

Figuras 4.4 e 4.5 Combinando e compondo objetos (SANTANCHÈ, 2004)

Resumindo: objetos de aprendizagem feitos em SVG podem ser manipulados, reutilizados, aperfeiçoados, integrados em outros objetos e muito mais sem muitos problemas, pois o SVG é um formato fácil de ser implementado, tem bastante suporte e, principalmente, é livre, o que proporciona inúmeras vantagens na hora de ser integrado em diferentes plataformas e ferramentas.

Mas o uso do SVG na confecção de objetos de aprendizagem é uma especificação relativamente nova. Está sendo cada vez mais usada, mas ainda apresenta algumas poucas dificuldades.

O visualizador mais utilizado atualmente é o Adobe SVG Viewer, tanto por ser de uma das empresas mais respeitadas atualmente, a Adobe, quanto porque é um software grátis; mas ele apresenta um pequeno defeito: não funciona juntamente com algumas versões do Netscape⁸, browser muito utilizado por adeptos do software livre. Isto se torna um problema, visto que a interoperabilidade dos objetos criados em SVG deve ser mantida.

Também o caso de se ter que fazer download de um plug-in, de um visualizador para ver o SVG hoje em dia será logo aperfeiçoado: os desenvolvedores afirmam que nas próximas versões dos browsers, o suporte ao SVG já virá embutido.

Observa-se também que usuários Macintosh devem levar em consideração a visualização de alguns arquivos SVG: o browser Internet Explorer para Macintosh não suporta acesso Javascript a plug-ins, e isso faz com que os plug-ins não consigam usar a engenharia JavaScript para browsers. Por esta razão, nesta plataforma em particular, o JavaScript numa página HTML não poderá acessar qualquer SVG embutido que usufrua do JavaScript.

Mas como o objetivo do trabalho é mostrar ao leitor a integração dos objetos em um ambiente de gerenciamento e reusabilidade, além de fornecer um exemplo de um minicurso utilizando objetos reutilizáveis, as limitações desta tecnologia hoje em dia passam a se tornar invisíveis pelo simples fato de que são restrições perfeitamente solucionáveis pelos desenvolvedores. Basta que se tenha boa vontade e um olhar no sentido de que o SVG já está mais do que presente em nossas vidas, mas também faz

⁸ Na verdade, deve-se fazer algumas modificações para que se possa ver um arquivo SVG em qualquer browser Netscape. Deve-se copiar os arquivos NPSVG3.dll e NPSVG3.zip da pasta de plug-ins do Adobe Viewer para a pasta de plug-ins do Netscape.

parte de muitos serviços por nós utilizados e por muitos aplicativos Web dos quais tanto gostamos e difundimos.

Diante de tudo que foi dito sugere-se que os responsáveis pela construção dos objetos de aprendizagem, e de cursos e especificações que usam esses objetos, levem em consideração esta nova tecnologia. Ela proporcionará a confecção de materiais de alta qualidade e interativos necessários aos cursos que tenham disciplinas como geometria descritiva, desenho industrial, desenho arquitetônico, desenho de máquinas, entre outros. No futuro com certeza praticamente todos os aplicativos aceitarão o formato SVG, facilitando assim, a modelagem e construção desses objetos e a vida de quem se interessa por educação de qualidade.

Ainda, se o leitor se interessar, pode encontrar exemplos completos e bem interessantes de objetos de aprendizagem construídos em SVG em sites como o da Adobe: <http://www.adobe.com/svg/demos/>. Eis um exemplo:

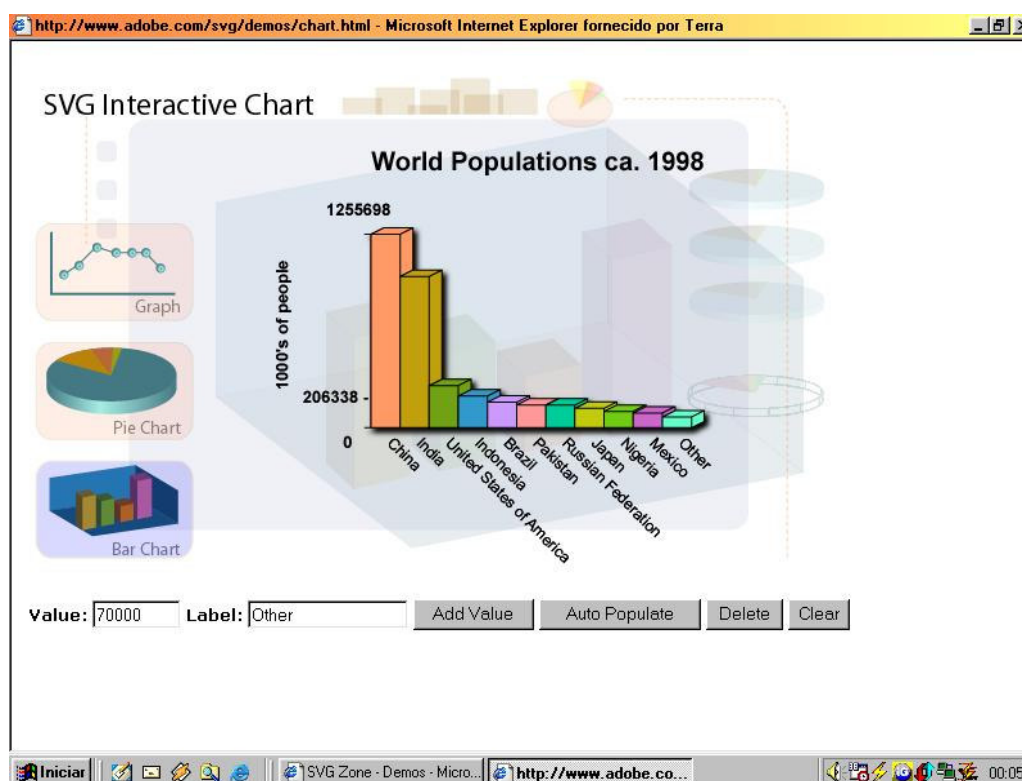


Figura 4.6 Exemplo de um gráfico interativo feito em SVG

Para que fique mais claro para o leitor a real importância do uso do SVG na aplicação de objetos de aprendizagem, foi construído um minicurso.

