados que verifica se as informações estão íntegras para serem postadas no SGBD.

- getPropriedadesDeVideo: este método solicita à camada de persistência de dados todas as informações do vídeo, para ser transmitido a aplicação cliente, eliminando a procura destas informações em todos arquivos XML dispostos no repositório de anotações.

- getTodasAnotacoesDoArquivoDeDescricao: solicita à camada de persistência de dados todas as anotações referentes a um vídeo passado por parâmetro pela aplicação cliente.

- pesquisaEmRepositorio: solicita à camada de persistência de dados uma lista de todos os vídeo e anotações conforme parâmetros passados .

- anotVideos: método criado para solicitar a camada de persistência de dados que retorna um *arraylist* com todas as anotações de um vídeo.

- montAnotXML: método criado para montar a estrutura temporária do XML referente a anotação do vídeo solicitado.

- montCabXML: método criado para montar o cabeçalho da estrutura temporária do XML.

- montRoadXML: método criado para montar o rodapé do XML.

A fim de possibilitarmos ao usuário da aplicação servidor, configurar as informações de conexão com SGBD e o repositório onde será salvo os vídeos e a criação das anotações temporárias, se alterou a classe `JanelaConfiguracoes` dentro do pacote `servidor.gui`.

Adicionou-se os campos usuário, senha, fonte e string de conexão com o SGBD e a funcionalidade de teste de conexão, que valida as informações configuradas nos campos acima mencionados, sofrendo alterações no pacote `servidor.controlador`.

O pacote `servidor.controlador` possui a classe `controlador` que gerencia todas as tarefas executadas pela aplicação servidor, onde dentro do contexto adicionou-se a manipulação do arquivo de configuração da aplicação, prevendo os campos adicionados na interface de configuração com o usuário.

No controlador foram criados os métodos para carregar as informações no objeto `dbDados` sendo deste objeto que parte os dados para conexão com o SGBD.

4.6 ALTERAÇÕES DE INTEGRAÇÃO - JVLC

Para deixar a aplicação `indexVideo` operante com a versão do *java runtime environment 1.6*, necessitou-se atualizar as bibliotecas de reprodução de vídeo (JVLC), refletindo que todas as chamadas desta biblioteca fossem revistas e alterado a ordem dos parâmetros de chamada para execução, alteração feita no nível de aplicação cliente e aplicação servidor. Assim possibilitando manter o padrão de reprodução de vídeo já adotado na aplicação, e não trazendo maiores prejuízos à mesma, tais como perda de performance ou até mesmo inconsistência nas informações a serem geridas.

4.7 CONFIGURAÇÃO DA APLICAÇÃO

Configurou-se a aplicação `indexVideo` para trabalhar com SGBD e atualizar os arquivos do JVLC. Existindo configurações específicas para aplicação cliente e aplicação servidor, informar o caminhamento dos repositórios e os dados do SGBD, tais como local, usuário e senha, habilitando a aplicação servidor se comunicar com o SGBD configurado, e possibilitando o teste de comunicação entre aplicação servidor e SGBD.

4.7.1 CONFIGURAÇÃO DA APLICAÇÃO SERVIDOR

Ao rodar a aplicação servidor, encontra-se no menu superior, configurações, configurações do servidor, a interface de configuração e se faz necessário informar os dados, conforme Figura 24:

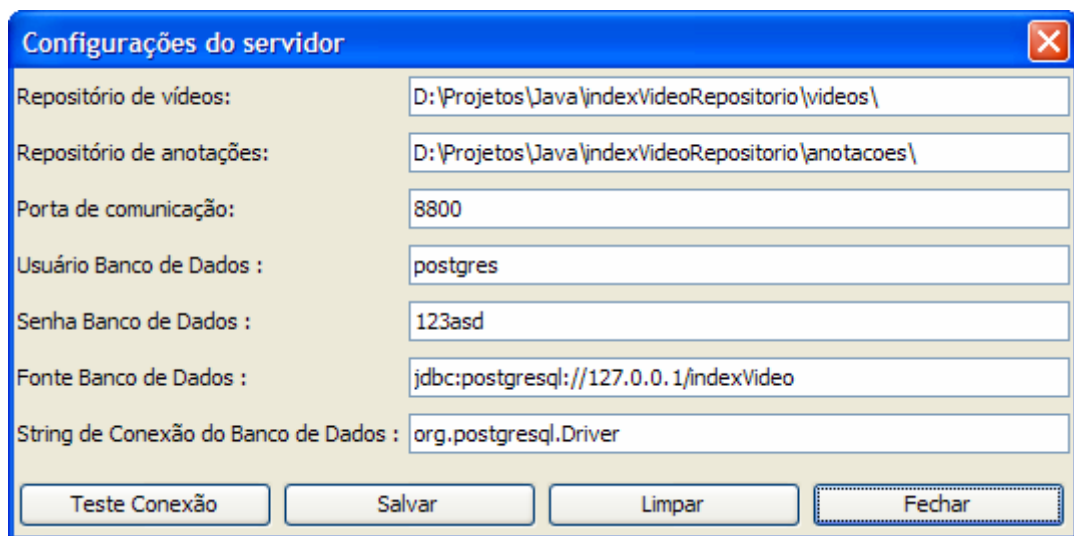


Figura 24 – Configuração Aplicação Servidor

- Repositório de Vídeos: o repositório indicado neste campo se refere ao local onde serão armazenado os vídeos da aplicação indexVideo.
- Repositório de Anotações: caminho que servirá como repositório para geração dos arquivos XML temporários, afim de enviar a aplicação cliente para reprodução do vídeo vinculado com as anotações.
- Porta de Comunicação: é a porta de comunicação que irá rodar a aplicação indexVideo, verificar se a porta informada não está sendo utilizada por outro programa.
- Usuário Banco de Dados: informar o usuário criado no SGBD, afim de utilizar na aplicação indexVideo para logar no mesmo e abrir comunicação.
- Senha Banco de Dados: senha do usuário do SGBD.
- Fonte Banco de Dados: caminho utilizado para conexão do SGBD.

