Consulta a Banco de Dados com o *Browser*[1]

Sérgio Furgeri[2]

Resumo

Nos últimos anos, com o avanço da Internet, uma das ferramentas de *software* que recebeu maior destaque, sendo utilizada mundialmente, se refere à ferramenta usada para a navegação na Internet ou, simplesmente, *Browser* como é conhecida. Sua utilização tem se tornado praticamente obrigatória em qualquer ambiente, seja uma residência, uma empresa, uma escola. Atualmente, já existem versões da ferramenta para diversos equipamentos distintos, como é o caso do celular, onde existe a possibilidade de utilizar o telefone para navegar em páginas elaboradas com a linguagem WML (Wireless Markup Language) através de um *micro Browser*. Esse artigo demonstra apenas um dos recursos já disponíveis: a consulta de banco de dados com o *Browser*. Serão demonstrados os procedimentos necessários para tornar possível uma consulta com instruções SQL através de uma applet desenvolvida com a linguagem Java, que fará a conexão com o banco de dados.

Palavras-chave

Browser; Java; *applet*; banco de dados; *policy file*; segurança

Comentário:

1 Trabalho desenvolvido na Faculdade do IPEP – Campinas – 2001.

2 Mestre em Informática pela PUC-C e professor do IPEP.

Database query with Browser[1]

Sérgio Furgeri[2]

Summary

In the last years, with the progress of the Internet, one of the software tools that received larger prominence, being globally used, refers to the tool used for the sailing in the Internet or, simply, Browser, as it is known. Its use has been turned essential in any it sets, be a residence, a company, a school. Now, versions of the tool already exist for several different equipments, as it is the case of the cellular, where the possibility exists of using the telephone to navigate in pages elaborated with the language WML (Wireless Markup Language) through a micro Browser. This article just demonstrates already one of the resources available: the database consultation with the Browser. The necessary procedures will be demonstrated to turn possible a consultation with instructions SQL through an applet developed with the language Java, that will make the connection with the database.

Keywords

Browser; Java; applet; database; policy file; security

Comentário:

1 Work developed at IPEP University – Campinas – 2001.
2 Informatics Master and Professor in IPEP.

1. Introdução

A utilização do *Browser* como uma ferramenta padrão tem se tornado uma realidade cada vez mais presente, tendo evoluído rapidamente nos últimos anos, recebendo destaque entre os *softwares* mais utilizados do planeta. Atualmente, a Internet é uma necessidade para pessoas e empresas, onde surgem novos e diversos serviços a todo instante, reduzindo distâncias, agilizando tarefas e tornando os processos mais eficientes. Entre os inúmeros serviços possíveis, destaca-se o acesso remoto a banco de dados, onde é possível, através de um *Browser*, acessar ou atualizar dados de qualquer local e de forma *on-line* através de inúmeras aplicações. Essas aplicações tem sido desenvolvidas a partir de Java, linguagem que tem recebido destaque nos últimos anos. Diversas entidades, principalmente financeiras, como é o caso do Banespa, investiram nessa tecnologia, apostando na segurança que a linguagem Java oferece.

Uma das grandes vantagens de uma aplicação criada em Java é sua característica multi-plataforma, isto é, pode ser executada a partir de diferentes tipos de sistemas operacionais e equipamentos. Apesar disso, ainda são raras as aplicações desenvolvidas nessa linguagem, em especial no mercado brasileiro, onde ainda estamos "engatinhando" em seu aprendizado. A utilização mais notória da linguagem Java se refere a aplicações voltadas à Internet, onde existem diversas ferramentas disponíveis que geram código Java, principalmente no campo das animações gráficas, como é o caso do tão conhecido Flash. Essa foi à utilização inicial dessa linguagem: a criação de animações para a Web. Entretanto, uma página apenas com animações, apesar de atraente, não contempla todas as funcionalidades exigidas por um site. Dessa forma, a linguagem Java começou a ser utilizada para o acesso de banco de dados via Internet, transformando o *Browser* numa ferramenta indispensável em qualquer ambiente empresarial, não apenas para o acesso a Internet, mas também para a troca de dados em uma Intranet.

O objetivo desse artigo é fornecer uma visão conceitual e técnica dos procedimentos necessários para criar uma aplicação desse tipo. Quais os itens relevantes para sua elaboração, seus requisitos de *software*, suas restrições de segurança, enfim, fornecer uma visão geral sobre o acesso remoto usando o *Browser*, executando um programa criado a partir da linguagem Java. Serão fornecidos todos os passos necessários para a criação e manipulação de um banco de dados no formato Access 2000. Para facilitar a criação e execução da aplicação, esse artigo considera que o banco de dados será consultado localmente com o *Browser*.

