

DESENVOLVENDO JAVA APPLETS PARA ENSINO DE ENGENHARIA

Sergio Scheer – scheer@cesec.ufpr.br

Manoel Theodoro Fagundes Cunha – mcunha@cesec.ufpr.br

Fernando Antonio Alves Pinto - faap@cesec.ufpr.br

Ricardo Claassen Damian - montanha@cesec.ufpr.br

João Arthur dos Santos Filho - joao@cesec.ufpr.br

Universidade Federal do Paraná, Centro de Estudos de Engenharia Civil

Caixa Postal 19011 - Centro Politécnico – Jardim das Américas

81531-990 - Curitiba - PR

***Resumo.** Neste artigo é discutido como o desenvolvimento de aplicações hipermídia para a Internet com o uso da linguagem Java pode ser útil para o ensino e a aprendizagem de conceitos de Engenharia. São descritos alguns detalhes sobre a construção de applets Java que tratam de tópicos relativos a disciplinas que tratam da disciplina específica de mecânica de estruturas. Os applets são pequenos programas em linguagem Java que são enviados pelo computador-servidor através da Internet e são instalados automaticamente no computador-cliente a partir de trechos de uma home-page da World Wide Web (WWW ou simplesmente web). Os applets podem ajudar na construção de uma boa interface para as aplicações educacionais, e são relativamente fáceis de desenvolver. As ferramentas de desenvolvimento necessárias para tratar de aplicações na Internet são a linguagem de descrição HTML, os códigos em CGI e a própria linguagem de programação Java. O artigo descreve de modo breve alguns destes conceitos fundamentais usados no desenvolvimento de pequenas aplicações de caráter educacional.*

***Palavras-chave:** Applets, Java, Internet, Ensino de Engenharia.*

1. INTRODUÇÃO

O ensino de engenharia têm sido fortemente afetado pela evolução acelerada da chamada Tecnologia da Informação e da Era do Conhecimento. Métodos e técnicas têm mudado para acompanhar esta evolução. O progresso do conhecimento humano exige que o professor se atualize constantemente e busque adaptação conseqüente a novos paradigmas de educação e de uso de tecnologia. O desenvolvimento de softwares educacionais têm sido uma prática comum nas últimas duas décadas e ganhou impulso com as interfaces tipo "windows" e mais recentemente com os desenvolvimentos da Internet e da chamada hipermídia (multimídia e hipertexto). Hoje em dia o uso intensivo de cores, gráficos e animações de alta resolução são

considerados indispensáveis a qualquer aplicação educacional. Áudio e vídeo também passam a ser usados com extrema frequência.

Atualmente a Internet permite a construção de aplicações educacionais com todos estes recursos visuais e com elevado grau de interatividade, trazendo, porém, benefícios adicionais e incomparáveis: a disseminação e a acessibilidade para o ensino de engenharia com uso das redes de computadores. Muda a forma de aprendizagem tradicional, posto que o estudante é estimulado a procurar material adicional para complementar e aprofundar a compreensão de fenômenos físicos e conceitos teóricos, de modo apropriado as suas próprias necessidades.

O uso doméstico de computadores conectados a Internet se torna uma alternativa para os laboratórios escolares que, na maior parte do tempo precisam ser compartilhados e também não podem ser atualizados adequadamente. A prontidão de vinte e quatro horas por dia é outra característica importante que permite o acesso a informação no momento mais conveniente para o usuário.

As ferramentas de desenvolvimento necessárias para tratar das aplicações na Internet desenvolvidas neste trabalho são a linguagem de descrição HTML (HyperText Markup Language) (Graham, 1995), os códigos em CGI (Common Gateway Interface) (Cunha, 1999) e a linguagem de programação Java (Naughton & Schildt, 1997). Noções destas ferramentas e em especial a linguagem Java são apresentadas e usadas na construção de alguns exemplos de aplicação na Internet através da *World Wide Web* (*WWW* ou *simplesmente web*) (Berners-Lee et al, 1994) com o uso de navegadores (ou *browsers*) padrão. Estas aplicações interativas tratam de alguns problemas comuns no ensino da Resistência dos Materiais (Mecânica das Estruturas ou Mecânica dos Sólidos) (Timoshenko, 1979) e (Nash, 1982).