□ String de Conexão do Banco de Dados: refere-se ao driver de comunicação, o mesmo nome do JAR adicionado na pasta LIB na raiz da aplicação.

□ Teste Conexão: ao clicar neste botão será verificado se os dados acima informados estão corretos, e a base de dados está disponível.

Após configurada a aplicação servidor se faz necessário instalar as bibliotecas JVLC, que se dá através do instalador “indexador.exe” localizado na raiz do programa, o qual possui um manual no anexo A mostrando o passo a passo de instalação .

4.7.2 CONFIGURAÇÃO DA APLICAÇÃO CLIENTE

Para configurar a aplicação cliente é necessário editar o arquivo “configuracoesCliente.txt” encontrado na raiz da aplicação e informar o *IP* do servidor a ser conectado e a porta de comunicação, a mesma porta informada na configuração da aplicação servidor, conforme demonstrado na Figura 25 , assim habilitado a aplicação cliente abrirá um canal de comunicação com a aplicação servidor .

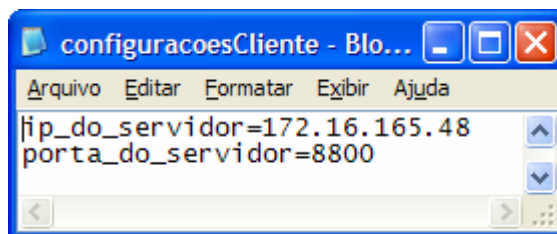


Figura 25 – Configuração Aplicação Cliente

4.8 MELHORIAS EXECUTADAS NA APLICAÇÃO

A aplicação indexVideo sofreu alterações na camada de persistência de dados e na geração dos arquivos XML no padrão MPEG-7, afim de armazenar os dados em um SGBD, com isso, melhoramos o desempenho e segurança da aplicação.

Agilizou-se o processo de consulta aos dados tanto das anotações, quanto dos vídeos, pois se deixou de pesquisar em arquivos XML, um a um, a informação em questão, e passou-se a pesquisar em uma massa de dados disposta em um SGBD que prevê rotinas de pesquisa

de dados, sendo o tempo de resposta ao usuário final diminuído significativamente.

Garantiu-se também a segurança das informações armazenadas no SGBD, através da política de segurança implementada no próprio SGBD, e a própria estrutura de *back-up*.

4.9 CONSIDERAÇÕES FINAIS

Com a elaboração das alterações desenvolvidas nesta aplicação, percebeu-se uma melhora significativa nas execuções das rotinas propostas nesta aplicação, tornando a mesma mais estável.

Eliminado erros de pesquisa de dados, a rotina antiga de pesquisa dos dados nos arquivos XML travava pois a mesma tinha necessidade de abrir arquivo a arquivo XML para consultar se a informação a ser selecionada estaria no corpo do arquivo XML em questão, e depois retornando a aplicação cliente as informações pesquisadas.

Com a implementação da camada de persistência de dados eliminou-se a rotina acima descrita, retirando este problema da aplicação, agilizando o processo de consulta.

A aplicação indexVideo operando com SGBD se tornou mais estável e segura, pois além de eliminarmos algumas rotinas que estariam travando a aplicação, incrementamos novas funcionalidades de segurança já prevista em qualquer SGBD, como restrição ao acesso aos dados da aplicação e as rotinas de *back-up*.

Afim de tornar a aplicação cliente mais amigável, fica como sugestão a re-estruturação da interface gráfica, prevendo uma identidade visual mais amigável e intuitiva ao usuário e, avaliar a atualização do componente JVLC, eliminando o travamento da aplicação cliente, adicionando login de usuário e políticas de acesso à aplicação.

Na aplicação servidor, implementar as rotinas de segurança de usuário, suprimindo as informações para aplicação cliente, melhorar a rotina de envio de vídeo e anotação para aplicação cliente afim de agilizar o processo de transmissão assim disponibilizando a aplicação cliente em um tempo menor.

No SGBD é aconselhável implementar as rotinas de *back-up* da base de dados, no mínimo uma vez ao dia, ou até mesmo a replicação da base de dados em outro servidor. Para a estruturação destas rotinas cabe uma análise do ambiente que a aplicação está rodando.

5 TESTES DA APLICAÇÃO

Com a finalidade de validar as transações executadas pela aplicação indexVideo integrada com SGBD postgres, utilizou-se dois cenários para reprodução dos teste e validação da transcrição dos dados entre as aplicações cliente, aplicação servidor e SGBD.

No cenário “A”, utilizando dois microcomputadores, num deles está rodando a aplicação servidor e o SGBD, e no outro está rodando a aplicação cliente, sustentados sob a mesma configuração de rede, onde os mesmos possuem liberação e acesso total ao seus compartilhamentos e portas de comunicação, conforme demonstrado na Figura 26 .



Figura 26 – Teste Cenário A.

No cenário B, utilizamos a rede da UCS (Universidade de Caxias do Sul), sendo um microcomputador conhecido rodando a aplicação servidor, um servidor desconhecido rodando o SGBD, previamente instalado na UCS para utilização de base teste dos alunos, e dois microcomputadores rodando a aplicação cliente, conforme demonstrado na Figura 27, estando todos sob a mesma configuração de rede .



Figura 27 – Teste Cenário B.

Sob estes dois cenários foi elaborado teste de funcionalidades, persistência, geração de dados, consultas na aplicação indexVideo, a qual possui resultados aceitáveis de trabalho.

Os testes avaliaram acesso a base de dados, geração de XML, reprodução de vídeo, integração entre XML e SGBD e consulta das informações salvas na base de dados.

5.1 ACESSO A BASE DE DADOS

Validado a disponibilidade da base de dados enquanto a aplicação indexVideo rodava em multi-seção, tanto em aplicação cliente e aplicação servidor, obtendo a disponibilidade integral do SGBD, este respondeu a todas as requisições da aplicação, não havendo atraso nas respostas e nem perda de informações.