2. Breve histórico dos *Browsers*

A ferramenta para navegação de páginas Internet, conhecida como *Browser*, é relativamente recente. Apesar da Internet já existir desde 1970, o surgimento do *Browser* é atribuído a década passada, em 1993, através de uma ferramenta conhecida como *Mosaic*. Logo a seguir, em 1994, surge a primeira versão da ferramenta da *Netscape*, trazendo algumas inovações e se consolidando no mercado, passando a ser o *Browser* mais usado durante vários anos, aproximadamente até meados de 1997. Durante esse período, a *Microsoft* desenvolveu o *Internet Explorer* e disputou a liderança com o *Netscape Navigator*, travando uma verdadeira "guerra" na disputa pela liderança e preferência dos usuários. A partir de 1998, o *Internet Explorer* passou a ser o *Browser* mais aceito e continua sendo até hoje, já em sua versão 6.0. Depois

disso, a *Netscape* passou por dificuldades e acabou sendo incorporada pelo AOL que optou por descontinuar o *Navigator*, uma pena. Sendo assim, o *Internet Explorer* tende a ser a ferramenta padrão de navegação, uma vez que já não existe um concorrente a altura. Nesse período de oito anos, desde o surgimento do *Mosaic*, surgiram diversas ferramentas de navegação, de diversos fabricantes, e o *Browser* teve que evoluir muito. A cada ano, novas funcionalidades tiveram que ser incorporadas à ferramenta, de modo a se adaptar as novas realidades do mercado. No princípio, o *Browser* necessitava apenas demonstrar conteúdo texto na tela, uma vez que a Internet estava surgindo e não havia possibilidade do transporte de imagens, devido às restrições de velocidade das linhas de transmissão. Mais tarde, surgiu à necessidade de incorporar figuras, animações, multimídia, linguagens de programação, linguagens de formatação, entre outras. Para adaptar o *Browser* as diferentes necessidades, surgiram os chamados *Plug-ins*, rotinas de *softwares* que ao serem instaladas, adicionam novas funcionalidades. Portanto, o *Browser* está em constante evolução e inovação. É bem provável que, em breve, tenhamos aplicações completas rodando sobre essa ferramenta.

A tabela seguinte demonstra as datas mais marcantes com relação aos principais versões dos *Browsers* do mercado.

Mosaic		***Nestcape***		***Internet Explorer***		***Opera***	
1.0	Nov 1993	1.0	Dez 1994	1.0	Ago 1995	2.1	Dez 1996
2.0	Nov 1995	1.1	Abr 1995	2.0	Nov 1995	3.0	Dez 1997
2.1	Jan 1996	2.0	Mar 1996	3.0	Ago 1996	3.5	Nov 1998
3.0	Jan 1997	3.0	Ago 1996	4.0	Out 1997	4.0	Jun 2000
	Encerrado	4.0	Jun 1997	5.0	Mar 1999		
		4.5	Out 1998	5.5	Jul 2000		
		6.0	Nov 2000	6.0	Jun 2001		
			Encerrado				

3. Itens necessários para a manipulação de banco de dados com o *Browser*.

A tarefa de se elaborar aplicações que possam manipular banco de dados com o *Browser*, apesar de ser conceitualmente simples, na prática são necessárias diversas ferramentas e procedimentos. Entretanto, "ver a coisa funcionar" é algo muito gratificante, pois são raras as aplicações que fazem isso, uma vez que o mercado nacional carece de materiais e livros sobre esse assunto.

Os itens necessários, envolvidos na criação e manipulação de banco de dados com o *Browser* são:

1. Uma ferramenta para desenvolvimento de aplicações em Java (SDK 1.3 for Windows), disponível para *download* no site da Sun. Essa ferramenta é a responsável pela compilação e execução de programas em Java.
2. Uma ferramenta para a criação de um arquivo de política (policy file) denominada **policytool**. Essa ferramenta já está inclusa no SDK (item 1).
3. O plug-in Java 2 (j2re), disponível para *download* no site da Sun. Esse plug-in é necessário para que o *Browser* consiga realizar diversas tarefas inclusas nos programas Java, entre elas manipular bancos de dados.

4. Um editor de textos qualquer para a criação do programa Java. O programa Java será construído na forma de um *Applet*, o formato que o *Browser* reconhece para executar programas desenvolvidos em Java. Além disso, para chamar a *Applet* será necessária a criação de um arquivo HTML.
5. Um aplicativo em Java denominado HTMLConverter que fará a conversão do arquivo HTML criado, adicionando certos códigos necessários para invocar o plug-in no momento em que o *Browser* carregar a *Applet*.
6. Um *Browser*, tipicamente o Internet Explorer.
7. O Access da Microsoft para a criação do banco de dados.
8. Um driver ODBC para realizar a ponte entre a fonte de dados ODBC do Access e a fonte JDBC do Java. Normalmente esse driver já está instalado em uma máquina com o Windows. Para que essa ponte seja possível é necessário instalar o driver JDBC da Sun, entretanto, ao instalar o SDK do item 1, esse driver já é incluído na máquina.