A idéia é que o leitor perceba o quão facilitador se torna o SVG na confecção de objetos de aprendizagem reutilizáveis, além de mostrar uma solução para melhoria do ensino/aprendizagem a distância e/ou presencial através da proposta de modelagem destes objetos, integração destes em um ambiente gerenciador buscando uma solução viável para produção e utilização de recursos computacionais, além de um ambiente via Web para visualização destes objetos.

Este minicurso é baseado em aplicações pré-definidas em diversas áreas da Matemática.

Para desenvolvimento deste curso foi usada a tecnologia SVG como base. Cada aula

do curso foi desenvolvida e integrada em um ambiente de visualização padrão, também feito em SVG, para visualização e conseqüente navegabilidade entre elas.

Cada conjunto (aula mais ambiente), foi contextualizado como sendo um só objeto (um componente de conteúdo digital), pois estes constituem um só arquivo e são estes que devem ser manipulados para fins de reuso, ou seja, se um usuário deseja usar algum objeto para algum outro fim pedagógico, ele poderá facilmente fazer isso.

Quando o usuário estiver interagindo com o curso, ele estará visualizando a aula e o ambiente dentro do mesmo contexto. Para que ele possa utilizar apenas a aula dentro de outro contexto de visualização ele deverá procurá-la dentro do arquivo SVG final apresentado, pois a aula está inserida dentro deste arquivo, o qual é constituído pela aula e pelo ambiente (formando o objeto todo)⁹.

A partir desta visão os objetos foram contextualizados dentro do padrão SCORM de empacotamento de conteúdos.

O leitor poderá entrar no ambiente do minicurso através de uma página Web e, a partir daí, estar totalmente inteirado num contexto SVG de visualização.

Este minicurso serve como um bom exemplo da interatividade e dos recursos visuais de alta qualidade que o usuário pode observar se tratando do uso do SVG como ferramenta de autoria.

<http://minerva.ufpel.edu.br/~lenapc/indexSCORM.html?StandAlone=true>: este é o endereço do minicurso.

Eis um exemplo de uma aula interativa onde, ao passar o mouse por cima de alguns elementos da tela, o usuário pode visualizar resultados imediatos:

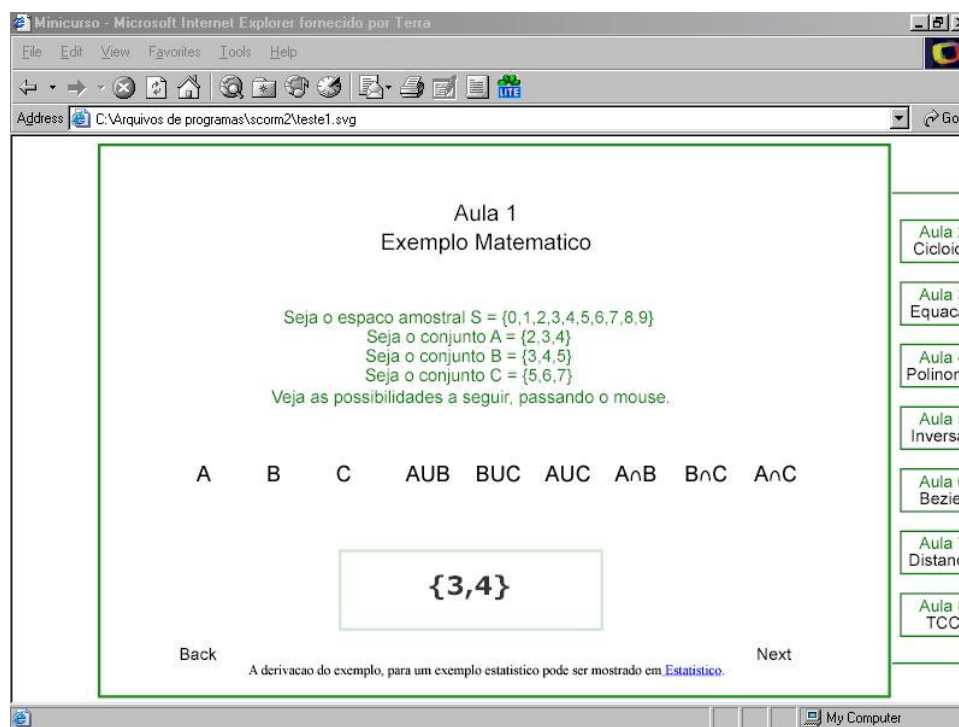


Figura 4.7 Exemplo de uma aplicação SVG interativa

⁹ A aula é um arquivo SVG que está inserido dentro de outro arquivo SVG que nada mais é do que o ambiente padrão visualizador de todas as aulas que compõem o minicurso. É este último arquivo SVG que é mostrado como resultado final.