A validação da consistência das informações se deu através da verificação dos dados imputados na aplicação cliente que foi inserido nos locais corretos na base de dados. Consultando direto na base de dados verificou-se que a informação atribuída na aplicação cliente chegou corretamente no SGBD e foi adicionado na base de dados de forma consistida e correta, nos locais corretos.

Toda a validação dos dados se deu através de uma consulta direta na base de dados, utilizando uma console pré-instalada com o SGBD, assim habilitando a verificação das informações nela imputadas, se estão corretas em seus devidos lugares ou não.

A comparação entre as informações imputadas e os seus devidos lugares, se deu através da verificação do modelo relacional adotado, conforme descrito na Seção 4.2 e demonstrado na Figura 16 e a informação propriamente imputada na aplicação cliente.

Neste contexto podemos afirmar que o acesso à base de dados e os dados estão íntegros e disponíveis em tempo integral, para consulta na aplicação cliente, e na console do SGBD utilizado, ou até mesmo para outras aplicações que possua acesso e liberação a esta base.

5.2 GERAÇÃO DE XML

Quando a aplicação cliente solicita à aplicação servidor um vídeo de seu repositório, a aplicação servidor envia à aplicação solicitante o vídeo que está em seu repositório e um

arquivo XML contendo as informações para sua reprodução tais como autor, unidades temporais e etc, no padrão MPEG-7. Para que sua reprodução seja perfeita, o arquivo XML tem que estar gerado corretamente.

A geração do arquivo XML passou por uma avaliação comparativa de estrutura. No momento que aplicação servidor gera um arquivo XML temporário a fim de enviar a aplicação cliente, neste momento extraiu-se o arquivo XML temporário e comparou-se com um arquivo XML gerado na versão antiga da aplicação indexVideo com os mesmos dados e, verificou a compatibilidade perfeita entre os dois arquivos.

Assim validamos entre as duas estruturas, comparando as amostras por semelhança, garantindo que a aplicação estará gerando o XML temporário no padrão MPEG-7, para isso mapeamos o arquivo temporário e validamos junto ao padrão da estrutura solicitada, pela MPEG e verificamos a compatibilidade total da estrutura gerada como padrão.

5.3 REPRODUÇÃO DE VÍDEO

A fim de verificarmos o correto funcionamento da reprodução do vídeo com suas anotações, através da recuperação dos dados do repositório, imputados três vídeos inicialmente sem anotações, recuperamos um vídeo por vez, e adicionamos anotações aos vídeos, com isso possuímos informações suficientes para o teste de reprodução dos vídeos.

Aos vídeos recuperados pela primeira vez retornaram todos os dados e conseguimos reproduzir normalmente estas informações, porém a partir de uma certa repetição de recuperação dos vídeos a interface gráfica da aplicação cliente deixa de se atualizar e gera travamento da mesma.

A fim de diagnosticarmos o problema, verificamos se o que estaria travando seria a recuperação das informações, porém as informações estavam chegando corretamente para aplicação cliente, com os dados regidos no padrão MPEG-7 constatamos que o problema era na atualização dos componentes da interface gráfica.

Devido a interface gráfica estar como sugestão de melhoria da aplicação cliente não foi adequado este erro, sendo assim a reprodução dos vídeos pode ser prejudicada levando ao usuário ter que reiniciar a aplicação para voltar a funcionar.

5.4 INTEGRAÇÃO ENTRE XML E SGBD

Como a aplicação cliente exige um XML compatível com o MPEG-7 para a reprodução do vídeo verificamos a montagem do XML em tempo de execução da aplicação, extraindo os dados necessários do SGBD para a confecção deste arquivo.

A sua montagem em tempo real é relativamente rápida, pois as informações já estão selecionadas devido às instruções executadas pela aplicação servidor junto ao SGBD.

Com o intuito de executarmos o teste de geração do XML e verificar se as informações contidas nele estão exatamente igual a do SGBD, cria-se em disco um arquivo XML temporário e envia para a mesma e no momento seguinte apaga o arquivo; isso garante que não haverá dados incorretos, e que a informação contida no SGBD seja a mesma que o usuário esteja vendo na aplicação cliente.

5.5 CONSULTAS

A aplicação cliente possui uma interface específica para consultas a qual foi desenvolvida com uma série de filtros e parâmetros, a fim de auxiliar no resultado das operações e retornar ao usuário as informações necessárias para seu conhecimento.

A fim da aplicação cliente demonstrar ao usuário o resultado de sua pesquisa a mesma irá consultar à aplicação servidor, passando os parâmetros informados e sua interface. É de responsabilidade da aplicação servidor selecionar os dados junto ao SGBD e retornar a sua origem os dados devidamente padronizados no formato MPEG-7 necessário para sua reprodução.

Validando todo o processo, diagnosticamos que existem problemas na atualização da interface gráfica, onde alguns de seus componentes não se atualizam corretamente levando ao travamento da aplicação.

Para corrigir este problema da aplicação cliente será necessária a atualização dos componentes de interface, porém os dados estão chegando à aplicação cliente na estrutura correta e adequada para sua reprodução.

5.6 CONSIDERAÇÕES FINAIS

Após executados os testes acima, verificamos que a aplicação indexVideo possui os resultados esperados na transformação e transporte dos dados entre suas aplicações, porém

constamos que há problemas nos componentes que se refere a interface gráfica os quais não estão se atualizando, gerando problemas na aplicação cliente levando até a circunstância do usuário ter que reinicializar a aplicação.

Nestes critérios podemos afirmar que a aplicação indexVideo está funcionando perfeitamente com SGBD, havendo a necessidade de ajuste na interface gráfica para chegar o resultado esperado ao usuário final.

6 CONCLUSÃO

A elaboração deste trabalho tem o objetivo de aprimorar a aplicação indexVideo para torná-la operante com SGBD, melhorando a performance e a segurança das informações trafegadas nesta aplicação.

Optou-se por utilizar o SGBD Postgres, o qual atende as especificações conforme descrito no anexo A, onde o mesmo passou pela fase de testes, sem apresentar problemas ou indisponibilidades, para melhorar a manutenção da aplicação a nível de código fonte, seguindo o padrão já desenvolvido, assim facilitando o entendimento e a manipulação dos dados da aplicação.