4. A criação do banco de dados

A criação do banco de dados é bem simples. Para realizar essa tarefa será utilizado o *Microsoft Access* 2000 e para facilitar os procedimentos, será definido um diretório de trabalho, onde todos os arquivos criados serão armazenados. Crie uma pasta chamada MeuBD no *drive* C (C:\MeuBD) e realize os procedimentos seguintes:

- Abra o *Microsoft Access* a partir do botão iniciar de seu desktop.
- Crie um novo banco de dados: escolha um novo banco de dados vazio.
- Quando o novo banco de dados abrir, escolha a pasta MeuBD e crie o banco de dados com o nome **Banco**.
- Quando o Access abrir o novo banco de dados, crie uma nova tabela no modo estrutura: dê um duplo clique em Criar tabela no modo estrutura.
- Crie a tabela de clientes, seguindo as informações seguintes:

Nome do campo	**Tipo de dados**	**Tamanho**
FICODIGO	Texto	5
FINOME	Texto	35
FIGENERO	Texto	8
FIPRODUT	Texto	15
FIDATCOM	Data/Hora	
FIANOPRO	Texto	4
FITEMDUR	Texto	3

- Defina o campo **FICODIGO** como sendo a chave primária. Para isso, pressione o botão direito do mouse sobre esse campo e escolha chave primária.
- Salve a tabela com o nome **Filmes**.
- Feche o Microsoft Access.

Para que seja possível consultar dados futuramente, obviamente o banco de dados deverá ser preenchido, isto é, diversos filmes deverão ser cadastrados.

5. A Configuração do sistema

Para que seja possível acessar banco de dados em Java, é necessário criar uma fonte de dados ODBC, de maneira a estabelecer uma ponte de comunicação entre o Access e o Java. A seguir, são demonstrados os procedimentos necessários para a criação da fonte de dados.

- Clique no botão Iniciar, escolha configurações e Painel de Controle.
- Dê um duplo clique sobre o item ODBC 32 bits.
- Acesse a guia NFD (Nome das Fontes de Dados) de usuário e pressione o botão adicionar, conforme demonstra a figura 1.

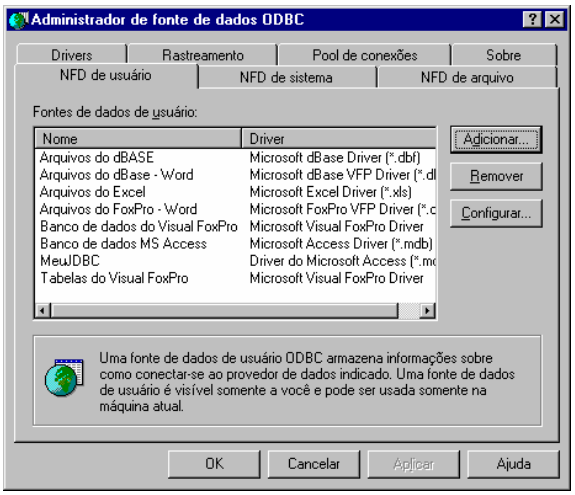

Fig. 1 – Os *drivers* instalados

- Escolha ***Microsoft Access Driver*** e pressione concluir. Dessa forma aparecerá uma janela apresentada pela figura 2. No campo Nome da Fonte de Dados digite **MeuBanco**.

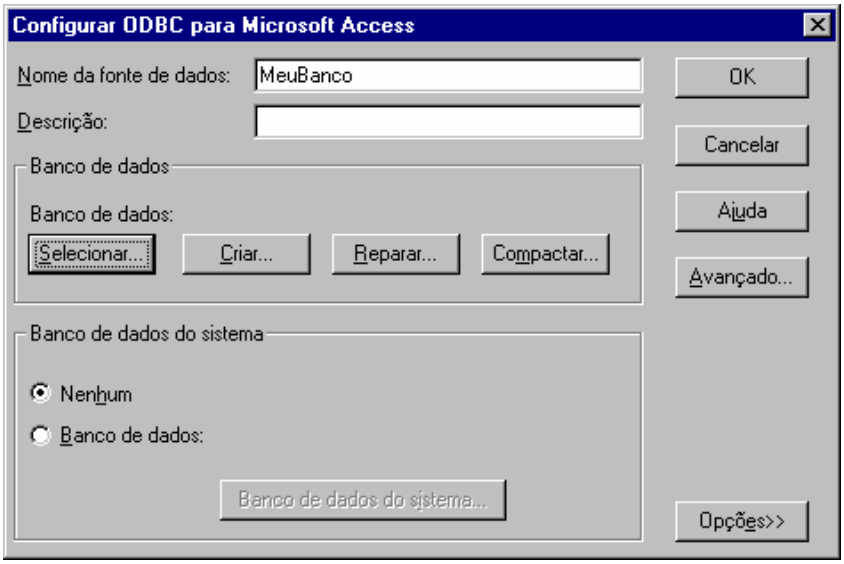

Fig. 2 – Criação da Fonte de dados

- Pressione o botão Selecionar e escolha o banco de dados criado pelo *Access* (**Banco.mdb**), localizado na pasta c:\MeuBD e clique em OK

- Sendo assim, o nome do banco de dados que será a fonte de dados ODBC será selecionado. Pressione OK para finalizar essa operação. A fonte de dados criada (**MeuBanco**) deve aparecer entre as Fontes de dados do usuário, conforme demonstra a figura 3.

