

INTERFACAMENTO COM A INTERNET PARA DIVULGAÇÃO DA TECNOLOGIA DA INFORMAÇÃO APLICADA AO AGRONEGÓCIO E CIÊNCIAS AMBIENTAIS

Dierone César Foltran Júnior

foltran@uepg.br
Universidade Estadual de Ponta Grossa
Dep. de Informática. Av. Gal. Carlos Cavalcante, 4748, CEP: 84030-000, Ponta Grossa, PR
Telefone (0xx42) 227-3444, Fax: (0xx42) 220-3072

Adriano Ferrasa

ferrasa@uepg.br
Universidade Estadual de Ponta Grossa
Curso de Bacharelado em Informática, Departamento de Informática, Ponta Grossa, PR
Telefone (0xx42) 223-1587

Islenho de Almeida

islenho@netscape.net
Universidade Estadual de Ponta Grossa
Curso de Bacharelado em Informática, Departamento de Informática, Ponta Grossa, PR
Telefone (0xx42) 226-5352

Marcelo Giovanetti Canteri

mgcanter@convoy.com.br
Universidade Estadual de Ponta Grossa
Dep. de Informática. Av. Gal. Carlos Cavalcante, 4748, CEP: 84030-000, Ponta Grossa, PR
Telefone (0xx42) 225-2619, Fax: (0xx42) 220-3342

Resumo

O emprego da informática avançada em atividades ligadas à produção agrícola permite maior controle dos processos produtivos. O desenvolvimento de aplicativos distribuídos, com acesso a base de dados e execução multiplataforma são as principais necessidades de um desenvolvimento consciente para aplicação na Web. E os aplicativos Web, utilizados inclusive em sistemas de e-business, vêm ocupando um lugar de destaque, principalmente entre as grandes empresas. As principais tecnologias utilizadas na implementação da ferramenta para construção de páginas interativas, sendo sua aplicação destinada ao uso em Agronegócios e Ciências Ambientais. Para tentar resolver esse problema, a linguagem Java tem se mostrado uma ótima alternativa, visto que um de seus principais argumentos é a independência de plataforma. Algumas das características do software: ser distribuído; acessado através de um navegador internet (browser); utilizar-se da tecnologia de Java (applets, RMI e servlets); provêr uma base consistente de dados para geração automática de páginas para Web; os clientes devem possuir uma interface fácil e completa para o desenvolvimento de tutoriais; possibilitar a utilização de recursos de imagens.

Abstract

The employment of the advanced computer science in activities linked to the agricultural production allows larger control of the productive processes. The development of distributed applications, with access the data bases and multiplatform execution are the main needs of a development for application in the Web. It is the applications Web, used besides in systems of and-business, they come occupying a prominence place, mainly among the great companies. The main technologies used in the implementation of the tool for construction of interactive pages, being its application destined to the use in Agrobusiness and Environmental Sciences. To try solve that problem, the language Java a great alternative it has been showing, because one of its main arguments is the platform independence. Some of the characteristics of the software: to be distributed; accessed through a navigator internet

(browser); to use of the technology of Java (applets, RMI and servlets); to prover a consistent data base for automatic generation of Web pages; the customers should possess an easy interface and completes for tutorials development; to facilitate the use of resources of images.

Palavras Chaves

Sistemas de informação, Java, aplicação distribuída, *agrobussiness*.

1. INTRODUÇÃO

Cada vez mais está se tornando comum a necessidade de interligação entre computadores, seja com a implantação de Intranets ou Extranets, sendo que o foco principal dessa interligação é fornecer recursos ao usuário que facilitem seu trabalho para que assim se obtenha aumento de produtividade.

Quando se trata de interligação de vários computadores, muitas vezes depara-se com um grande problema, que é justamente a adequação de um *software*, que seja executado em um servidor e/ou cliente, à diferentes sistemas operacionais e plataformas hoje presentes.

E o uso da linguagem Java no desenvolvimento de aplicativos Web, inclusive em *e-business*, vêm ocupando um lugar de destaque, principalmente entre grandes empresas, como a IBM (IBM, 1999).