Baseando-se em um modelo relacional desenvolveu-se a estrutura de dados para armazenamento em um SGBD, seguindo os padrões já definidos anteriormente na aplicação afim de manter o padrão de dados e a compatibilidade entre aplicação indexVideo e o padrão MPEG-7 sendo utilizado ao máximo o código fonte já desenvolvido.

No nível de código fonte foi criada uma camada de persistência de dados, possuindo todas as validações necessárias para operacionalizar a aplicação e a conversão dos dados, vindo da aplicação cliente, antes de serem salvas no SGBD, afim de já salvar os dados da aplicação no padrão MPEG-7, economizando tempo de execução na hora de recuperar as informações para enviar a aplicação cliente.

Com a utilização de SGBD, obteve-se ganhos significativos na de segurança de dados, agilidade nas pesquisas entre os dados, e os mesmos quando armazenados no SGBD seguem o padrão MPEG-7, tornando o retorno dos dados à aplicação cliente, mais ágil, pois não é necessário uma camada de conversão dos dados, afim de montar o arquivo XML temporário.

A montagem do arquivo XML temporário só se faz necessário, no momento de enviar o vídeo à aplicação cliente, a qual exige um arquivo de vídeo e um arquivo XML correspondente com os dados necessários para sua reprodução. Esta operação tem início com a solicitação dos dados do vídeo e suas devidas anotações ao SGBD, assim montando o arquivo XML para ser enviado a aplicação cliente junto ao vídeo.

Perante a funcionalidade de recuperação dos vídeos para seu envio, a aplicação cliente ficou mais lenta que anteriormente, devido à necessidade de montar um arquivo XML correspondente, onde no fluxo antigo da aplicação já estaria montado, porém se ganha em desempenho nas consultas e inclusão de dados e vídeos. No momento de incluir um vídeo não

se gera mais um arquivo XML correspondente ao vídeo, inclui-se os dados diretamente no SGBD, com isso se tornou mais seguro e mais ágil a inclusão dos vídeos no repositório da aplicação indexVideo e as consultas deste na aplicação são feitas diretamente no SGBD e retornando os dados direto a aplicação cliente, facilitando e agilizando a operacionalização do sistema.

A aplicação indexVideo torna-se mais segura e estável na conservação dos dados condizentes a mesma, devido a segurança já disponível no SGBD, e a facilidade na consulta dos dados.

A fim de garantir os dados armazenados no SGBD, sugere-se a implementação das rotinas de back-up, testes de desastres tais como perda de informação total e parcial, e as rotinas de restauração da base junto ao SGBD.

Alterando a interface gráfica para melhorar o aspecto da aplicação, aprimorar a sua identidade visual, visando melhor usabilidade junto ao usuário, buscando a comodidade, facilidade, a melhoria da aprendizagem e a alta disponibilidade da aplicação.

Para abranger maior volume de utilização junto ao indexVideo e estar disponível a qualquer instante, será necessário desenvolver uma interface na aplicação para rodar sob um browser de navegador da internet, possibilitando ao usuário final acessar o seu repositório de vídeos e anotações em qualquer lugar que possua acesso a mesma.

7 REFERÊNCIAS

BRAY, Tim; PAOLI, Jean; C. M. Sperberg-McQueen, **Extensible Markup Language (XML) 1.0** 13/02/1998 (<http://www.dis.uniroma1.it/~marte/sofflatina/documentazione.specificaXHTML.1.0.pdf>) – Acessado em 05/2010.

BOURRET, R. (2005) **XML and Databases**, <http://www.rpbouret.com/xml/XMLAndDatabases.htm> - Acessado em 05/2010.

CARBONERA, Joel Luis - **Ferramenta de indexação e recuperação semântica de conteúdo baseado em vídeo** – 2007 – Universidade de Caxias do Sul (UCS).

CHELLA, Marco Túlio - **Sistema para Classificação e Recuperação de Conteúdo Multimídia Baseado no Padrão MPEG-7** - 2003 - Universidade Estadual de Campinas (UNICAMP) (<http://inf.unisul.br/~ines/workcomp/cd/pdfs/2232.pdf>) Acessado em 05/2010.

GARCIA, Maxiwell Salvador; MENEZES, Anderson Luiz; BOSCARIOLI, Clodis - **Um Estudo sobre Gerenciamento de Dados XML em SGBDs Objeto-Relacionais e em SGBDs com suporte nativo a XML – 2008 – UNIOESTE** – (<http://www.students.ic.unicamp.br/~ra089057/pub/Alguns%20Trabalhos/Um%20Estudo%20sobre%20Gerenciamento%20de%20Dados%20XML%20em%20SGBDs%0AObjeto-Relacionais%20e%20em%20SGBDs%20com%20suporte%20nativo%20a%20XML,%20CONGED,%20Curitiba,%202008.pdf>) – Acessado em 05/2010

GRAVES, Mark. **Banco de Dados com XML** – 2003 – Editora Pearson.

MARTÍNEZ, José; KOENEN, Rob; PEREIRA, Fernando. **MPEG-7: The generic Multimedia Content Description Standard** (http://mpeg.chiariglione.org/tutorials/papers/IEEEEMM_mp7overview_withcopyrighth.pdf) 2002 - Acessado em 04/2010.

MARTÍNEZ, José. **MPEG-7 Overview**(version 10), 10/2004 (<http://mpeg.chiariglione.org/standards/mpeg-7/mpeg-7.htm>) – Acessado em 04/2010.

MPEG-7 (2009), **Sistema para Classificação e Recuperação de Conteúdo Multimídia Baseado no Padrão MPEG-7** (<http://inf.unisul.br/~ines/workcomp/cd/pdfs/2232.pdf>)
Acesso em fevereiro de 2010.

NAVATHE, Elmasri, **Sistemas de Banco de Dados**, - 2005 – Editora Pearson.