Nesse ponto, o banco de dados que será consultado pelo *Browser* já está corretamente criado e devidamente configurado.

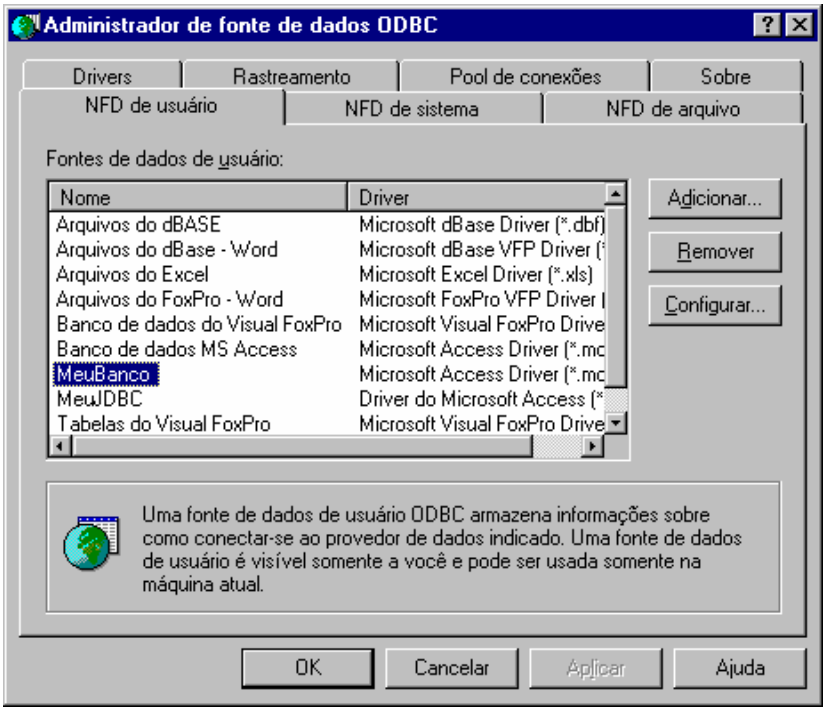

Fig. 3 – A Fonte de dados criada

6. A criação da aplicação em Java

Nesse item é apresentada a *Applet* que será utilizada para a visualização do banco de dados. Essa *Applet* permite que instruções SQL sejam utilizadas para a consulta ao banco de dados. Nesse item, é considerado que o leitor possui conhecimentos básicos sobre a linguagem Java, uma vez que diversos procedimentos serão necessários para a execução da aplicação.

```
import java.awt.*;                    // carrega as bibliotecas
import java.awt.event.*;
import java.applet.Applet;
import java.sql.*;

public class UsaSQL extends Applet
{
```

```
   TextArea area=new TextArea();                    // inicializa os objetos
   Label l1 = new Label("Instrução SQL");
   TextField sql = new TextField(50);
   TextField status = new TextField(50);
   Connection MinhaConexao;   // declara uma conexão

   public void init()    // método executado quando a Applet é carregada
   {
   sql.setEditable(false);   // objetos não editáveis
   area.setEditable(false);
   status.setEditable(false);
   add(area); add(l1);add(sql);add(status);  // adiciona os objetos a Janela
   try
    {
    String url = "jdbc:odbc:MeuBanco";
    // carrega o driver jdbc-odbc
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

    /* estabele a conexão chamada MinhaConexao com o URL definido
      anteriormente usando um nome de usuário e uma senha */
    MinhaConexao = DriverManager.getConnection(url, "anyname", "anypswd");
    // define a instrução SQL
    sql.setText("SELECT ficodigo,finome FROM filmes WHERE ficodigo>'03400'");
    status.setText("");
    area.setText("");
    Statement MeuState = MinhaConexao.createStatement();

    // Executa a instrução SQL na tela
    ResultSet rs = MeuState.executeQuery(sql.getText());
    area.append("Codigo   Nome\n");
    area.append("--------     ----------------------------------------------------------------\n");
    String codigo="", nome="";

    while(rs.next())   // realiza a leitura de todos os registros da tabela
       {
        codigo = rs.getString("ficodigo");
        nome = rs.getString("finome");
        area.append(codigo + "    " + nome + "\n");
       }
     status.setText("Instrução executada com sucesso!");
    MinhaConexao.close();          // Encerra a Conexão
    }
    catch(java.lang.Exception ex)
     {
     status.setText("Instrução nao reconhecida");
     }
  }
}
```

A *Applet* UsaSQL que abre o banco de dados e permite instruções SQL

Digite e salve a *Applet* UsaSQL com o nome **UsaSQL.java** dentro do diretório **MeuBD** e realize sua compilação. Se esta for realizada com sucesso, será gerado o arquivo **UsaSQL.class**. Esse é o arquivo que será carregado e executado pelo *Browser* através do arquivo HTML, demonstrado a seguir.