2. JAVA E HTML

Java é uma linguagem de programação orientada a objetos que pode ser utilizada para a criação de dois tipos de aplicações: programas e *applets* (Naughton & Schildt, 1997). Quando utilizada no desenvolvimento de programas a linguagem Java não difere muito de outras linguagens de computador. A criação de *applets* é, na verdade, o que destaca a linguagem Java. Um *applet* é um pequeno programa enviado pelo servidor através da Internet e automaticamente instalado e executado como parte de uma página *web* (ou *home-page*) no computador do usuário ou computador cliente. *Applets* tem uma interface gráfica para o usuário que deve satisfazer regras estritas para garantir a apropriada integração com um navegador padrão.

Considere o exemplo, cujo código simplesmente mostra uma mensagem na janela do *applet*:

```
import java.awt.*;
import java.applet.*;
public class HelloApplet extends Applet {
    public void paint(Graphics g) {
        g.drawString("Hello Internet World !",20,20);
    }
}
```

Para usar este tipo de programa, é necessária a criação de uma página em linguagem HTML, que é uma linguagem de descrição baseada no uso de marcas (*tags*) para formatação de páginas na *WWW*.

Para executar um *applet* a partir de um *browser* é necessário escrever uma página *HTML* (*HyperText Markup Language*) contendo um marcador (*tag*) apropriado, conforme mostrado a seguir :

```
<HTML>
  <BODY>
    <APPLET CODE=HelloApplet.class" WIDTH=200 HEIGHT=100>
  </APPLET>
</BODY>
</HTML>
```

Java tem como uma característica importante sua portabilidade, baseada na chamada *Java Virtual Machine* (*Java VM*). Portabilidade significa independência de plataforma, i.e., um programa ou applet escrito pode ser executado em diferentes computadores com pouquíssima ou nenhuma adaptação. Ao se compilar um programa Java, geram-se instruções para a *Java VM*, que são processadas no computador cliente ou no servidor, criando instruções para o ambiente operacional específico, dando a independência de sistema.

De modo simplificado, a orientação a objetos é caracterizada pela existência do encapsulamento de dados e de métodos para modificação destes dados em pequenas unidades discretas chamadas objetos. Estes objetos somente interagem entre si mediante interfaces bem definidas, com minimização de efeitos indesejados e reutilização de objetos.

Outras vantagens são a sintaxe, mais simples que a da linguagem C, e a manipulação de entidades gráficas independentes de dispositivo.

Como desvantagem, pode-se considerar que Java é uma linguagem muito nova e, embora tenha a parte básica bem definida, as interfaces de programação da aplicação (*Application Programmer's Interface - API*) têm estado em evolução acentuada, levando à obsolescência as versões mais antigas das bibliotecas desenvolvimento (*Java Development Kit - JDK*).

3. COMMON GATEWAY INTERFACE (CGI)

Common Gateway Interface ou *CGI* é um padrão de interface entre aplicações e servidores *web* (Felton, 1997). Programas CGI são aplicações colocadas em um diretório específico de um servidor *web* e executadas por este servidor, usualmente como resposta a uma solicitação de usuário a partir de uma ação em um navegador em execução em um computador cliente. Programas CGI podem ser desenvolvidos em uma variedade de linguagens, como C/C++, Delphi, Visual Basic e Perl, e podem ser implementados em diferentes plataformas de sistema operacional, incluindo o MS-Windows e os diferentes Unix. O exemplo abaixo foi desenvolvido em linguagem C/C++ devido a sua velocidade de execução, tamanho de código executável, portabilidade e disponibilidade de compilador.

```
#include <iostream.h>
void main(void) {
    cout << " Content-type: text/html\n\n ";
    cout <<" <HTML>\n ";
    cout <<" <BODY>\n ";
    cout <<" <P>Hello Internet World! </P>\n ";
    cout <<" </BODY>\n ";
    cout <<" </HTML>\n ";
}
```

O código apresentado se refere a um programa simples que mostra uma cadeia de caracteres (*string*) na tela do computador. Como qualquer outro programa, ele pode ser executado pelo usuário e o resultado é uma cadeia de caracteres (*output*). Entretanto, o programa pode ser colocado em um servidor *web* e executado a partir de um computador cliente com um navegador padrão. Neste caso, o resultado da execução do programa é redirecionado pelo servidor para o navegador que solicitou a execução.

Uma página *web* (*home-page*) pode ser usada para executar um programa CGI ou para permitir que uma entrada de dados seja efetivada pelo usuário. Todas as informações são enviadas pelo servidor para o programa como uma cadeia contínua de caracteres que pode ser convenientemente dividida.