Este projeto surgiu da necessidade de implementação de sistemas automatizados para divulgação de informações na Internet, principalmente sobre doenças da cana-de-açúcar, em um projeto financiado pelo CNPq denominado CANASIS - Ampliação e Modernização da Rede de Informática para o Desenvolvimento de sistemas computacionais a fim de implementar o manejo integrado de doenças da cana-de-açúcar. Este artigo apresenta uma ferramenta de publicação via Web, criada com a tecnologia Java, JDBC, *Servlets* e RMI.

2. ARQUITETURAS CLIENTE/SERVIDOR

Em um modelo cliente/servidor, pode-se utilizar o conjunto Java, JDBC e CORBA¹ de dois modos: em um sistema de duas ou três camadas. Nos sistemas de duas camadas os *drivers* JDBC encontram-se do lado cliente, e com uma interface apropriada, o cliente conecta-se com o servidor de banco de dados. Nos sistemas de três camadas, é usado um intermediário (*middleware*) entre o cliente e a base de dados, este é responsável por encontrar e manipular os dados. Esta camada pode ser concebida através da arquitetura CORBA (Orfali, 1998), realizando a conexão entre a interface do cliente com os *drivers* e toda a lógica de negócios, que estão no lado servidor.

Um sistema em três camadas é muito mais seguro. O cliente não tem como acessar diretamente a base de dados. Um sistema em duas camadas permite que as aplicações e/ou *applets* manipulem diretamente uma base de dados. Este modo de acesso à base de dados pode ser utilizado em situações controlado, mas não é adequada quando se precisa de segurança através da Internet / Intranet.

¹: *Common Object Request Broker Architecture*

3. A ARQUITETURA JDBC

A especificação JDBC é uma API² da linguagem Java, que permite sua integração com diferentes tipos de banco de dados, provendo suporte para comandos SQL³. Assim como ODBC (*Open Database Connectivity*), a especificação JDBC foi baseada no padrão X/Open SQL *Call Level Interface* (CLI)⁴. Com isto, a similaridade entre eles torna-se notória. A diferença mais relevante está na forma de sua implementação: um driver ODBC é dependente da plataforma para o qual está sendo desenvolvido, enquanto que o JDBC pode ser implementado independentemente de plataforma, seguindo os ideais da linguagem Java. (Orfali, 1999; Sun, 1998)

Antes da especificação JDBC, os projetistas de banco de dados precisavam cuidar da interação entre o aplicativo Java e as próprias ferramentas padrão de banco de dados. Eles geralmente obtinham sucesso executando um aplicativo auxiliar (codificados para cada tipo de ambiente), ou servidor em uma máquina acessível através da rede, que traduzia os dados entre um *applet* em Java na máquina de um usuário e o banco de dados no sistema local.

A especificação JDBC acaba com o incômodo de escrever esse aplicativo auxiliar. Os *applets* e aplicativos em Java são capazes de se comunicar com os bancos de dados necessários através da camada JDBC, e seus *drivers* podem manipular toda a conectividade do banco de dados, de forma transparente ao desenvolvedor (Ferrasa, 2000).

4 DESENVOLVIMENTO DE CGI COM USO DE *SERVLETS*

As *servlets* são programas em Java que são usados para estender as funcionalidades de um servidor Web. *Servlets* são para o servidor o que as *applets* são para o cliente: (Davidson, 1998)

- Programas em Java normais;
- Em formato *bytecode*⁵;

Uma *Servlet* difere de um *applet* por não rodar em um navegador ou com uma interface gráfica. Ao invés disso, uma *Servlet* trabalha juntamente com a *Servlet Engine*⁶ que roda no servidor Web que por sua vez trata das requisições e respostas aos clientes. [Alm99]

O programa cliente é qualquer programa capaz de fazer conexões e requisições ao servidor Web. Essa requisição é processada pela *Servlet Engine* que está rodando no servidor Web. Depois de feita a requisição do cliente para a *Servlet Engine*, ela retorna uma resposta a *Servlet*. Após isso a *Servlet* envia uma resposta dentro do protocolo HTTP para o cliente.