Da mesma forma, digite o arquivo a seguir e salve-o na pasta MeuBD com o nome **UsaSQL.html**.

```
<html>
<applet code="UsaSQL.class" width=450 height=400>
</applet>
</html>
```

Nesse ponto, o diretório MeuBD deve conter três arquivos: **UsaSQL.java, UsaSQL.class e UsaSQL.html**. Ainda não é possível carregar o arquivo com o *Browser*, pois ainda faltam diversos procedimentos a serem realizados.

7. Questões relativas à segurança da linguagem Java

As *applet*s foram planejadas para serem carregadas de um site remoto e então serem executadas localmente. Desta forma, para evitar que *applet*s criadas por pessoas sem ética danifiquem o sistema local onde estão sendo executadas, isto é, atuem como um vírus, acessando informações sigilosas, alterando ou ainda apagando dados armazenados no sistema, foram impostas algumas restrições a seu funcionamento.

As *applet*s são executadas pelo *Browser* e monitoradas por um gerenciador de segurança (***applet security manager***) que lança uma interrupção do tipo ***SecurityException*** caso a *applet* viole alguma das regras de segurança impostas. Pelo motivo da *applet* ser monitorada durante sua execução, é comum dizer que as *applet*s operam dentro de uma caixa de areia, algo que impossibilita que elas saiam de seu raio de atuação. Esse modelo é conhecido por ***the sandbox model***.

As tarefas que as *applet*s podem realizar são: exibir imagens, executar sons, processar o acionamento do teclado e mouse e se comunicar com o *host* de onde foram carregadas. Por outro lado, as *applet*s estão sujeitas as seguintes restrições: não podem executar programas localmente instalados, não podem se comunicar com outros *hosts* exceto de onde a *applet* foi carregada, não podem ler e escrever no sistema de arquivos local e não podem obter informações do sistema onde operam exceto sobre a JVM onde operam.

Para retirar essas restrições pode ser criado um arquivo de política (*policy file*) o qual será demonstrado a seguir.

7.1 - Criação do *Policy File*

Para a criação do arquivo que controla o acesso das *applet*s pode ser usado qualquer editor de textos comum, entretanto, para que o aprendizado da sintaxe não seja necessário, a Sun criou uma ferramenta para a criação desses arquivos. Trata-se da PolicyTool, executada diretamente na linha de comando do DOS. Ao executar essa ferramenta aparecerá a tela demonstrada pela figura 4.

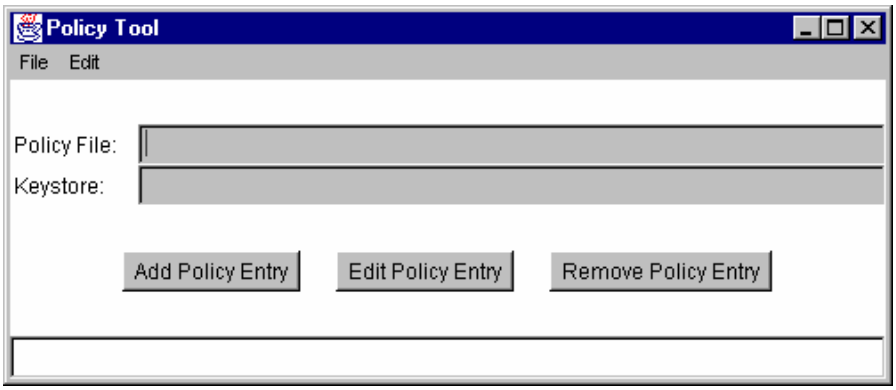

Fig. 4 – A Tela de abertura da ferramenta *Policytool*

Ao iniciar a ferramenta, provavelmente ela enviará uma mensagem alertando não ter encontrado o arquivo policy default, que normalmente fica armazenado na pasta do *Windows*. Para criar um novo arquivo de *policy* clique no botão ***Add Policy Entry***. Aparecerá uma outra tela contendo os campos:

***CodeBase*:** representa o diretório onde o arquivo de *policy* está armazenado.

Digite → **file:/c:/**MeuBD.