Este exemplo pode ainda ser melhor explorado. A seguir pode-se ver o código em SVG deste exemplo:

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE svg
  PUBLIC "-//W3C//DTD SVG 1.0//EN" "http://www.w3.org/TR/2001/REC-
  SVG-20010904/DTD/svg10.dtd">

  <svg
    xmlns="http://www.w3.org/2000/svg"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:svg="http://www.w3.org/2000/svg"
    height="100%"
    width="100%"
    viewBox="0 0 1400 1000">

    <title>Minicurso</title>
    <rect x="5" y="15" width="1400" height="975"
      style="fill:white;stroke:green;stroke-width:5;
      fill-opacity:0.1;stroke-opacity:0.9"/>
    <text x="690" y="150" style="text-anchor:middle; fill:black; font-
    size:40; font-family:Arial">
      Aula 1
      <tspan x="690" dy="50">Exemplo Matematico</tspan>
    </text>
    <text x="1500" y="100" style="text-anchor:middle; font-size:30;
    font-family:Arial">
      <a xlink:href="teste2.svg">
        <tspan x="1500" dy="80" fill="green">Aula 2</tspan>
        <tspan x="1500" dy="30">Cicloide</tspan></a>
        <a xlink:href="teste4.svg">
          <tspan x="1500" dy="80" fill="green">Aula 3</tspan>
          <tspan x="1500" dy="30">Equacao</tspan></a>
          <a xlink:href="teste5.svg">
            <tspan x="1500" dy="80" fill="green">Aula 4</tspan>
            <tspan x="1500" dy="30">Polinomio</tspan></a>
            <a xlink:href="teste7.svg">
              <tspan x="1500" dy="80" fill="green">Aula 5</tspan>
              <tspan x="1500" dy="30">Inversao</tspan></a>
              <a xlink:href="teste9.svg">
                <tspan x="1500" dy="80" fill="green">Aula 6</tspan>
                <tspan x="1500" dy="30">Bezier</tspan></a>
                <a xlink:href="teste10.svg">
                  <tspan x="1500" dy="80" fill="green">Aula 7</tspan>
                  <tspan x="1500" dy="30">Distancia</tspan></a>
                  <a xlink:href="teste11.svg">
                    <tspan x="1500" dy="80" fill="green">Aula 8</tspan>
                    <tspan x="1500" dy="30">TCC</tspan></a>
                </text>
                <rect x="1425" y="148" width="150" height="75"
                  style="stroke:green; stroke-width:3; fill:none" />
                <rect x="1425" y="258" width="150" height="75"
                  style="stroke:green; stroke-width:3; fill:none" />
                <rect x="1425" y="368" width="150" height="75"
                  style="stroke:green; stroke-width:3; fill:none" />
                <rect x="1425" y="478" width="150" height="75"
                  style="stroke:green; stroke-width:3; fill:none" />
                <rect x="1425" y="588" width="150" height="75"
                  style="stroke:green; stroke-width:3; fill:none" />
                <rect x="1425" y="698" width="150" height="75"
                  style="stroke:green; stroke-width:3; fill:none" />
                <rect x="1425" y="808" width="150" height="75"
                  style="stroke:green; stroke-width:3; fill:none" />
                <rect x="1410" y="100" width="400"
```

```

        style="stroke:green; stroke-width:3; fill:none" />
<rect x="1410" y="931" width="400"
        style="stroke:green; stroke-width:3; fill:none" />

<svg width="100%" height="100%" onload="onload(evt)" x="170" y="275"
viewBox="50 170 1100 275">

    <text x="250" y="20" style="fill:green; font-family:Arial; font-
size:20; text-anchor:middle;" transform="translate(120, 0)"
startOffset="0">
        <tspan>Seja o espaço amostral  $S = \{0,1,2,3,4,5,6,7,8,9\}$ 
</tspan>
        <tspan x="250" dy="21">Seja o conjunto  $A = \{2,3,4\}$ </tspan>
        <tspan x="250" dy="22">Seja o conjunto  $B = \{3,4,5\}$ </tspan>
        <tspan x="250" dy="23">Seja o conjunto  $C = \{5,6,7\}$ </tspan>
        <tspan x="250" dy="24">Veja as possibilidades a seguir,
passando o mouse. </tspan>

    </text>
    <text id="A" x="55" y="200" style="fill:black; font-family:Arial;
font-size:25">A</text>
    <text id="B" x="135" y="200" style="fill:black; font-family:Arial;
font-size:25">B</text>
    <text id="C" x="215" y="200" style="fill:black; font-family:Arial;
font-size:25">C</text>
    <text id="AUB" x="295" y="200" style="fill:black; font-family:Arial;
font-size:25">AUB</text>
    <text id="BUC" x="375" y="200" style="fill:black; font-family:Arial;
font-size:25">BUC</text>
    <text id="AUC" x="455" y="200" style="fill:black; font-family:Arial;
font-size:25">AUC</text>
    <text id="AIB" x="535" y="200" style="fill:black; font-family:Arial;
font-size:25">A&#8745;B</text>
    <text id="BIC" x="615" y="200" style="fill:black; font-family:Arial;
font-size:25">B&#8745;C</text>
    <text id="AIC" x="695" y="200" style="fill:black; font-family:Arial;
font-size:25">A&#8745;C</text>
    <script>
        <![CDATA[
            var doc;
            function onload(evt){
                var target = evt.getTarget();
                doc = target.getOwnerDocument();
            }
            function setColor(evt, id, color) {
                // Get element over which event happened
                var element = doc.getElementById(id);
                element.setAttribute('style', 'fill:' + color);
            }
            function setLegend(legend, visibility){
                var legendNode = doc.getElementById(legend +
'Legend');
                legendNode.setAttribute('style', 'visibility:' +
visibility);
            }
        ]>
    </script>
    <g x="20" y="200" style="fill:black; font-family:Arial; font-
size:30">
        <use id="whiteS" xlink:href="#A"

```