4. INTEGRAÇÃO DE PROGRAMAS CGI E APPLETS JAVA

Ainda que os programas CGI e *applets* Java sejam aplicações diferentes, estes programas podem interagir de várias maneiras. Programas CGI podem enviar *applets* Java como parte de uma página HTML enviada do servidor para o *browser* do usuário. Assim, o programa CGI poderia definir os parâmetros de um *applet* ou definir o próprio *applet*. Como a linguagem Java inclui a capacidade de enviar uma URL (*Universal Resource Locator*) é possível para um *applet* Java executar um programa CGI.

5. O PROCESSO DE APRENDIZAGEM E DESENVOLVIMENTO DOS APPLETS

A abordagem de desenvolvimento utilizada estabelece que cada *applet* Java é um projeto composto por três fases: a compreensão completa do problema a ser modelado, o desenvolvimento do *applet*, e a otimização do código.

Na primeira fase é proposto um problema, por exemplo, um clássico da Mecânica das Estruturas, como a determinação da linha elástica (Timoshenko, 1979) e (Nash, 1982), cujos aspectos teóricos são estudados.

O processo de desenvolvimento do *applet* envolve a criação do algoritmo que corresponde ao problema e o estudo de aspectos da linguagem Java para transcrever este algoritmo em um código que possa ser compilado. Conceitos de desenvolvimento e programação *top-down* são utilizados.

O estudo da linguagem inclui o tratamento da parte gráfica de exibição do *applet* (Amaraal, 1998) e (Knudsen, 1999) e da interface entre o usuário e o programa. A interatividade é realizada através de programação por eventos (controle do *mouse* e do teclado) (Sun Microsystems, 2000).

Para desenvolver um *applet* com código adequado é necessário o estudo de outros programas desenvolvidos por profissionais (Cunha, 1999), (Sun Microsystems, 2000), (Miller e Cooper, 1998), (Nakao et al, 1999) e (The Educational Objects Economy, 2000), sejam estes *applets* ou não. Nestes programas o código e os mecanismos de interação e de interface são analisados.

Se o *applet* estiver sendo executado corretamente, pode ser iniciado o terceiro passo que consiste em aperfeiçoar o código, testando comandos mais eficientes, distribuindo melhor o código e verificando a correção de uso das variáveis, de modo a obter o menor e mais rápido programa possível (Sun Microsystems, 2000) e (Java Source, 2000).

6. AS FERRAMENTAS

Foram desenvolvidos *applets* Java usando os softwares Java™ 2 SDK, Standard Edition versão 1.2.2 e o Java WorkShop 3.0.6 Community Edition da Sun Microsystems.

O software Java™ 2 SDK, Standard Edition, v1.2.2 é um conjunto que contém as ferramentas para desenvolver um applet ou uma aplicação em Java. Este programa é distribuído sem custos (livremente) no *site* <http://www.java.sun.com/products/jdk/1.2/>.

O software Java Workshop 3.0.6 Community Edition é um aplicativo com interface gráfica de janelas que permite editar códigos, compilar vários programas ao mesmo tempo e facilita a depuração e a correção de erros. Este software é distribuído livremente e está disponível no *site* <http://www.sun.com/workshop/java/>.

Para visualização dos applets, o *appletviewer* utilizado foi o disponível no pacote Java™ 2 SDK e o navegador selecionado foi o Microsoft Internet Explorer 5, distribuído livremente no *site* <http://www.microsoft.com/brasil/ie50/>.

7. A COLEÇÃO EVOLUTIVA DE APPLETS

Seguindo a idéia de criar um conjunto organizado e documentado de objetos de caráter educacional (The Educational Objects Economy, 2000), e usando o processo de desenvolvimento descrito, um extrato da primeira coleção de *applets* pode ser consultado em <http://www.cesec.ufpr.br/~scheer/java.html> (Scheer et al, 2000). Esta coleção de *applets* apresenta a solução interativa para alguns problemas básicos de Mecânica das Estruturas e em (Cunha,1999) aparecem algumas idéias sobre o processo evolutivo de utilização dos conceitos de programação para ambientes Internet/WWW. O desenvolvimento dos *applets* tratados nesta seção constitui parte de um processo de aprendizagem em termos de engenharia de software e de conceitos de engenharia.

