

**ESTUDO EXPLORATÓRIO SOBRE POTENCIALIDADES DAS TECNOLOGIAS DE
INFORMAÇÃO DE PADRÃO ABERTO NA CONSTRUÇÃO DE SIMULADORES
PARA O DESENVOLVIMENTO DE CONCEITOS FÍSICOS**

**EXPLORATORY STUDY ON POTENTIALITIES OF OPEN TECHNOLOGIES IN THE
CONSTRUCTION OF SIMULATORS AIMING THE PHYSICAL CONCEPTS
DEVELOPMENT**

**Vania Elisabeth Barlette¹
Paulo Henrique Guadagnini²**

¹Centro Universitário Franciscano, Santa Maria, RS, vebarlette@globo.com

²Universidade Estadual do Rio Grande do Sul, Porto Alegre, RS, paulo-guadagnini@uergs.edu.br

Resumo

Este trabalho diz respeito a um estudo exploratório sobre potencialidades do uso combinado da tecnologia de informação de padrão aberto para gráficos vetoriais SVG (Scalable Vector Graphics) com tecnologias de padrão aberto XML (Extensible Markup Language), XHTML (Extensible Hypertext Markup Language) e ECMAScript na construção de simuladores visando o desenvolvimento de conceitos físicos. Essas tecnologias abertas são utilizadas neste trabalho para operacionalizar a estratégia de simulação interativa e viabilizar a interação com situações de ensino, na perspectiva de desenvolvimento conceitual referenciada pela teoria dos campos conceituais de Vergnaud. Alguns exemplos de simulação interativa usando as tecnologias abertas citadas são apresentados, e sua aplicabilidade no desenvolvimento de conceitos físicos é discutida.

Palavras-chave: Simulação interativa, Tecnologias de informação de padrão aberto, Simulador, Campos conceituais de Vergnaud, Ensino de física.

Abstract

This work concerns an exploratory study on potentialities of the combined use of the SVG (Scalable Vector Graphics) open standard technology with the XML (Extensible Markup Language), XHTML (Extensible Hypertext Markup Language) e ECMAScript open standard technologies in the construction of simulators aiming the physical concepts development. Such open technologies are used in this work to operationalize the interactive simulation strategy and to allow the interaction with teaching situations, on the perspective of the conceptual development in the framework of the Vergnaud's conceptual fields theory. Some examples of interactive simulation using the already mentioned open technologies are presented, and its applicability for the physical concepts development is discussed.

Keywords: Interactive simulation, Open standard technologies, Simulator, Vergnaud's conceptual fields, Physics education.

INTRODUÇÃO

As possibilidades educacionais criadas pelos ambientes virtuais na Web (World Wide Web) têm se expandido nos últimos anos, desde a distribuição de material instrucional de cursos presenciais e teleconferências, até cursos de graduação e de pós-graduação à distância. Atualmente, os ambientes virtuais de aprendizagem são desenvolvidos utilizando um conjunto de tecnologias de padrão aberto e de padrão proprietário. Especificamente, no desenvolvimento de ambientes de aprendizagem baseados em simulação, a tecnologia proprietária Flash vem sendo largamente utilizada. No entanto, novas tecnologias exclusivamente de padrão aberto vem se apresentando com grande potencial de difusão na Web, tais como construção de ambientes gráficos interativos com aplicação em cartografia (Neumann & Winter, 2001), apresentação de dados em tempo real, e construção de sites. Na área educacional, entretanto, ainda são poucos os estudos sobre aplicações de tecnologias exclusivamente de padrão aberto para o desenvolvimento de ambientes de aprendizagem baseados em simulação interativa (Bogaard et al., 2004).

A nova tecnologia SVG (Scalable Vector Graphics) é uma tecnologia de padrão aberto que apresenta tendência crescente de utilização na Web, mas ainda pouco difundida como tecnologia educacional na área de educação em ciências muito embora com potencialidades inovadoras para a educação nesta área. Essa tecnologia, em conjunto com outras tecnologias de padrão aberto, tem potencial de utilização para o desenvolvimento de simuladores (*softwares*) que auxiliam o aluno na compreensão e na solução de problemas complexos por meio da estratégia de simulação de experiência (Neuman, 2005).

Atentos a esta tendência, neste trabalho reportamos um estudo exploratório sobre potencialidades de uso das tecnologias de informação de padrão aberto SVG, XML (Extensible Markup Language), XHTML (Extensible Hypertext Markup Language) e ECMAScript na construção de simuladores para o desenvolvimento de conceitos físicos. Adotamos a hipótese de trabalho de que o desenvolvimento de conceitos físicos é potencializado quando o aluno interage com situações de ensino por meio de simulação de experiência. Essa hipótese encontra suporte na filosofia cognitivista de aprendizagem e na teoria dos campos conceituais de Vergnaud (Vergnaud, 1997).