```

onmouseover="setLegend('{2,3,4}', 'visible');"
onmouseout="setLegend('{2,3,4}', 'hidden');"
style="fill:white"/>
<use id="whiteV" xlink:href="#B"
onmouseover="setLegend('{3,4,5}', 'visible');"
onmouseout="setLegend('{3,4,5}', 'hidden');"
style="fill:white" />
<use id="whiteG" xlink:href="#C"
onmouseover="setLegend('{5,6,7}', 'visible');"
onmouseout="setLegend('{5,6,7}', 'hidden');"
style="fill:white"/>
<use id="whiteH" xlink:href="#AUB"
onmouseover="setLegend('{2,3,4,5}', 'visible');"
onmouseout="setLegend('{2,3,4,5}', 'hidden');"
style="fill:white"/>
<use id="whiteI" xlink:href="#BUC"
onmouseover="setLegend('{3,4,5,6,7}', 'visible');"
onmouseout="setLegend('{3,4,5,6,7}', 'hidden');"
style="fill:white"/>
<use id="whiteJ" xlink:href="#AUC"
onmouseover="setLegend('{2,3,4,5,6,7}', 'visible');"
onmouseout="setLegend('{2,3,4,5,6,7}', 'hidden');"
style="fill:white"/>
<use id="whiteK" xlink:href="#AIB"
onmouseover="setLegend('{3,4}', 'visible');"
onmouseout="setLegend('{3,4}', 'hidden');"
style="fill:white"/>
<use id="whiteL" xlink:href="#BIC"
onmouseover="setLegend('{5}', 'visible');"
onmouseout="setLegend('{5}', 'hidden');"
style="fill:white"/>
<use id="whiteM" xlink:href="#AIC"
onmouseover="setLegend('{}', 'visible');"
onmouseout="setLegend('{}', 'hidden');"
style="fill:white"/>

</g>
<g transform="translate(310, 330)" style="fill:rgb(00, 00, 00);
font-family:Verdana; font-size:30; font-weight:bold">
<text id="{2,3,4}Legend"
style="visibility:hidden">{2,3,4}</text>
<text id="{3,4,5}Legend"
style="visibility:hidden">{3,4,5}</text>
<text id="{5,6,7}Legend"
style="visibility:hidden">{5,6,7}</text>

</g>
<g transform="translate(300, 330)" style="fill:rgb(00, 00, 00);
font-family:Verdana; font-size:30; font-weight:bold">
<text id="{2,3,4,5}Legend"
style="visibility:hidden">{2,3,4,5}</text>
</g>
<g transform="translate(270, 330)" style="fill:rgb(00, 00, 00);
font-family:Verdana; font-size:30; font-weight:bold">
<text id="{3,4,5,6,7}Legend"
style="visibility:hidden">{3,4,5,6,7}</text>
</g>
<g transform="translate(260, 330)" style="fill:rgb(00, 00, 00);
font-family:Verdana; font-size:30; font-weight:bold">
<text id="{2,3,4,5,6,7}Legend"
style="visibility:hidden">{2,3,4,5,6,7}</text>
</g>

```

```

    <g transform="translate(320, 330)" style="fill:rgb(00, 00, 00);
font-family:Verdana; font-size:30; font-weight:bold">
        <text id="{3,4}Legend"
style="visibility:hidden">{3,4}</text>
    </g>
    <g transform="translate(340, 330)" style="fill:rgb(00, 00, 00);
font-family:Verdana; font-size:30; font-weight:bold">
        <text id="{5}Legend" style="visibility:hidden">{5}</text>
    </g>
    <g transform="translate(350, 330)" style="fill:rgb(00, 00, 00);
font-family:Verdana; font-size:30; font-weight:bold">
        <text id="{}Legend" style="visibility:hidden">{}</text>
    </g>
    <rect x="220" y="280" width="300"
height="90" style="fill:white; stroke:green;
stroke-width:3; opacity:0.2"/>

</svg>

    <text x="270" y="950" style="font-family:serif; font-size:18pt;
visibility:hidden;">
        <set attributeName="visibility" attributeType="CSS"
to="visible"
        begin="2s" dur="1s" fill="freeze"/>
        A derivacao do exemplo, para um exemplo estatistico pode ser
mostrado em<tspan
        style="fill:blue; text-decoration:underline;">
        <a xlink:href="teste13.svg">Estatistico</a></tspan>.
    </text>
    <a xlink:href="teste13.svg">
    <text x="1200" y="925" style="fill:black; font-size:30; font-
family:Arial; text-anchor:middle">
Next</text>
    </a>
    <a xlink:href="index.html">
    <text x="180" y="925" style="fill:black; font-size:30; font-
family:Arial; text-anchor:middle">
Back</text>
    </a>