7.1. Viga Simplesmente Apoiada com Carga Concentrada

Este *applet* trata de um problema que representa o caso mais simples de solicitação de uma viga. Para a solução de uma viga simplesmente apoiada com uma carga concentrada, o usuário obtém de modo automático os diagramas de esforços solicitantes: momento fletor e esforço cortante.

Neste primeiro *applet* o objetivo foi o de aprender a programação para manipulação gráfica em Java, sem oferecer interatividade. Com o intuito de auxiliar os propósitos didáticos do projeto foi apresentado junto com este primeiro *applet*, o método de seções que representa a teoria matemática que permite calcular os diagramas dos esforços solicitantes de uma viga isostática conforme mostra a Fig. 1.

O applet na Fig. 2 mostra a solução do problema onde a posição de força só pode ser modificada diretamente no código do programa.

7.2. Objetos Móveis

Tratando ainda do mesmo problema do item anterior, nesta nova versão de *applet*, a interação com o usuário é realizada através de objetos gráficos que permitem o "arraste" da carga aplicada pelo vão livre da viga com o uso do *mouse*.

É possível reconhecer o evento de *mousedrag* (arrastar o *mouse* com o botão apertado). Porém, este método é dito *deprecated*, sendo substituído por outros procedimentos em applets subsequentes, obedecendo a evolução da linguagem Java.

Os resultados providos nesta versão de *applet* são numéricos, sendo possível somente modificar os valores da força aplicada e do comprimento de viga no código de programa.