FUNDAMENTAÇÃO TEÓRICA

A Formação de Conceitos segundo a Teoria dos Campos Conceituais de Vergnaud

A formação de um conceito, segundo Vergnaud (Vergnaud, 1997), se dá quando dominamos três conjuntos relacionados a este conceito: o conjunto das representações simbólicas aceitas pela comunidade partícipe do conceito; o conjunto dos invariantes operatórios ou propriedades do conceito; e, o conjunto das situações diversas que conferem sentido ao conceito.

O desenvolvimento dos domínios das representações, dos invariantes operatórios e das situações diversas do conceito ocorrem simultaneamente ao desenvolvimento das relações ou esquemas que conseguimos estabelecer entre eles. Para Vergnaud, um conceito faz sentido para nós quando relacionado a outros conceitos, na forma de uma rede de conceitos. Esses argumentos conduziram ao conceito de campos conceituais por Vergnaud, em que (a) um só tipo de situações não é suficiente para a formação de um conceito, mas sim uma variedade de situações com as quais nos confrontamos, (b) e progressivamente dominamos no âmbito dos três domínios como um processo de longo prazo; e, (c) que a análise de uma situação requer mais de um conceito.

Simulação Apoiada por Computador para a Aprendizagem de Conceitos

Por simulação apoiada por computador entende-se uma representação dinâmica dos aspectos essenciais de uma situação (Anido et al., 2005). A simulação apoiada por computador, no âmbito das ciências naturais, tem como base um modelo científico de representação dos aspectos essenciais de uma situação do mundo que se deseja explorar, em que o ambiente computacional é o meio no qual aspectos do modelo podem ser capturados na forma de dados ou mesmo na forma de imagens dinâmicas, e não somente o curso de eventos pode ser alterado como também outros aspectos do modelo podem emergir quando parâmetros do modelo são modificados (Anido et. al., 2005).

Aplicada ao ensino das ciências, pode-se dizer que a simulação apoiada por computador adquire uma dimensão de estratégia que proporciona aprendizagem ativa, a qual envolve a interação do aluno com elementos incorporados nas situações de ensino (*reality on*), agregando significados aos mesmos por meio da observação e manipulação (*hands on*), análise e síntese (*minds on*) (Nascimento, 2005; Campbell, 2005). A uma simulação com tais características passaremos a nos referir por simulação computacional interativa, ou simplesmente simulação interativa. A interação entre o aluno e o ambiente computacional geralmente ocorre por meio da entrada de dados pelo teclado e *mouse*, utilizando elementos da interface gráfica, tais como caixas de texto, *clicks* e movimentos com o *mouse*, listas selecionáveis, caixas de diálogo, entre outros.

Em uma perspectiva construtivista, os significados são adquiridos pelo aluno através da sua interação com as situações de ensino proporcionadas pela simulação. Nessa perspectiva, à medida que situações diversas e mais complexas são simuladas pelo aluno, melhor ele estará preparado para compreender e resolver problemas em situações reais (Campbell, 2005). O educador deve, no entanto, ou ao menos deveria, deixar claro ao aluno quais aspectos da realidade estão sendo simulados, e que a simulação, ela mesma, não é a realidade, mas sim uma simplificação da realidade.

Ressalta-se, ainda, que a estratégia da simulação tem a vantagem de extrair de uma situação da realidade complexa apenas aspectos selecionados para integrar à situação de ensino, compatibilizando o grau de dificuldade inerente na situação de ensino ao nível de desenvolvimento do aluno. Com isso, é possível evitar aspectos que dificultariam o aprendizado em um momento específico. Ela tem, portanto, um caráter seletivo, desejável pedagogicamente (Anido et al., 2005).

Tecnologias de Informação para a Operacionalização de Simulações Interativas

A operacionalização de simulações por meio do computador requer o uso de um conjunto de tecnologias para incorporar as atividades interativas em um *software*, chamado simulador. Usualmente, o simulador é disponibilizado em um ambiente de rede, permitindo que haja um ambiente cooperativo de trabalho. Essa possibilidade é de grande importância em ambientes de aprendizagem distribuídos na Web que dão suporte a cursos presenciais e à distância por meio da estratégia de aprendizagem colaborativa.

Quanto à arquitetura de um simulador, podemos ter: (a) processamento local em um computador cliente, que no contexto de ensino, se refere ao equipamento utilizado pelo aluno-usuário; e, (b) processamento em um computador servidor, o qual é acessado remotamente através de uma interface de rede. Para a escolha da arquitetura de um simulador, deve-se levar em conta um conjunto de fatores, tais como *performance* da rede e do computador, tamanho dos arquivos necessários para o simulador, tempo necessário para transferir estes arquivos ao computador cliente, e necessidade de atualização periódica do simulador.

Existem casos em que a arquitetura baseada em um servidor que executa a simulação é mais vantajosa para aplicações educacionais, como por exemplo, em casos em que o tempo necessário para transferir dados para a execução do simulador no computador cliente é grande, ou quando o simulador exige uma capacidade computacional consideravelmente superior àquela disponível no computador cliente. Simuladores com processamento em servidor podem ser desenvolvidos utilizando uma grande variedade de tecnologias, tais como, C, C++, PHP (PHP: Hypertext Preprocessor), e ASP (Active Server Pages).