***SignedBy*:** representa um apontamento para um arquivo de assinatura que pode ser usado como chave para criptografia. Essa opção não será utilizada nesse artigo.

Para prosseguir, pressione o botão ***Add Permission***. Aparecerá a tela demonstrada na figura 5, onde deverá ser escolhida a opção ***AllPermission***. Normalmente, configurações mais específicas para a applet poderiam ser escolhidas, entretanto, com fins de simplificação, será escolhida essa opção, isto é, a applet poderá realizar qualquer tipo de operação com o banco de dados.

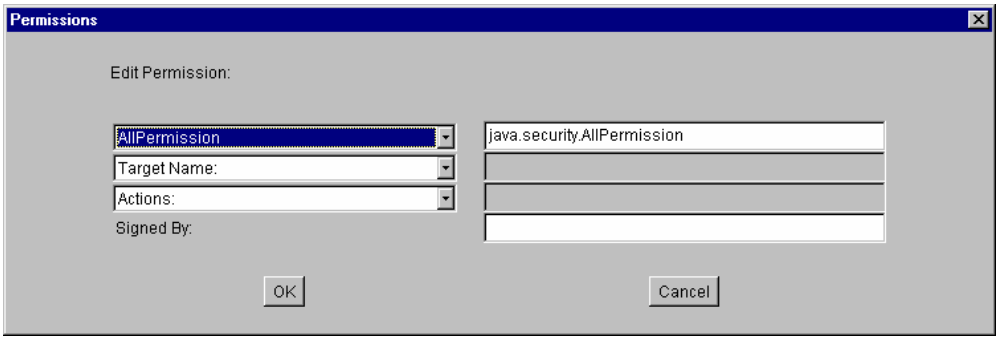

Fig. 5 – Configuração do arquivo de *policy*

A seguir, pressione o botão **OK** e na tela que surge ***Done***. Nesse ponto o arquivo de *policy* está completo, só falta ser salvo. No menu File escolha ***Save As***, localize o diretório **C:**\MeuBD, coloque o nome **meu_policy** no nome do arquivo e pressione o botão **Salvar**. Uma mensagem apontando o sucesso da gravação deverá aparecer.

Com isso foi criado um arquivo de *policy* com as seguintes informações: O *Applet* localizado no diretório C:\MeuBD (poderia ser um site), tem total permissão para manipular qualquer arquivo. Ressalta-se que esses diversos tipos de permissões são possíveis com a utilização da *Policytool*, entretanto, para simplificar nossos

procedimentos, definimos que a *applet* possui permissão total. O código fonte gerado pela *policytool* para o arquivo **meu_policy**, é mostrado a seguir.

```
/* AUTOMATICALLY GENERATED ON Wed Sep 26 16:18:09 BRT 2001*/
/* DO NOT EDIT */
grant codeBase "file:/c:/MeuBD" {
  permission java.security.AllPermission;
};
```

7.2 - **Acrescentando o *Policy File* criado ao arquivo *Security* do Sistema.**

Para manipular arquivos com o *Browser* é necessário alterar o arquivo que contém as propriedades de segurança do sistema. Esse arquivo encontra-se no diretório JAVA_HOME (de acordo com o que está definido no autoexec.bat) e seu nome é **java.security**. Ele está armazenado em uma pasta chamada ***security*** dentro do diretório JAVA_HOME. Um endereço típico pode ser: C:\Arquivos de programas\JavaSoft\JRE\1.3.1\lib\security\java.security.

O arquivo java.security da máquina deve ser alterado e isso pode ser realizado com o Edit do DOS. Encontre a linha **policy.url.2=file:${user.home}/.java.policy)**. Depois dessa linha, acrescente a linha **policy.url.3=file:/C:/MeuBD/meu_policy**. Nesse ponto o sistema já está configurado corretamente para permitir que o *Browser* manipule o banco de dados, entretanto, ainda é necessário converter o arquivo UsaSQL.html com a ferramenta ***HTMLConverter***. Só assim será possível executar a *applet* UsaSQL.class com o *Browser*.

8.0 Conversão do Arquivo HTML

O arquivo HTML (UsaSQL.html) criado anteriormente, não permite que a *applet* funcione corretamente, apesar de ser carregada pelo *Browser*. É necessário invocar o *plug-in* do Java, no momento de carregar a *applet*. Isso é realizado pelo próprio arquivo HTML, ou seja, ele deve conter códigos especiais que "chamarão" o plug-in, quando a *applet* for carregada. Esses códigos são relativamente complexos e para facilitar um pouco o trabalho do desenvolvedor, a Sun criou uma ferramenta que realiza automaticamente a conversão do arquivo HTML, trata-se da ferramenta **HTMLConverter**, disponível no site da Sun para *download*.