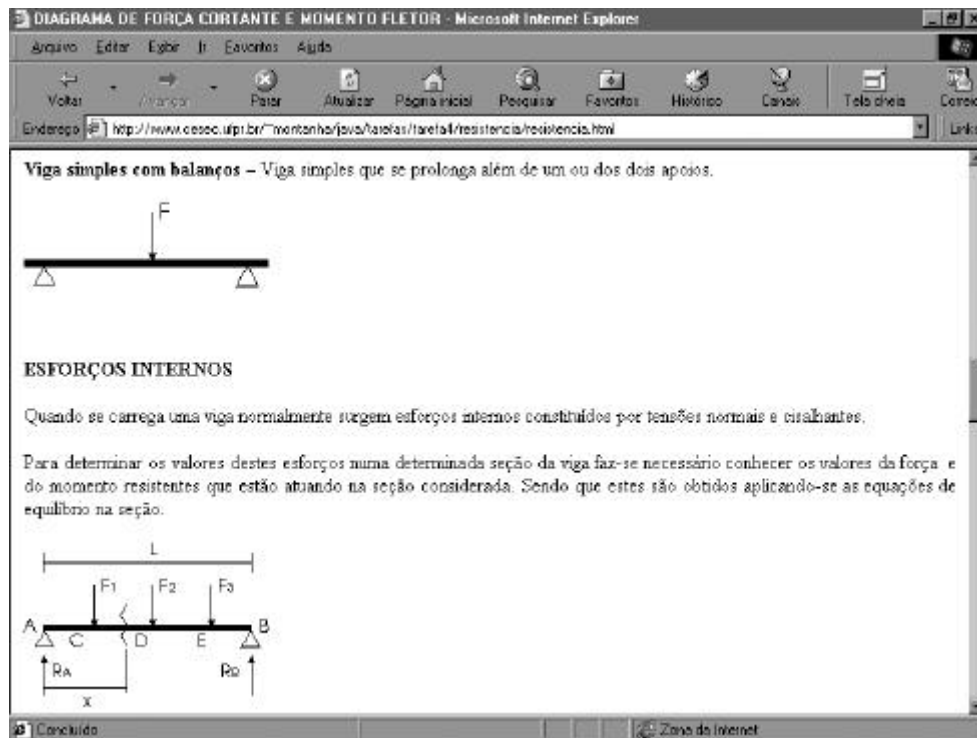


Figura 1. A Teoria do Método das Seções

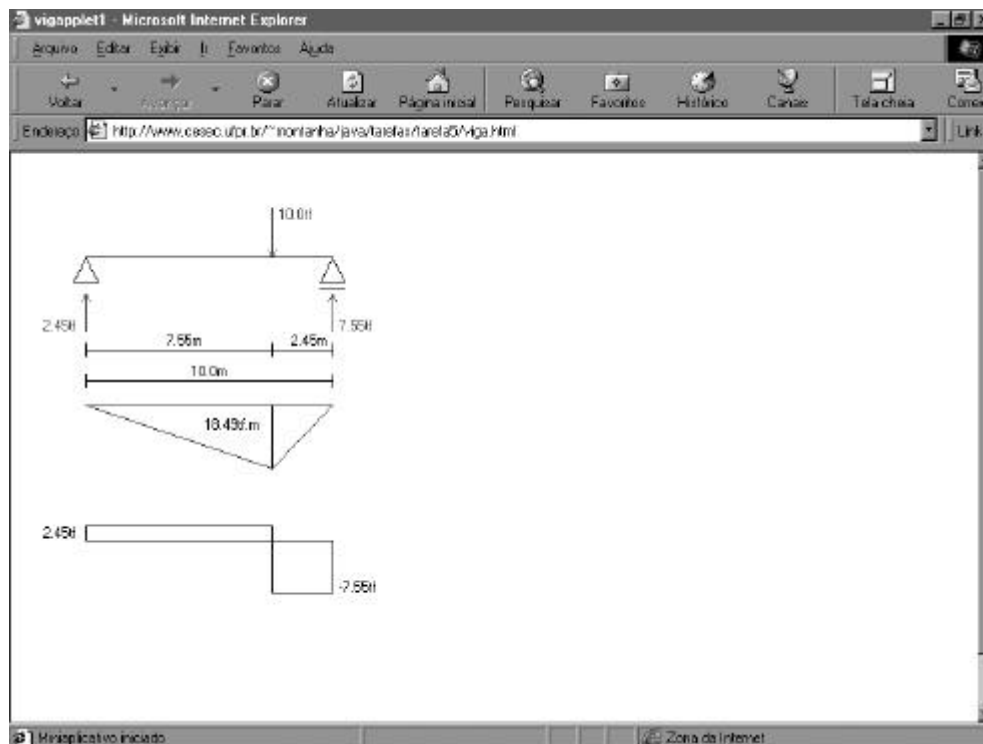


Figura 2. Viga simplesmente apoiada com carga concentrada

7.3. Viga com Carga Distribuída Parcial

Com uma versão nova de compilador Java, alguns métodos ditos *deprecated* não são mais usados e a alternativa disponível é a de *listeners*. O evento de *mousedrag* passa a ser realizado pela interface *MouseMotionListener*.

O applet abaixo permite "arrastar" as extremidades da carga distribuída, modificar o valor da carga distribuída e a posição dos apoios. Ele fornece como resultados as reações de apoio e os diagramas de força cortante e de momento fletor (Fig. 3).