</svg>

```

Em **negrito** pode-se observar os dois arquivos SVG que formam o objeto em si: mais externamente está o arquivo que define o ambiente de visualização. O arquivo SVG interno define a aula em si. Os dois complementam-se formando o objeto como um todo.

A reusabilidade pode ser facilmente implementada quando se fazem pequenas transformações neste arquivo. Este exemplo, assim como está, pode ser usado para fins matemáticos.

Clicando no link “Estatístico” em azul abaixo na tela, pode-se chegar à derivação deste exemplo para fins estatísticos. Para que ele possa ser usado para estes fins, as seguintes incrementações no arquivo devem ser feitas: onde está o comando para aparecer o texto correspondente a cada vez que se passa o mouse sobre um evento (A, B, C, AUB e etc), deve-se acrescentar a seguinte linha de código: `<tspan x="-35" dy="45" font-size="25">P=3/10=30%</tspan>`. Neste caso, esta linha de código foi inserida para aqueles eventos que mostram três elementos como resultado.

A seguir, em **negrito**, pode-se ver a implementação total resultante para adaptação do arquivo para fins estatísticos, em cada caso (cada evento):

```

<g transform="translate(310, 315)" style="fill:rgb(00, 00, 00); font-
family:Verdana; font-size:30; font-weight:bold">
  <text
    id="{2,3,4}Legend"
    style="visibility:hidden">{2,3,4}

    <tspan
      x="-35"
      dy="45"
      font-
      size="25">P=3/10=30%</tspan>
      EVENTO "A"

    </text>
    <text
      id="{3,4,5}Legend"
      style="visibility:hidden">{3,4,5}

    <tspan
      x="-35"
      dy="45"
      font-
      size="25">P=3/10=30%</tspan>
      EVENTO "B"

    </text>
    <text
      id="{5,6,7}Legend"
      style="visibility:hidden">{5,6,7}

    <tspan
      x="-35"
      dy="45"
      font-
      size="25">P=3/10=30%</tspan>
      EVENTO "C"

    </text>
  </g>
<g transform="translate(320, 310)" style="fill:rgb(00, 00, 00); font-
family:Verdana; font-size:20; font-weight:bold">
  <text
    id="{2,3,4,5}Legend"
    style="visibility:hidden">{2,3,4,5}

    <tspan
      x="-40"
      dy="45"
      font-
      size="25">P=4/10=40%</tspan>
      EVENTO "AUB"

    </text>
  </g>
<g transform="translate(310, 310)" style="fill:rgb(00, 00, 00); font-
family:Verdana; font-size:20; font-weight:bold">
  <text
    id="{3,4,5,6,7}Legend"
    style="visibility:hidden">{3,4,5,6,7}

    <tspan
      x="-35"
      dy="45"
      font-
      size="25">P=5/10=50%</tspan>
      EVENTO "BUC"

    </text>
  </g>
<g transform="translate(300, 310)" style="fill:rgb(00, 00, 00); font-
family:Verdana; font-size:20; font-weight:bold">
  <text
    id="{2,3,4,5,6,7}Legend"
    style="visibility:hidden">{2,3,4,5,6,7}

    <tspan
      x="-20"
      dy="45"
      font-
      size="25">P=6/10=60%</tspan>
      EVENTO "AUC"

    </text>
  </g>
<g transform="translate(335, 310)" style="fill:rgb(00, 00, 00); font-
family:Verdana; font-size:20; font-weight:bold">
  <text id="{3,4}Legend" style="visibility:hidden">{3,4}

    <tspan
      x="-60"
      dy="45"
      font-
      size="25">P=2/10=20%</tspan>
      EVENTO "A∩B"

```

```

</text>
</g>
<g transform="translate(345, 310)" style="fill:rgb(00, 00, 00); font-
family:Verdana; font-size:20; font-weight:bold">
  <text id="{5}Legend" style="visibility:hidden">{5}

      <tspan          x="-70"          dy="45"          font-
size="25">P=1/10=10%</tspan>          EVENTO "B∩C"

  </text>
</g>
<g transform="translate(350, 310)" style="fill:rgb(00, 00, 00); font-
family:Verdana; font-size:16; font-weight:bold">
  <text id="{ }Legend" style="visibility:hidden">{ }

      <tspan          x="-125"          dy="29">Eventos          mutuamente
exclusivos</tspan>          EVENTO "A ∩ C"