POSTGRESSQL, **Documentação do Postgresql 8.0.0 – 2010** – Postgresql – (<http://www.postgresql.org.br/docs>) – Acessado 05/2010.

SIMÕES, GONÇALO, **Conceitos Gerais Sobre Base de Dados**, - 2006 - <http://www.aprendercomastics.net/tic/materiaisapoio/ConceitosGeraiSGBD.pdf> - Acessado em 05/2010.

SOLOMON, Martin K., **Introdução a Sistemas de Bancos de Dados**, - 2004 – Editora Campus.

W3C (2001), **XMLSchema**, <http://www.w3.org/2001/XMLSchema-instance> - Acessado em 05/2010.

ANEXO A

1 MANUAL DE INSTALAÇÃO DA APLICAÇÃO INDEXVIDEO.

Caso o sistema operacional do servidor for microsoft, deverá ser instalado o programa “indexador.exe”, este programa irá extrair arquivos .dll necessários para a aplicação IndexVideo que necessita para funcionar corretamente, e se o sistema operacional for linux não é necessário instalar o “indexador.exe”.

Este instalador encontra-se na pasta raiz da aplicação indexVideo, abaixo os passo para sua instalação, na Figura 28 selecionamos a língua que será instalado, só disponível em *English*, após pressionar o botão *next*.

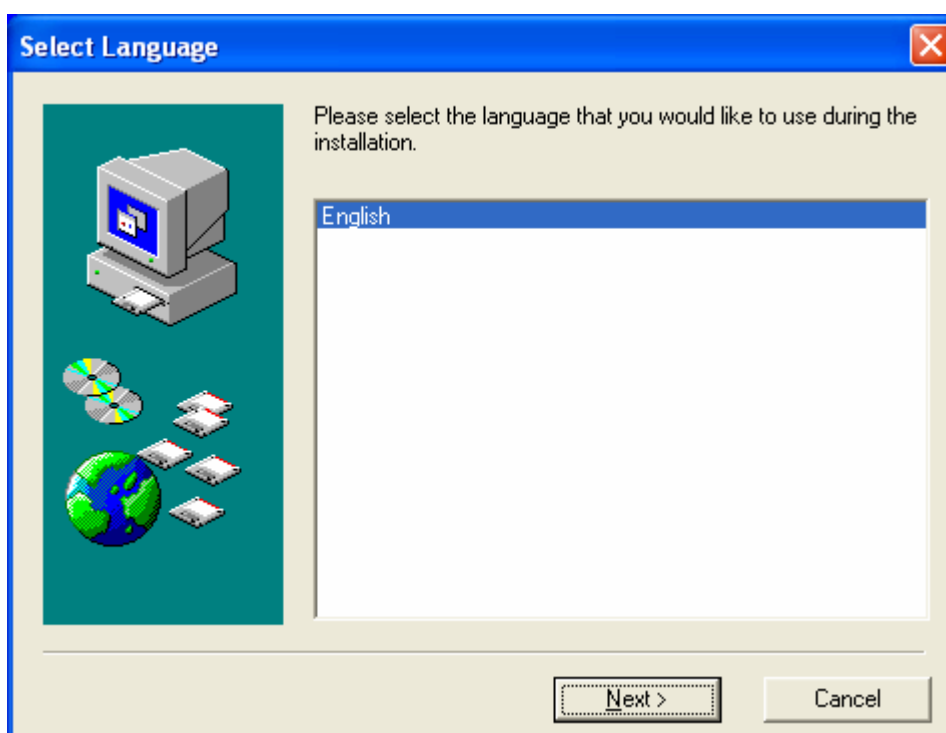


Figura 28 – Seleção de Idioma.

A Figura 29, demonstra o caminho a ser informado para sua instalação, manter o caminho padrão do sistema, se Windows 2003, XP, Vista e Windows 7, o caminho é C:\Windows\System32 demais sistema C:\Windows\System, após pressionar o botão *next*.

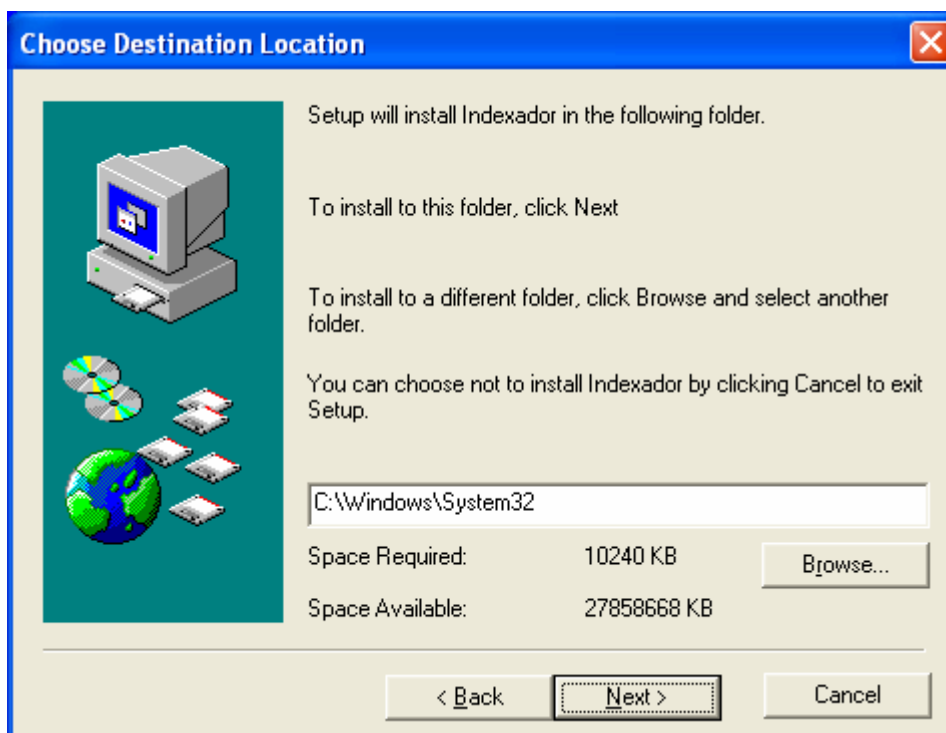


Figura 29 – Seleção do Caminho de Instalação

Na Figura 30, selecionamos o local a ser demonstrado no menu do sistema operacional, , após pressionar o botão next.