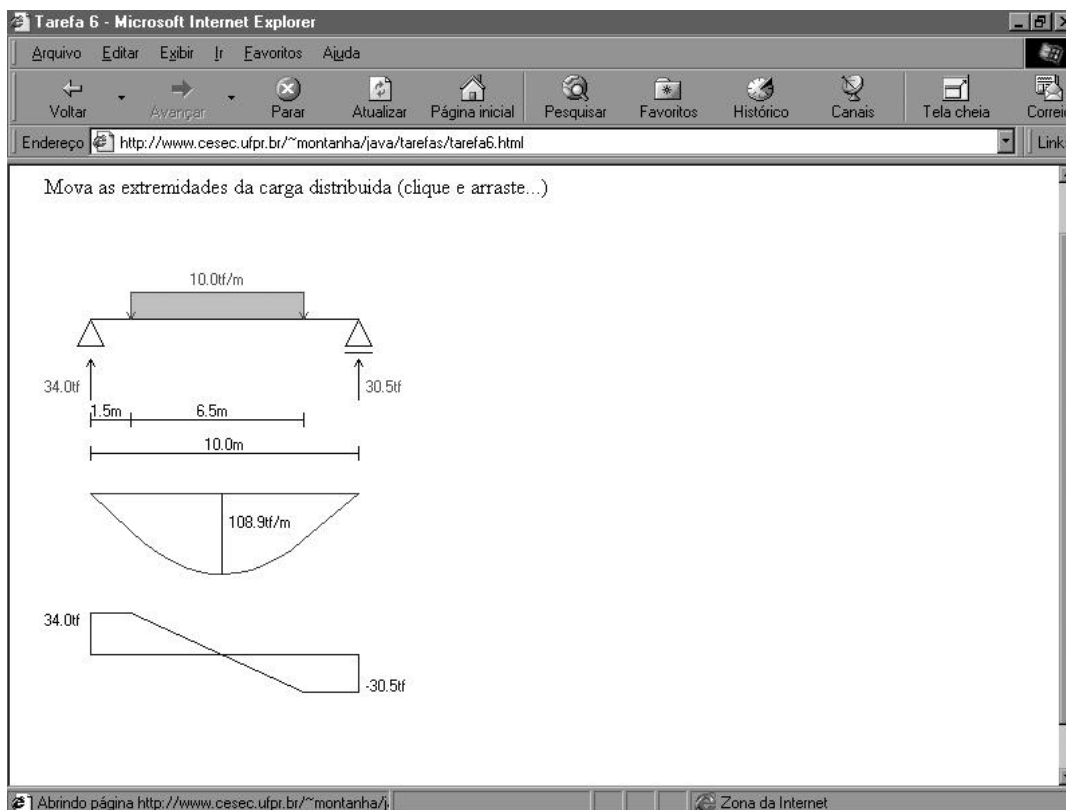


Figura 3. Viga bi-apoiada com carregamento parcialmente distribuído

7.4. Linha Elástica de uma Viga Engastada

Neste *applet* são tratados os diagramas de momento fletor e força cortante e o comportamento da linha elástica de uma viga engastada (Fig. 4). A linha elástica representa o eixo da viga e pela análise de seu comportamento é possível determinar as deformações da mesma.

Este *applet* permite que o usuário introduza os dados relacionados pelo teclado, através de campos de texto e de botões: a intensidade da carga concentrada, o comprimento do vão da viga e a sua seção transversal.

Outras versões dos *applets* com interação direta sobre os objetos gráficos na tela podem ser encontradas em Scheer et al (2000).