Funcionalmente, *Servlets* estão entre programas CGI e extensões servidoras proprietárias NSAPI⁷. Exceto em relação ao fato de que em se tratando de programas em Java, você não precisa modificar uma *Servlet* para especificar um tipo de plataforma. Além disso as *Servlets* apresentam outras vantagens, como:

- Fornecem uma maneira de gerar documentos dinâmicos que são fáceis de escrever e rápidos para executar;
- São mais rápidos que *scripts* CGI;
- Usam uma API padrão que é suportada por muitos servidores Web;

²: *Application Programatic Interface*: biblioteca de funções.

³: *Structured Query Language* : Linguagem Estruturada de Consulta.

⁴: Interface de procedimentos para acesso remoto.

⁵: Código de máquina independente gerado por um compilador Java

⁶: Ambiente escrito por uma distribuição de servidor Web de acordo com a especificação Java *Servlet* API

⁷: *Netscape Server API*

- Possuem todas as vantagens da linguagem Java, incluindo facilidade de desenvolvimento e independência de plataforma;
- Podem ter acesso a um grande número de API's disponíveis para a plataforma Java.

As *Servlets* possuem um ciclo de vida que define como a *Servlet* é carregada e inicializada, como ela recebe e responde por requisições e como ela é retirada de funcionamento. Em código, o ciclo de vida de uma *Servlet* é definido pela interface *javax.servlet.Servlet*, sendo que todas as *Servlets* devem implementar direta ou indiretamente essa interface que roda na *Servlet Engine*.

5. DESENVOLVIMENTO DE TUTORIAL ON-LINE

O Tutorial On-line é um software voltado à tecnologia Internet com o objetivo de suprir as necessidades de coleta de informações aplicadas ao Agronegócio e Ciências Ambientais, para o laboratório INFOAGRO (<http://www.infoagro.deinfo.uepg.br>), desenvolvido na Universidade Estadual de Ponta Grossa, financiado pelo CNPq, através do projeto CANASIS.

Algumas características do software:

- software distribuído;
- acessado através de um navegador internet (*browser*);
- utiliza-se da tecnologia de Java (*applets*, RMI e *servlets*);
- provê uma base consistente de dados para geração automática de páginas para *Web*;
- clientes possuem uma interface fácil e completa para o desenvolvimento de tutoriais;
- possibilita utilização de recursos de imagens;

5.1 - ESTRUTURAÇÃO DO PROJETO

O modelo de aplicação distribuída escolhido é o de três camadas, conforme a Figura 1. Isto se deve as vantagens obtidas com esta arquitetura, tais como:

- **independência de plataforma:** lógica de negócios e driver JDBC encontra-se no lado servidor;
- **eficiência:** como a lógica de negócios está toda do lado servidor, a interface cliente (*applet*), torna-se mais “leve” (menos código implementado), e com isso, mais rápida;
- **segurança:** a camada servidora (*middle tier*), oculta qualquer informação da base de dados aos clientes. Somente esta camada sabe como encontrar e manipular os dados. Esta segurança intermediária pode fornecer meios para evitar que clientes tenham acesso direto as informações importantes da base de dados.