Para instalar essa ferramenta é necessário fazer seu *download* a partir do site da Sun no seguinte endereço: http://java.sun.com/products/plugin/converter.html. Ao entrar no site, basta escolher a plataforma de desenvolvimento e realizar o *download*. Quando esse artigo foi escrito o *download* era fornecido gratuitamente e a versão usada foi *HTMLConverter* 1.1.1 para a plataforma *Windows* (arquivo htmlconv111-win32.exe, tamanho 3,447,127 bytes).

Ao executar a ferramenta a janela do *HTMLConverter* deverá abrir, conforme demonstrado na figura 6.

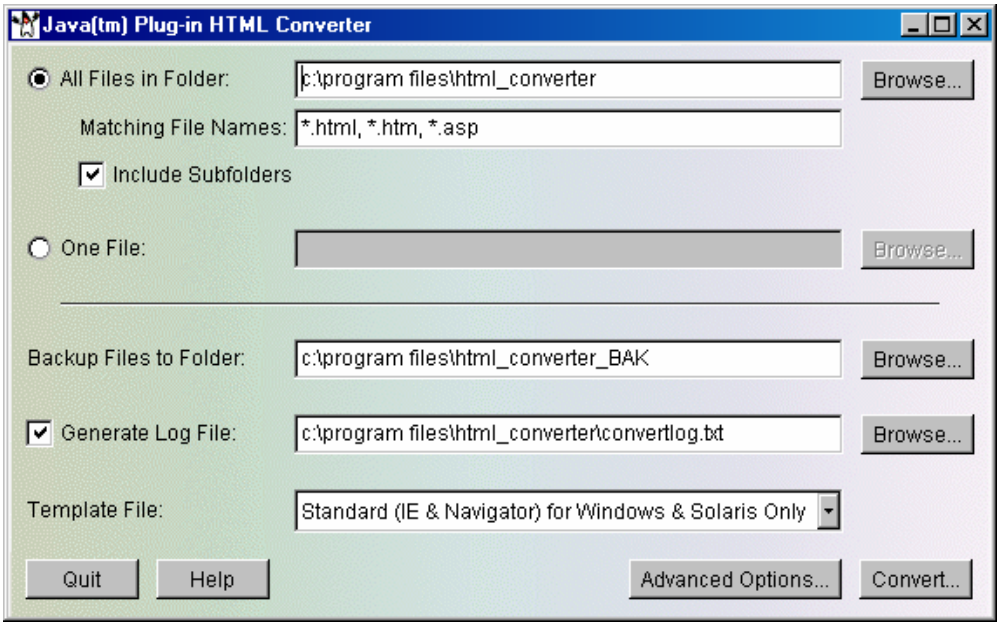

Fig. 6 – A ferramenta *HTMLConverter*

Para converter o arquivo HTML, em ***One File*** escolha o diretório C:\MeuBD e o arquivo UsaSQL.html e pressione o botão ***Convert***. Com isso, o arquivo UsaSQL.html será convertido e está preparado para ser utilizado pelo *Browser*.

9.0 Execução da *applet* com o *Browser*

Uma vez realizados todos os procedimentos descritos anteriormente, a aplicação está pronta para ser executada. Para isso, o arquivo UsaSQL.html deve ser aberto com o *Browser*. Se tudo estiver correto, deverá aparecer a tela mostrada pela figura 7. Essa aplicação é extremamente simples, demonstrando apenas o resultado de uma consulta realizada através de uma instrução SQL. Evidentemente, novas funcionalidades podem ser adicionadas dependendo das necessidades do usuário.