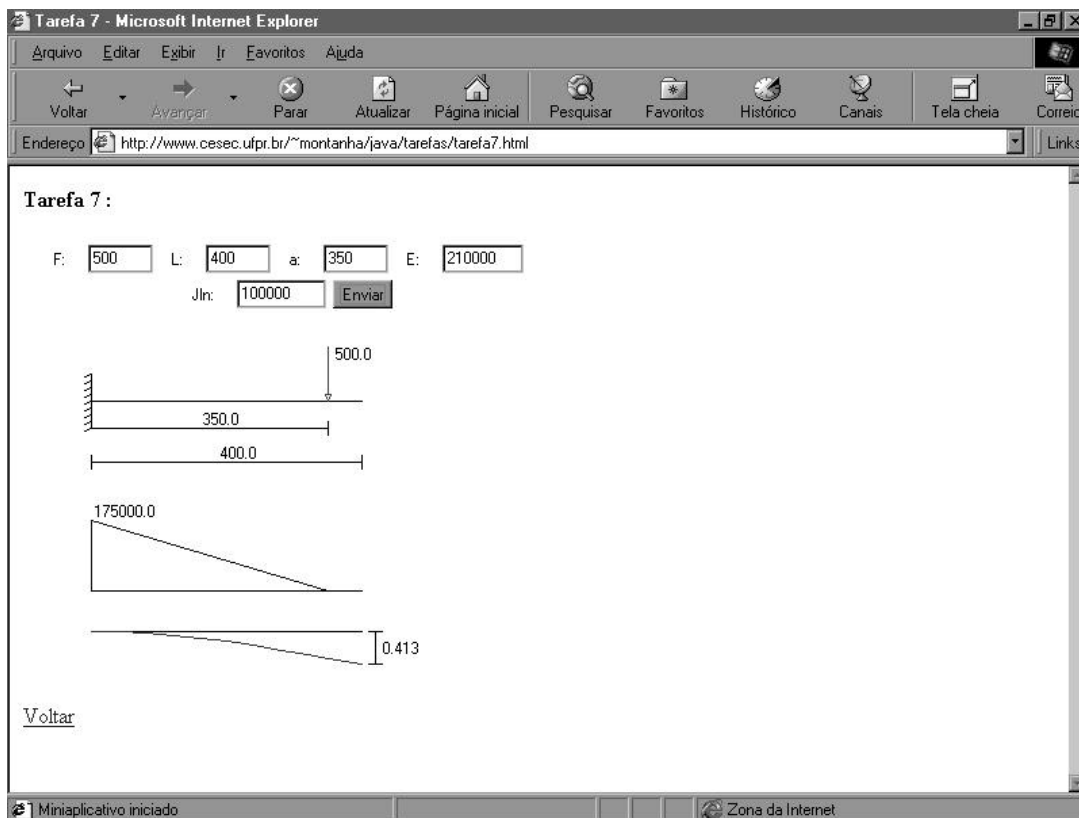


Figura 4. Viga engastada

7.5. Dimensionamento de Viga de Concreto Armado à Flexão

Este *applet* segue a normalização brasileira para estruturas de concreto armado e trata do dimensionamento de uma viga submetida a esforços de flexão. O usuário pode interagir em duas partes do *applet*: na primeira, modificando a posição da seção longitudinalmente ao longo da barra, para a qual deseja determinar o esforço de flexão; e na segunda, modificando valores relativos ao dimensionamento da viga. O usuário também pode modificar a intensidade da carga distribuída (Fig. 5) e pode visualizar graficamente e de modo imediato a variação dos valores calculados. Isto facilita a compreensão dos assuntos teóricos.

Como resultado final, é fornecida a seção da armadura de aço para suportar o momento fletor resultante devido ao carregamento. Todas as informações pertinentes são calculadas em tempo real e exibidas na tela, como por exemplo, os domínios de dimensionamento preconizados pela norma brasileira NBR-6118 (ABNT, 1995) e conforme o material correntemente em uso na disciplina de Sistemas Estruturais A do curso de Engenharia Civil da Universidade Federal do Paraná.