  </text>
</g>

```

Há ainda mais outros exemplos de aulas interativas e de aulas animadas que vêm a reforçar toda a conceitualização de aulas interessantes, não apenas lineares, onde o aluno possa descobrir o conhecimento, aprender da maneira que achar melhor, com o tempo que achar necessário dentro de um escopo de e-learning.

Eis outro exemplo que mostra a animação como característica fundamental:

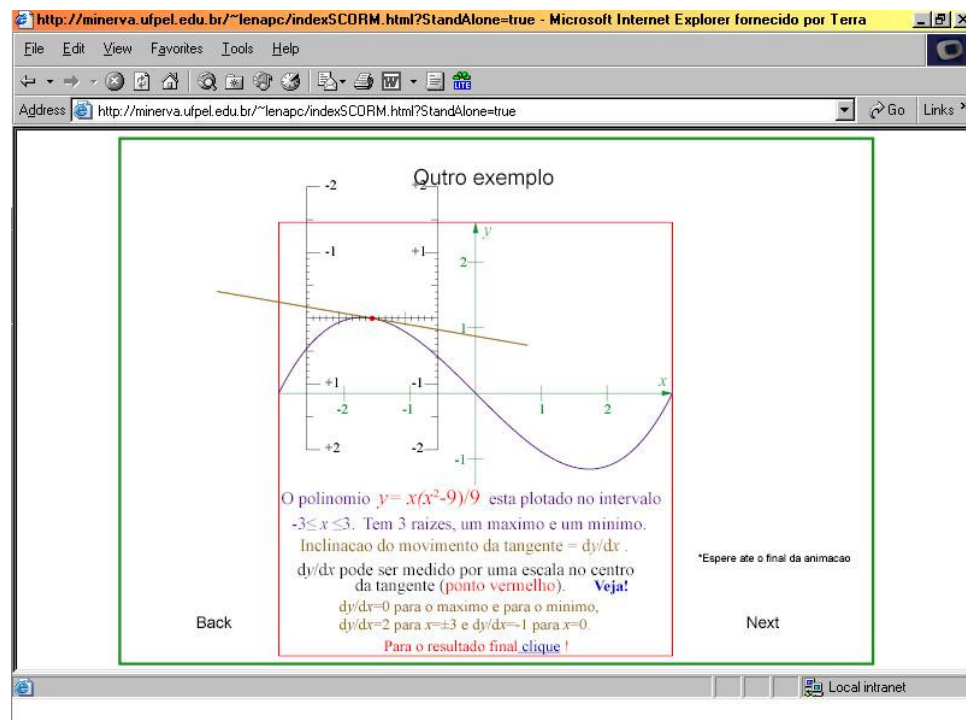


Figura 4.8 Exemplo de um recurso pedagógico desenvolvido em SVG

Há ainda um outro exemplo que pode ser mostrado na figura a seguir. Este é um exemplo de aula multimídia onde um som foi adicionado ao arquivo, mostrando como uma aula em SVG pode ser bastante atrativa. É fato que o som do arquivo não pode ser

ouvido aqui neste trabalho, mas o exemplo dado serve para o leitor se encorajar no uso da multimídia em aulas informatizadas. Eis o exemplo:

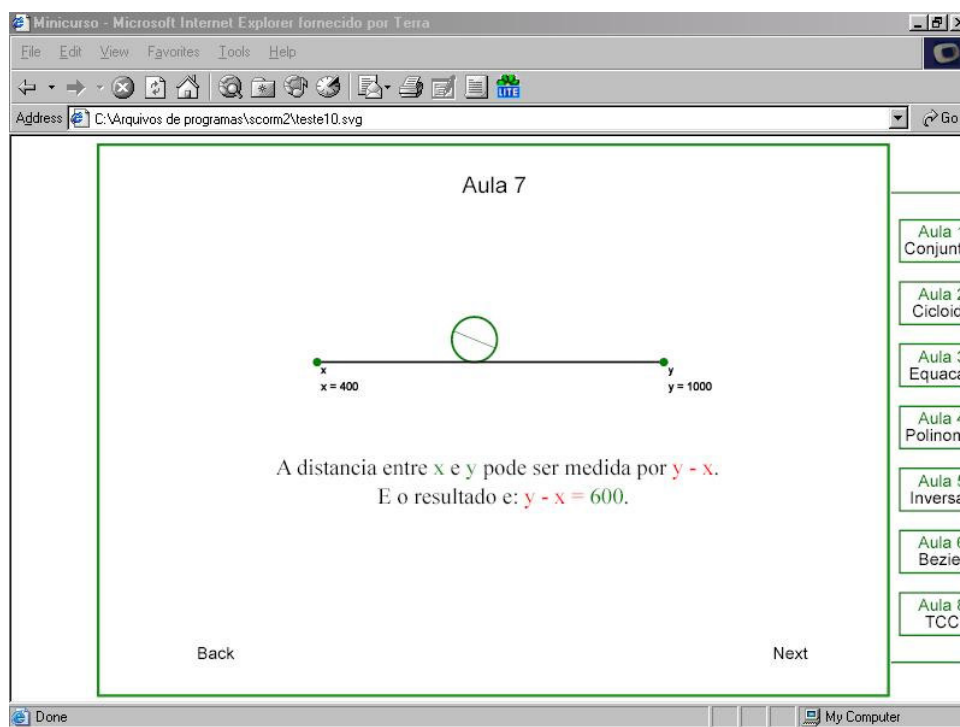


Figura 4.9 Exemplo de aula multimídia

A confecção deste minicurso serviu para que fosse usada a teoria de padronização de objetos. Como cada arquivo (ambiente + aula) foi considerado um objeto, cada um deles foi inserido dentro do contexto SCORM de padronização de objetos.

A seguir estão todos os passos para que isso pudesse ser feito:

1 – Através do site <http://www.scorm.com/SCORMProducts.php> pode-se fazer download grátis (baixar) do RSSCA, um programa que contém as APIs necessárias para inserir todos os objetos no padrão SCORM. O nome do arquivo SimpleContentAPI.zip.

2 – Logo após extrair os arquivos em alguma pasta que contenha todos os objetos que serão usados, deve-se editar o arquivo chamado “indexSCORM”. Lá deve-se fazer seguintes alterações:

2.1 – Na linha número 166: `<frame name="content" src="#CONTENT_URL#">`, deve-se mudar as palavras `#CONTENT_URL#` para o nome do caminho que servirá como início do conteúdo que se deseja mostrar. Por exemplo: como o arquivo inicial usado para navegação entre os objetos era uma página HTML chamada `index.html`, este foi colocado no lugar de `#CONTENT_URL#`. A modificação encontra-se destacada em negrito:

```
<HTML>
<HEAD>
<title></title>
<script>
var blnFoundAPI = false;
var blnCalledFinish = false;
var blnStandAlone = false;
var findAPIAttempts = 0;
```

```

var objAPI;
var SCORM_FALSE = "false";
function SCORM_Start(){
    var strStandAlone;
    var strResult;
    strStandAlone = GetQueryStringValue("StandAlone",
window.location.search);
    if (ConvertStringToBoolean(strStandAlone)){
        blnStandAlone = true;
    }
    if (! blnStandAlone){

        objAPI = getAPI();

        if ((objAPI == null) || (typeof(objAPI) ==
"undefined")){
            alert("SCORM API could not be found. Results
may not record properly.");
            return false;
        }
        else{
            blnFoundAPI = true;
        }
        strResult = objAPI.LMSInitialize("");
        if (strResult == SCORM_FALSE){
            alert("Content failed to establish
communication with the LMS. Results may not record properly.");
            return false;
        }
    }
    return true;
}
function SCORM_Finish(){
    if ((! blnStandAlone) && (blnFoundAPI) && (!blnCalledFinish)){
        blnCalledFinish = true;
        strResult = objAPI.LMSFinish("");
        if (strResult == SCORM_FALSE){
            alert("Content failed to communicate your
results to the LMS.");
            return false;
        }
    }
    return true;
}
function GetQueryStringValue(strElement, strQueryString){
    var aryPairs;
    var i;
    var intEqualPos;
    var strArg = "";
    var strValue = "";
    //get rid of the leading "?"
    strQueryString = strQueryString.substring(1);
    //split into name/value pairs
    aryPairs = strQueryString.split("&");
    strElement = strElement.toLowerCase();
    //search each querystring value and return the first match
    for (i=0; i < aryPairs.length; i++){
        intEqualPos = aryPairs[i].indexOf('=');
        if (intEqualPos != -1){
            strArg = aryPairs[i].substring(0,intEqualPos);
            if (EqualsIgnoreCase(strArg, strElement)){

```

```

                                strValue =
aryPairs[i].substring(intEqualPos+1);
                                return new String(strValue);
                                }
                                }
                                //if we didn't find a match, return an empty string
                                return "";
                                }
                                function ConvertStringToBoolean(str){
                                    var intTemp;
                                    if (EqualsIgnoreCase(str, "true") || EqualsIgnoreCase(str,
                                "t")){
                                        return true;
                                    }
                                    else{
                                        intTemp = parseInt(str);
                                        if (intTemp == 1){
                                            return true;
                                        }
                                        else{
                                            return false;
                                        }
                                    }
                                }
                                function EqualsIgnoreCase(str1, str2){
                                    str1 = new String(str1);
                                    str2 = new String(str2);
                                    return (str1.toLowerCase() == str2.toLowerCase());
                                }
                                //code from the SCORM specification - www.adlnet.org
                                function findAPI(win)
                                {
                                    while ((win.API == null) && (win.parent != null))
                                    {
                                        findAPITries++;
                                        // Note: 7 is an arbitrary number, but should be more than
sufficient
                                        if (findAPITries > 7)
                                        {
                                            return null;
                                        }
                                        win = win.parent;
                                    }
                                    return win.API;
                                }
                                function getAPI()
                                {
                                    var theAPI = findAPI(window);
                                    if ((theAPI == null) && (window.opener != null) &&
                                (typeof(window.opener) != "undefined"))
                                    {
                                        theAPI = findAPI(window.opener);
                                    }
                                    return theAPI
                                }
                                </SCRIPT>
                                </HEAD>
                                <frameset rows="100%" onload="SCORM_Start();"
onbeforeunload="SCORM_Finish()" onunload="SCORM_Finish()">

```

```

<frame name="content" src="index.html">

</frameset>
<noframes>
  Your browser must be able to view frames for this content to
  display.
</noframes>
</HTML>

```

3 – O próximo passo é editar o arquivo “imsmanifest.xml”, abrindo-o em qualquer editor xml (bloco de notas por exemplo):

3.1 – Na linha 115 do arquivo encontra-se o seguinte: `<!--***** FILES *****-->` `<file href="indexSCORM.html"/>`. Cada objeto deve ser listado como um elemento. Todos o objetos devem ser listados, até os arquivos de mídia que são utilizados, como os arquivos mp3 por exemplo.

3.2 – Os metadados descrevem todo o conteúdo encapsulado. Podem ser encontrados logo após cada dado mostrado como `<!--*****1*****-->`, por exemplo. Neste exemplo são definidos 10 tipos de metadados. A seguir pode ser visualizado o código do arquivo com as modificações melhor explicadas:

```

<?xml version="1.0"?>
<manifest identifier="RSSCA" version="1.2"
  xmlns="http://www.imsproject.org/xsd/imscp_rootv1p1p2"
  xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_rootv1p2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.imsproject.org/xsd/imscp_rootv1p1p2
imscp_rootv1p1p2.xsd

http://www.imsglobal.org/xsd/imsmd_rootv1p2p1 imsm�_rootv1p2p1.xsd

http://www.adlnet.org/xsd/adlcp_rootv1p2 adlcp_rootv1p2.xsd">
  <metadata/>
  <organizations default="B0">
    <organization identifier="B0">

<!--***** 1 *****--> NOME DADO AO CONTEÚDO
<title>Minicurso de Matematica Aplicada</title>
<item identifier="i1" identifierref="r1" isvisible="true">

<!--***** 1 *****--> REPETIÇÃO DO NOME DADO AO CONTEÚDO
<title>Minicurso de Matemática Aplicada</title>