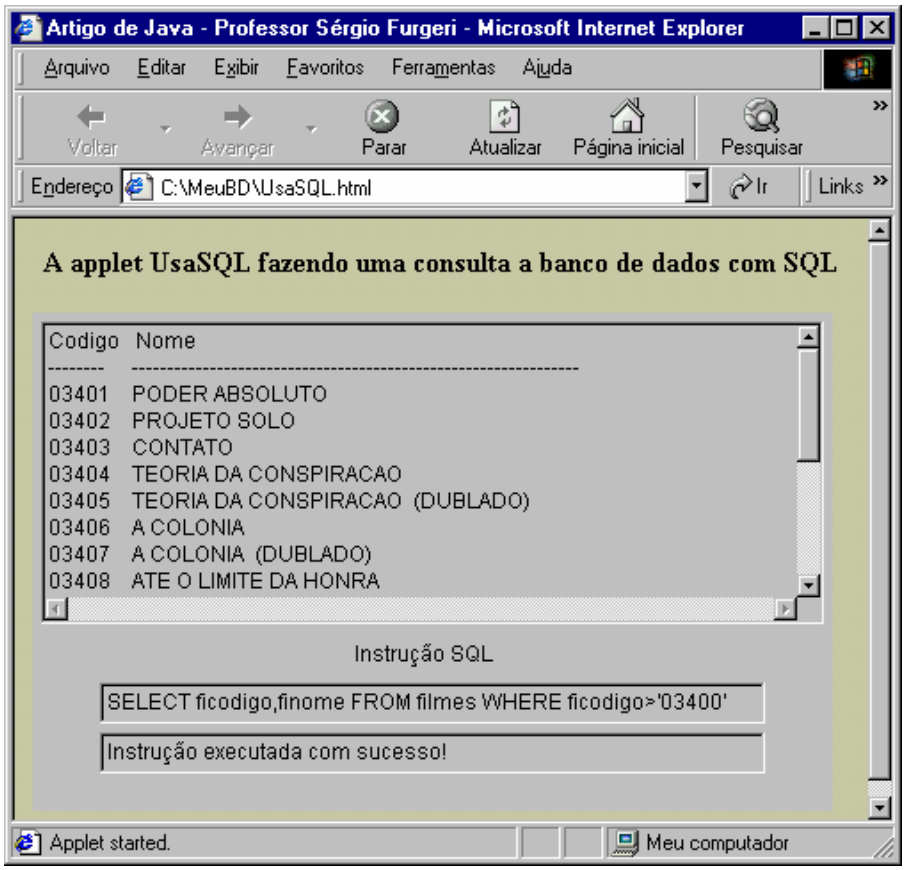

Fig. 7 – A tela de resultado da instrução SQL

10. Conclusão

A linguagem Java promete subir no *ranking*, assumindo a condição de estar entre as linguagens de programação mais utilizadas nos próximos anos, dada sua independência de plataforma e recursos para tratamento de dados remotos. A cada dia novos recursos são adicionados a linguagem, aumentando o leque de aplicações que podem ser executadas através de um *Browser*.

Apesar das limitações de segurança impostas para o uso *applet*s, nota-se que a consulta à banco de dados pode ser realizada de maneira satisfatória, apesar das limitações com relação à velocidade. Evidentemente, a utilização do *Browser* para consulta direta ao banco de dados conforme foi apresentado nesse artigo, não representa a maneira mais eficiente para tal processo, uma vez que aplicações multiusuárias podem representar um sério risco, principalmente no que diz respeito ao acesso concorrente. A maneira mais eficiente de acessar banco de dados se refere à utilização de *Servlets*, aplicações executadas através do Servidor Java, item não abordado nesse artigo. Apesar dessas limitações, o acesso à banco de dados através do *Browser* pode ser indicado para pequenas aplicações, onde velocidade e acesso concorrente não sejam uma real necessidade. Uma outra vantagem dessa abordagem se refere à simplicidade da aplicação, uma vez que aplicações envolvendo *Servlets* tendem a ser mais complexas.

Apesar da aplicação aqui apresentada não estar completa para o uso na Internet, prevendo o acesso concorrente, creio que o objetivo em demonstrar a possibilidade de consultar um banco de dados via *Browser* tenha sido alcançado.

11. Referências

Livros:

[Damasceno, 1997] Américo Damasceno, *Aprenda Intranets e Extranets com Java 1.1.*, Editora Érica, São Paulo, 1997

[Deitel, 2000], Harvey M. Deitel e Paul J. Deitel - *Java How to Program (Java 2), Third Edition,* 2000.

[OAKS, 1999] Oaks Scott, Segurança de dados em JAVA, Editora Ciência Moderna Ltda, Rio de Janeiro, 1999.

[Speegle, 2001] Gregory Speegle, *JDBC: Practical Guide for Java Programmers* – 2001.

Consulta a Sites:

Jaco Web Security – *Assinatura de Applet*
http://www.lcmi.ufsc.br/jacoweb/restrito/documentos/assinatura/index.htm#assinar
Consulta realizada em agosto de 2001.

Java Security Architecture (JDK1.3) Version 1.0.
Sun Microsystems Inc. Moutain View, CA, October 1998.
(http://java.sun.com/j2se/1.3/docs/guide/security/spec/security-spec.doc.html) -
Consulta realizada em setembro de 2001.

Policy Tool - *Policy File Creation and Management Tool*.
Sun Microsystems Inc. Moutain View, CA, October 1998.
http://java.sun.com/j2se/1.3/docs/tooldocs/win32/keytool.html -
Consulta realizada em setembro de 2001.

Summary of JDK 1.2 Security Tools.
Sun Microsystems Inc. Moutain View, CA, October 1998.
(http://java.sun.com/j2se/1.3/docs/guide/security/SecurityToolsSummary.html) -
Consulta realizada em setembro de 2001.

Tony Lonton - *Access the world's biggest database with Web DataBase Connectivity* -
http://www.javaworld.com/javaworld/jw-03-2001/jw-0316-webdb. Html?
Consulta realizada em setembro de 2001.

Wayne J Mallek, *Step by Step Instructions for Implementing JDBC*
http://shrike.depaul.edu/~wmallek/420/database.html.
Consulta realizada em agosto de 2001.