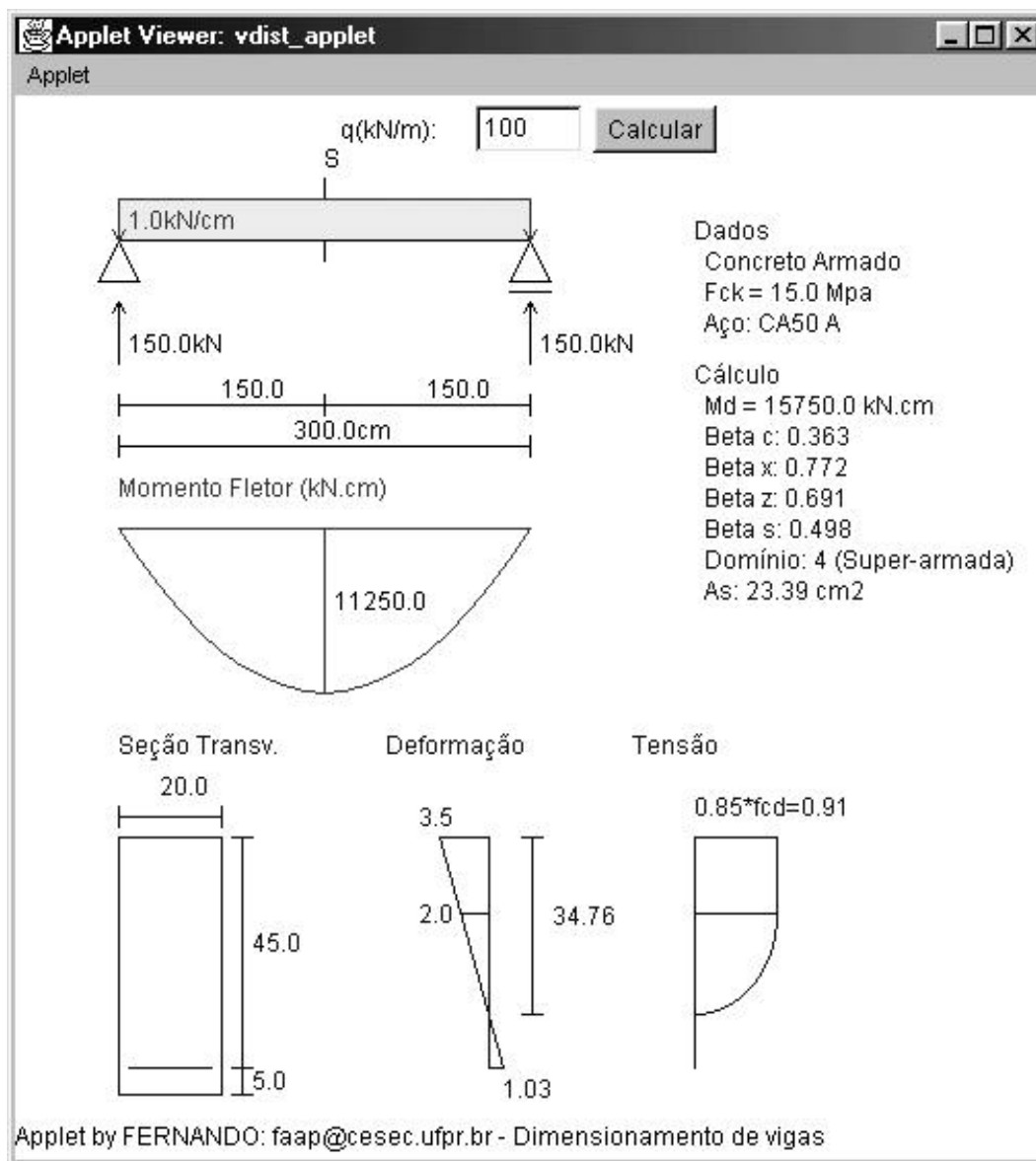


Figura 5. Dimensionamento à flexão de viga de concreto armado

8. CONCLUSÃO

Com a capacidade de controlar o mouse e eventos de teclado, os *applets* Java permitem construir interfaces interativas efetivas e que auxiliam na compreensão de conceitos teóricos.

A visualização associada a verificação de entrada de dados realizada diretamente no computador cliente pode reduzir a carga processamento no servidor, o que é bastante útil quando se trata do uso da Internet e de redes a distância.

Do ponto de vista de cálculos de engenharia, programas CGI com aplicação de applets podem executar quaisquer cálculos realizados por programas convencionais. Suas capacidades de manipulação da interação e da visualização possibilitam o desenvolvimento de programas educacionais de qualquer nível de complexidade. Além disso, acrescentam-se os benefícios de acessibilidade, prontidão e disseminação, que só uma aplicação para a Internet oferece.

REFERÊNCIAS

- Ammeraal, L.. *Computer Graphics for Java Programmers*. Wiley, 1998. 271p.
- Associação Brasileira de Normas Técnicas. *NBR-6118: Projeto de Estruturas de Concreto Armado*.
- Berners-Lee, T. et al. *The World Wide Web*. Communications ACM, v.37, n.8, p.76-82. Aug, 1994.
- Cunha, M.T.F. *A practical approach in the development of engineering applications to the internet using object oriented programming*. Dissertação de mestrado. Pós-Graduação em Métodos Numéricos em Engenharia, Univ. Federal do Paraná, Curitiba, Brasil, 1999. 64p.
- Cunha, M.T.F. <http://www.cesec.ufpr.br/~mcunha>. 25/06/2000.
- Felton, M. *CGI Internet Programming with C++ and C*. Prentice-Hall, 1997.
- Graham, I. *The HTML Sourcebook*. John Wiley, 1995.
- JavaWorld. Make Java fast. [http://www.javaworld.com / javaworld / jw-04-1997 / jw-04-optimize_p.html](http://www.javaworld.com/javaworld/jw-04-1997/jw-04-optimize_p.html). 15/03/2000.
- Knudsen, J. *JAVA 2D Graphics*. O'Reilly, 1999. 339p.
- Miller, G.R. & Cooper, S.C. *Visual Mechanics – Beams e Stress States*. Boston, PWS Publishing Company, 1998. 150p.
- Nakao, O.S.; Torres, F. F. L.; Lindenberg Neto, H. Ensinando fundamentos da resistência dos materiais com o auxílio de um programa didático de computador. Congresso Brasileiro de Ensino de Engenharia (27. : Natal : 1999). *Anais Eletrônicos*. Natal , setembro de 1999.
- Nash, W. A. *Resistência dos Materiais*. 2. ed. São Paulo : McGraw-Hill do Brasil, 1982.
- Naughton, P.; Schildt, H. *Java: The Completes Reference*. McGraw-Hill, 1997.
- Scheer, S. et al. *Desenvolvimento de Objetos Educacionais para Ensino e Aprendizagem em Engenharia*. <http://www.cesec.ufpr.br/~scheer/doe.html>. 25/06/2000.
- Sun Microsystems. *The Source for Java Technology*. <http://www.java.sun.com>, 31/01/2000.
- The Educational Objects Economy Foundation. *Java Applet Library*. <http://www.eoe.org>, 27/03/2000.
- Timoshenko, S. *Resistência dos Materiais*. 3. ed. Rio de Janeiro : Livros Técnicos e Científicos, 1979.