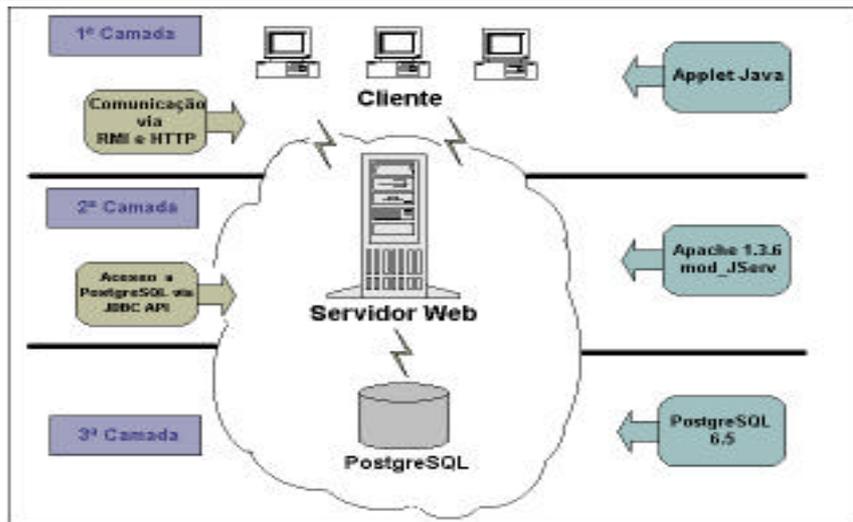


Figura 1: Estrutura do projeto

Primeira Camada

É a interface de interação com o usuário. É visualizada através de um *applet* Java, é acessada com o uso de um *browser* (*Netscape*, *Internet Explorer*), suportando *Java Plug-in* 1.1.2 ou posterior. A utilização desta tecnologia de *applets* se deve pelo grande número de recursos disponíveis para o desenvolvimento de um software interativo com o usuário.

Esta aplicação se utiliza o protocolo padrão HTTP, para a comunicação inicial entre cliente e servidor, e RMI, para comunicação entre o *applet* cliente e o servidor do tutorial. Uma posterior versão desta aplicação contará com protocolo CORBA/IIOP, visando maior flexibilidade. (Orfali, 1998)

Segunda Camada

É a camada onde se encontra toda a lógica de negócios. Nela encontra-se o servidor de aplicação (Tutorial On-line), sendo executado em JDK 1.2.2. É de responsabilidade do servidor de tutorial atender a todas as requisições dos clientes e, após receber deve validar os dados, gravando-os na base de dados em caso de sucesso. O servidor Web é o Apache 1.3.6 com módulo *Jserv*, executando sob *Linux Yellow Dog Champion Server 1.1*, provendo suporte ao uso de *Servlets*. A escolha de *Servlets* se deve ao fato da melhor interação entre aplicação Java – *Servlet* – *Applet*.

Terceira Camada

Dados são armazenados nesta camada, com uso da arquitetura JDBC para acesso à base de dados. As *Servlet's* e o Servidor de Tutorial se utilizam desta camada para armazenamento e recuperação de dados. O SGDB utilizado é o PostgreSQL 6.5.

A interface com o usuário foi criada de forma simples e concisa, para que a *applet* para a criação de tutoriais seja bem utilizada, sua aparência é apresentada na **Figura 2**:

devida a heterogeneidade de equipamentos, pois uma mesma *Servlet* pode ser executada tanto em arquiteturas do tipo Intel quanto RISC.

7. REFERÊNCIAS BIBLIOGRÁFICAS

- ORFALI, Robert; HARKEY, Dan. *Client/server programming with Java and Corba*. 2ª John Wiley & Sons Inc, 1998.
- Sun Microsystems Inc *JDBC™ 2.0 API*. 1998.
- DAVIDSON, James Duncan; AHMED, Suzanne. *Java Servlet API Specification*. Version 2.1a. Sun Microsystems Inc, 1998.
- *The Java Tutorial - A practical guide for programmers*. 2ª ed. Sun Microsystems Inc, 1999.
- MAZZOCCHI, Stefano; FUMAGALLI, Pierpaolo. *Advanced Apache JServ Techniques*. ApacheCon, 1998.
- FERRASA, Adriano. *Interfaceamento com a Internet para Divulgação da Tecnologia da Informação Aplicada ao Agronegócios e Ciências Ambientais*. Relatório Final de Iniciação Científica, UEPG, Ponta Grossa-PR, 2000.
- FOLTRAN JR, Dierone C., FERRASA, Adriano, ALMEIDA, Islenho de. *Tecnologia Java Aplicada no Desenvolvimento de Aplicativos Web Distribuídos*. In: 1º Simpósio Catarinense de Computação, UNIVALI, Itajaí-SC, 2000.
- IBM. *IBM Application Framework for e-business: Understanding technology choices for your Web applications*. URL: <http://review.software.ibm.com/developer/library/framework2/understanding.html>; 1999