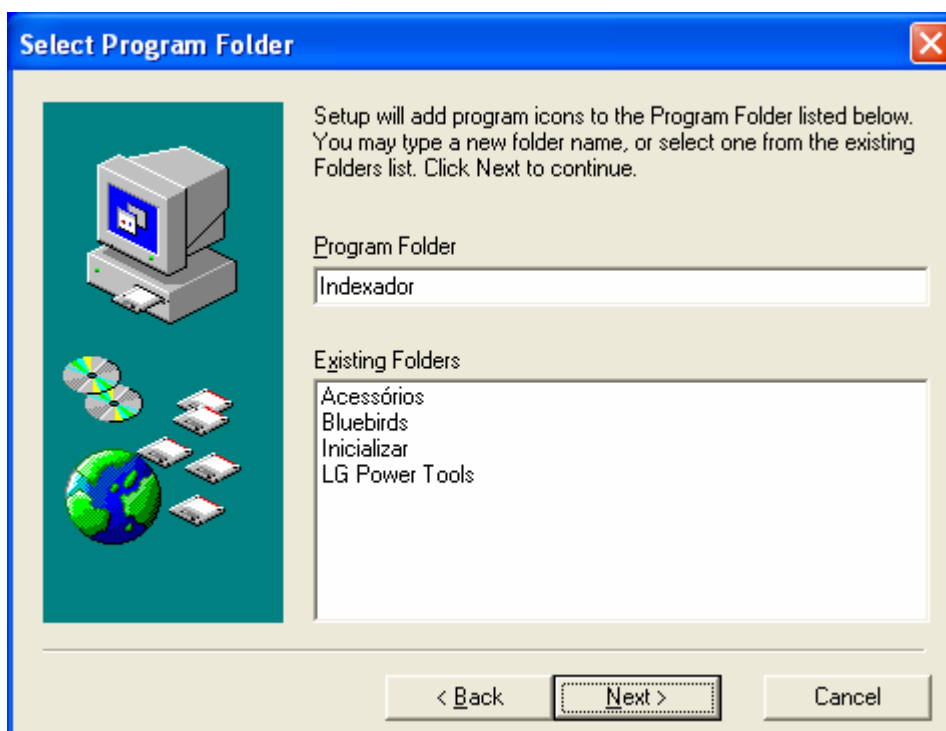


Figura 30 – Seleção do Menu a ser disponibilizado.

Na Figura 31, ao pressionar o botão *next* irá iniciar a instalação das DLL's conforme configurado nos passos anteriores.

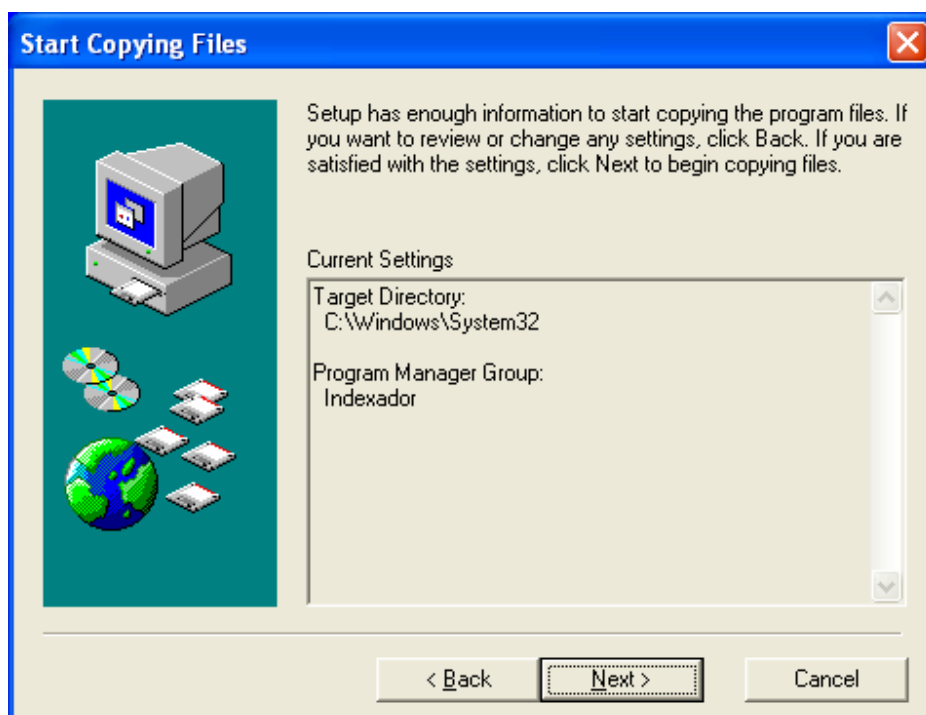


Figura 31 – Start da Instalação e copia dos arquivos

Após executado os passos anteriores, finalizaremos a instalação conforme a Figura 32, pressionar o botão *Finish*.