    </item>
  </organization>
</organizations>
<resources>
  <resource identifier="r1" type="webcontent"
adlcp:scormtype="sco" href="indexSCORM.html">
    <metadata>
      <schema>ADL SCORM</schema>
      <schemaversion>1.2</schemaversion>
      <lom
xmlns="http://www.imsglobal.org/xsd/imsmd_rootv1p2p1">
      <general>

<!--***** 1 *****--> REPETIÇÃO DO NOME DADO AO CONTEÚDO
<title>

```

```

<langstring>Minicurso de Matematica Aplicada</langstring>
</title>

<!--***** 2 *****--> CATEGORIZAÇÃO DO CONTEÚDO
<catalogentry>
<catalog>Maths</catalog> NOME
<entry>
<langstring>1</langstring> ENTRADA
</entry>
</catalogentry>
<language>por</language> LINGUA=PORTUGUÊS

<!--***** 3 *****--> DESCRIÇÃO TEXTUAL DO CONTEÚDO
<description>
<langstring>Here you can find a Maths minicourse</langstring>
</description>

<!--***** 4 *****--> PALAVRAS CHAVES
<keyword>
<langstring>Maths</langstring> 1ª PALAVRA CHAVE
</keyword>
<keyword>
<langstring>Interaction</langstring> 2ª PALAVRA CHAVE
</keyword>
<keyword>
<langstring>SVG</langstring> 3ª PALAVRA CHAVE
</keyword>

    </general>
    <lifecycle>

<!--***** 5 *****--> VERSÃO DO CONTEÚDO
<version>
<langstring>1</langstring> VERSÃO 1
</version>
<status>
<source>
<langstring xml:lang="x-none">LOMv1.0</langstring>
</source>
<value>
<langstring xml:lang="x-none">Final</langstring>
</value>
</status>
    </lifecycle>
<metametadata>
<metadatascheme>ADL SCORM 1.2</metadatascheme>
</metametadata>
    <technical>

<!--***** 6 *****--> FORMATO DOS ARQUIVOS
<format>SVG/html</format> FORMATOS=SVG E HTML
<location>indexSCORM.html</location>
</technical>
    <rights>

<!--***** 7 *****--> CUSTO (SE O CONTEÚDO É PAGO)
<cost>
<source>

```

```

<langstring xml:lang="x-none">LOMv1.0</langstring>
</source>
<value>                                NO=CONTEÚDO NÃO PAGO
<langstring xml:lang="x-none">no</langstring> </value>
</cost>

<!--***** 8 *****-->  REGISTRO E OUTRAS RESTRIÇÕES
<copyrightandotherrestrictions>
<source>
<langstring xml:lang="x-none">LOMv1.0</langstring>
</source>
<value>                                NO=NÃO REGISTRADO
<langstring xml:lang="x-none">no</langstring>
</value>
</copyrightandotherrestrictions>
</rights>
<classification>
<purpose>
<source>
<langstring xml:lang="x-none">LOMv1.0</langstring>
</source>
<value>                                CONTEÚDO DE OBJETIVO EDUCACIONAL
<langstring xml:lang="x-none">Educational Objective</langstring>
</value>
</purpose>

<!--***** 9 *****-->  CLASSIFICAÇÃO (DESCRIÇÃO)
<description>
<langstring>Here you can find a Maths minicourse</langstring>
</description>

<!--***** 10 *****-->  CLASSIFICAÇÃO (PALAVRAS CHAVES)
<keyword>
<langstring>Maths</langstring>
</keyword>
<keyword>
<langstring>Interaction</langstring>
</keyword>
<keyword>
<langstring>SVG</langstring>
</keyword>
    </classification>
  </lom>
</metadata>

<!--***** FILES *****-->
<file href="index.html"/>
<file href="teste.svg"/>
<file href="teste1.svg"/>
<file href="teste2.svg"/>
<file href="teste3.svg"/>
<file href="teste4.svg"/>
<file href="teste5.svg"/>
<file href="teste6.svg"/>
<file href="teste7.svg"/>
<file href="teste8.svg"/>
<file href="teste9.svg"/>
<file href="links.html"/>
<file href="teste10.svg"/>

```

**AQUI OS
OBJETOS
SÃO
LISTADOS**

```

        <file href="rev3.mp3"/>
        <file href="teste11.svg"/>
        <file href="teste12.svg"/>
        <file href="teste13.svg"/>

        <!--include one file element for each file in
the content.  Ex: <file href="index.html"/>
        <file href="wordfiles/mycontent.doc"/>
        <file href="course/module01.html"/>
        <file href="quiz/quiz01.html"/>
        -->
        </resource>
    </resources>
</manifest>

```

Esta padronização através do uso do SCORM incrementa a idéia prevista para aquisição de conteúdo através do uso de objetos de aprendizagem.

Este minicurso é efetivamente um simples exemplo, mas é apropriado para ilustrar tudo aquilo de adicional que os conceitos de reusabilidade dos objetos de aprendizagem vem trazendo para o e-learning, além de propor o uso do SVG como tecnologia de autoria o que baseia mais ainda o todo que já foi mencionado durante o trabalho.

O projeto deste minicurso vem ao encontro da proposta fundamental do trabalho de buscar uma melhoria no processo de aprendizagem a distância (ou até mesmo presencial com o uso de computador), através de uma solução viável de modelagem de objetos de aprendizagem e de um exemplo de um ambiente visualizador destes objetos.