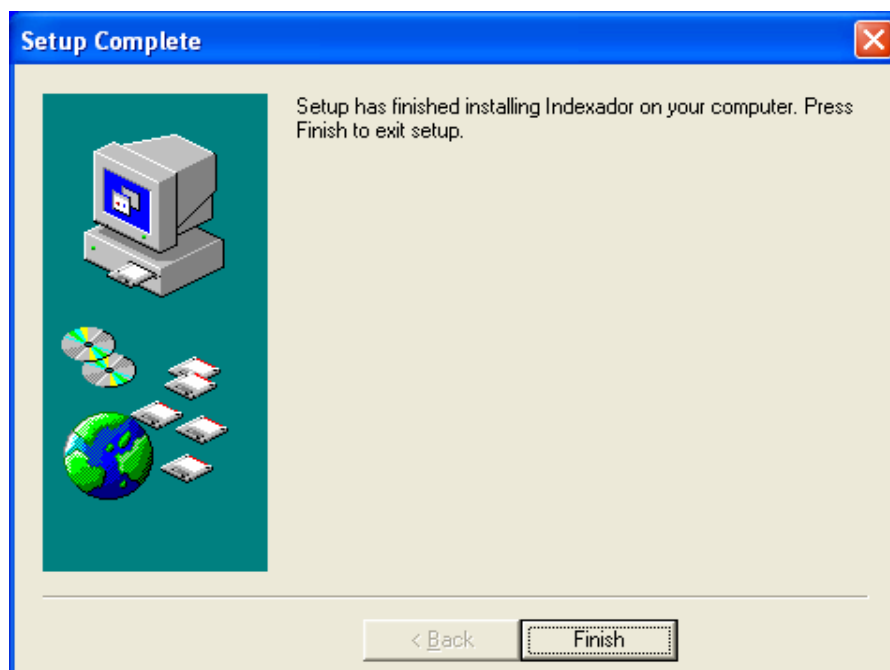


Figura 32 – Finalização do processo de instalação

Ao finalizar a instalação o programa “indexador.exe”, irá informar a versão e a instalação está concluída, conforme Figura 33, após pressionar o botão *OK*.

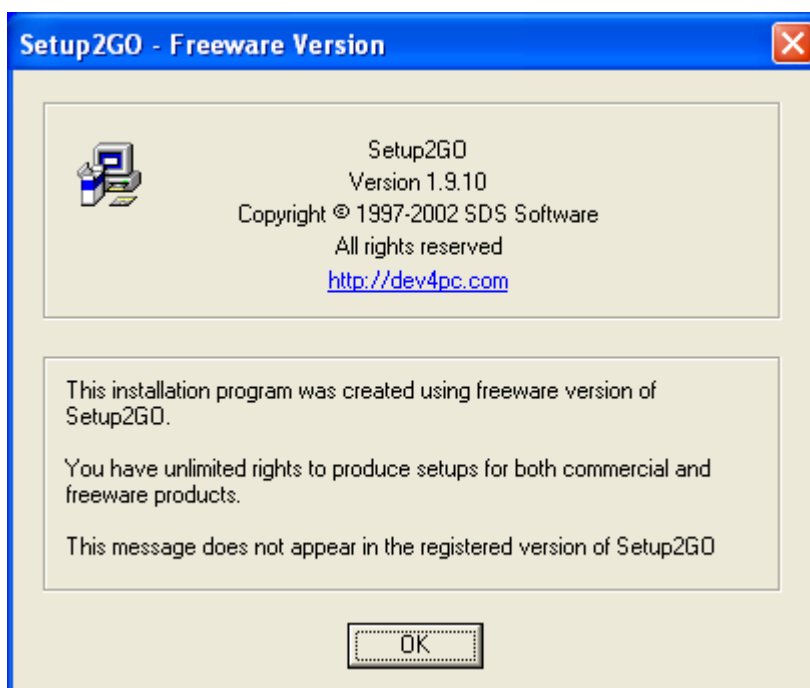


Figura 33 – Instalação Concluída

2 CRIAÇÃO DE DIRETÓRIOS

Após a instalação do programa “indexador.exe”, criamos os caminhos necessários para executar o programa.

Criar um diretório no servidor onde ficará armazenada a aplicação, por exemplo, “D:\Projetos\Java\indexVideo\”, criar o diretório onde será armazenado os vídeos, por exemplo, “D:\Projetos\Java\indexVideoRepositorio\videos\”, e criar um diretório onde será armazenados os XML’s temporários, por exemplo, “D:\Projetos\Java\indexVideoRepositorio\anotacoes\”.

Verificar se estes diretórios possuem as políticas de acesso do sistema operacional, liberado para a aplicação servidor poder manipular informações dentro dos diretórios acima informados.

No diretório que foi criado para armazenar a aplicação indexVideo criar dois arquivos, com os seguintes nomes, “configuracoesCliente.txt”, “configuracoesServidor.txt”.

No arquivo configuracoesCliente.txt, irá constar dados da porta de comunicação com o servidor e o *IP* do servidor que irá prover as informações e, no arquivo configuracoesServidor.txt, irá constar dados da porta de comunicação com o cliente que deverá ser a mesma do cliente, os dados dos repositórios, e os dados para conexão com o SGBD.

3 CONTEÚDO DOS ARQUIVOS DE CONFIGURAÇÕES

Dentro do arquivo configuracoesCliente.txt deverá constar o `ip_do_servidor=192.168.0.3`, este IP informado é do servidor que proverá a aplicação, a `porta_do_servidor=8800`, informará a porta do servidor que está provendo o serviço, conforme Figura 34.

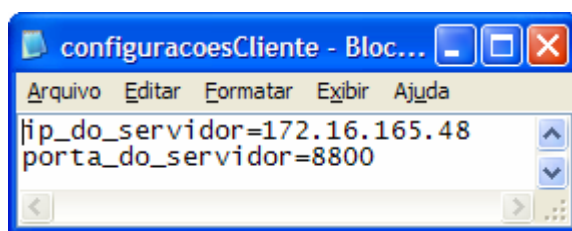
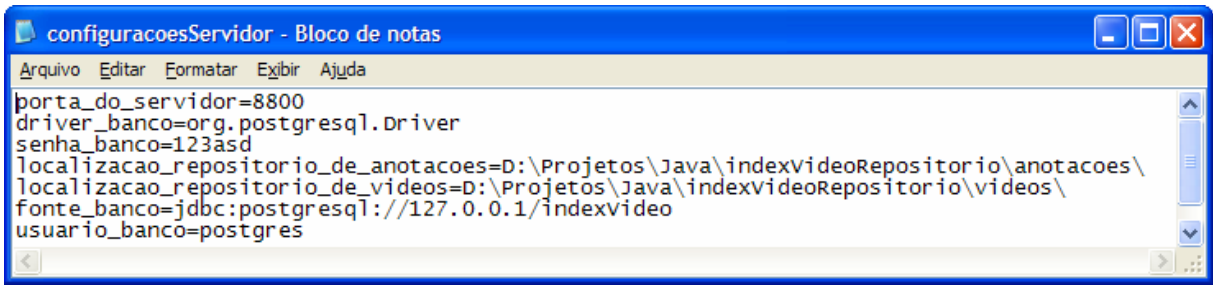


Figura 34 – Configuração do Cliente

Dentro do arquivo configuracoesServidor.txt deverá constar, `porta_do_servidor=8800`, informar a porta do servidor que irá se comunicar com o cliente, nos dois arquivos deverá possuir a mesma porta, `localizacao_repositorio_de_anotacoes=D:\Projetos\Java\indexVideoRepositorio\anotacoes\` informar o diretório que será gerado os XML temporários com as informações dos vídeos, `localizacao_repositorio_de_videos=D:\Projetos\Java\indexVideoRepositorio\videos\` informar o diretório que será armazenado os vídeos, e informar os dados de conexão com o SGBD, `driver_banco=org.postgresql.Driver`, `senha_banco=123asd`, `fonte_banco=jdbc:postgresql://127.0.0.1/indexVideo`, `usuario_banco=postgres`, conforme Figura 35, estes estão disponíveis a serem alterados através de uma interface gráfica, junto a aplicação servidor.



```
configuracoesServidor - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda
porta_do_servidor=8800
driver_banco=org.postgresql.Driver
senha_banco=123asd
localizacao_repositorio_de_anotacoes=D:\Projetos\Java\indexVideoRepositorio\anotacoes\
localizacao_repositorio_de_videos=D:\Projetos\Java\indexVideoRepositorio\videos\
fonte_banco=jdbc:postgresql://127.0.0.1/indexvideo
usuario_banco=postgres
```

Figura 35 – Configuração do Servidor

Para a aplicação identificar o arquivo de configuração o mesmo deverá ser criado com o nome `configuracoesServidor.txt` e constar as seguintes *tags* : `porta_do_servidor=`, `localizacao_repositorio_de_anotacoes=`, `localizacao_repositorio_de_videos=`, `driver_banco=`, `fonte_banco=`, `usuario_banco=`, para o seu perfeito funcionamento.

ANEXO B

1 ANÁLISE DE SGBD

Utilizando as informações de funcionamento da aplicação indexVideo versus um SGBD compatível com as tarefas propostas a desempenhar de armazenamento e gerenciamento de uma base de dados, iremos analisar o SGBD PostgreSQL, um sistema gerenciador de banco de dados objeto-relacional com código aberto, uma ferramenta free, disponível para download no site <http://www.postgresql.org.br/downloads>.

PostgreSQL surgiu em meados de 1986, liderado pelo Professor Michael Stonebraker, da universidade da Califórnia em Berkeley, inicialmente denominado de Postgres que anos mais tarde derivaria um pacote da estrutura do Postgres para surgir o PostgreSQL tornando-se um poderoso SGBD com código aberto, com abrangência e aceitação mundial.

O SGBD PostgreSQL suporta uma grande parte do padrão SQL:2003 e também oferece outras funcionalidades tais como, comandos complexos, chaves estrangeiras, gatilhos, visões, integridade transacional e controle de simultaneidade (multiversão), além disso permite ao usuário adicionar ou alterar, tipos de dados, funções, operadores, funções de agregação, métodos de índices e acesso a uma linguagem procedural para desenvolvimento de rotinas específicas dentro do SGBD, até permitindo fazer configurações específicas para o negócio de cada empresa poder se moldar a necessidades bem específicas do mercado.

2 DADOS TÉCNICOS DO POSTGRESQL

O SGBD PostgreSQL roda sobre as plataformas mais utilizadas pelo mercado atualmente, inclui-se nestas plataformas GNU/Linux, Unix (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), e MS Windows, suportando chaves estrangeiras, junções, visões, gatilhos e procedimentos armazenados em múltiplas linguagens, possuindo uma boa parte da tipagem de dados do SQL:1999, inclui-se nesta lista os tipos INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, e TIMESTAMP, suportando o armazenamento de objetos binários, inclui-se figuras, sons e vídeos, e suporta através de uma interface nativa de programação o C/C++, Java, .Net, Perl, Python, Ruby, Tcl, e ODBC.

Entre algumas sofisticações desta ferramenta podemos destacar o controle de concorrência multiversionado (MVCC), recuperação em um ponto no tempo (PITR), tablespaces, replicação assíncrona, transações agrupadas (savepoints), cópia de segurança a quente (on-line/hot backup), planejador de consultas(otimizador) e registrador de transações sequencial (WAL) para tolerância a falhas do sistema.

Suporta um conjunto de caracteres internacionais, codificação de caracteres multibyte, Unicode, sensibilidade ao tipo de letra (maiúscula e minúscula), possuindo uma grande escalabilidade, ao se tratar de usuários concorrentes ou até na quantidade de dados a serem geridos pela ferramenta.

Tendo uma visão dimensional das bases de dados geridas por este SGBD destacam-se alguns sistemas ativos e operantes em ambiente de produção, gerenciando mais de 4TB em sua base de dados com mais de 40.000 acessos simultâneas, nos demonstrando que este SGBD é robusto e suporta um grande conglomerado de dados e informações, e permite um número grande de acessos simultâneos, praticamente sendo o limitador o hardware que está provendo sustentabilidade a esta aplicação, porém a ferramenta possui alguns limitadores descritos na tabela abaixo:

Tabela de limitadores SGBD Postgress

Limitador	Valor
Tamanho máximo do banco de dados	Ilimitado
Tamanho máximo de uma tabela	32 TB
Tamanho máximo de uma linha	1.6 TB
Tamanho máximo de um campo	1 GB
Máximo de linha por tabela	Ilimitado
Máximo de colunas por tabela	250 – 1600 dependendo do tipo da coluna
Máximo de índices por tabela	Ilimitado

Tabela 01 – Limitadores SGBD Postgress