Para os casos de processamento local, citamos as seguintes vantagens em aplicações educacionais: a minimização do tráfego de dados na rede, a atualização automática do simulador que será executado no computador cliente, e o pequeno tempo de resposta aos comandos do usuário. Como exemplos de simuladores de arquitetura baseada em processamento local, podemos citar aqueles desenvolvidos utilizando as tecnologias Java e ECMAScript. Atualmente, a tecnologia Java é a mais difundida no desenvolvimento de simuladores baseados em interatividade, especialmente na forma de *applets* que executam em navegadores Web. Outra possibilidade para essa finalidade é a utilização da linguagem ECMAScript em conjunto com outras tecnologias. A linguagem ECMAScript possui a vantagem de ser mais simples para o desenvolvimento de simuladores comparada à linguagem Java, além de ser amplamente difundida em aplicações Web (Neuman, 2005).

Para a implementação de simuladores, deve-se preferir que os mesmos incorporem as características de portabilidade, disponibilidade, interatividade e flexibilidade. A portabilidade de um simulador permite o seu funcionamento nas plataformas computacionais mais utilizadas (Windows, Unix e Macintosh), e a sua disponibilidade permite atingir o maior número de usuários sem a necessidade de distribuição física através de CD-ROM ou disquete. A portabilidade e a disponibilidade são características inerentes aos simuladores cuja arquitetura é baseada no cliente (processamento local). Quanto à interatividade e à flexibilidade, a simulação de experiência interativa é conseguida por intermédio de *softwares (scripts)* os quais implementam as simulações, com entrada de parâmetros, possibilidade de análise gráfica, e entrada de novos parâmetros para nova execução e análise pela intervenção do usuário. Neste trabalho, essas características foram implementadas em uma linguagem de *scripts* que possuem a flexibilidade necessária. A linguagem ECMAScript satisfaz esses requisitos e foi preferida para o desenvolvimento deste trabalho.

A tecnologia XML não se refere a um *software* mas sim a um formato de arquivo texto baseado em *tags* (marcadores). Essa tecnologia forma uma base tecnológica para padrões na Web que estão sendo utilizados atualmente, e para padrões que ainda estão por ser implementados na Web. O formato XML foi projetado para a descrição de dados, permitindo que a informação possa ser estruturada, armazenada e transferida de modo padronizado. Por outro lado, o formato HTML tem como finalidade principal a apresentação de dados, cuja preocupação nesta apresentação é a aparência que os dados assumirão frente ao usuário. O formato XML é independente de plataforma e do tipo de aplicativo, não tendo um conjunto pré-definido de *tags* (Eisenberg, 2002). Esse padrão é recomendado pelo W3C (World Wide Web Consortium), que é um consórcio internacional formado por universidades, instituições e empresas, cuja finalidade é o desenvolvimento de padrões para a Web (W3C, 2005). Várias tecnologias foram desenvolvidas com base no formato XML, cada uma relacionada com uma área específica de interesse, tais como o CML (Chemical Markup Language), SMIL (Synchronized Multimedia Integration Language), XHTML, SVG, entre outros. O padrão XHTML é uma reformulação do HTML 4.01 que incorpora o padrão XML, e cuja finalidade é substituir o formato HTML. O XHTML é um padrão recomendado pelo W3C.

Dentre as várias tecnologias disponíveis para a apresentação de gráficos vetoriais bidimensionais interativos, a tecnologia Flash é a mais utilizada com suporte em praticamente todos os navegadores atuais. Entretanto, uma questão que deve ser considerada é que esta

tecnologia é proprietária, ou seja, é dependente de uma única empresa (Adobe, anteriormente Macromedia). Ressalta-se que o interesse em gráficos vetoriais para a apresentação na Web é devido ao tamanho reduzido dos arquivos que contêm comandos e coordenadas, ao invés de um mapa de *bits* utilizados em arquivos de imagens (gif, jpeg, etc.). Devido a necessidade de estabelecer um padrão aberto de gráficos vetoriais para a Web, o W3C propôs uma recomendação do padrão SVG, o qual possui recursos iguais ou superiores ao formato Flash e com tendência crescente de utilização na Web. O padrão SVG possui vantagens em relação ao padrão Flash: (a) por ser um padrão aberto, baseado no modelo XML, garante-se que existirá um suporte a longo prazo para a utilização em ambientes Web, bem como a compatibilidade com outros formatos de dados: e, (b) o tipo de formato de arquivo SVG é do tipo texto e não binário, como ocorre no formato Flash e que limita as possibilidades de edição.

METODOLOGIA

Os procedimentos metodológicos, as estratégias para implementar os procedimentos, e os recursos didáticos/tecnológicos necessários à construção de simuladores são descritos abaixo. O procedimento 1 foi estabelecido visando uma perspectiva de aprendizagem de conceitos físicos referenciada pela teoria dos campos conceituais de Vergnaud. Os procedimentos 2 e 3 foram estabelecidos para explorar as potencialidades das tecnologias de informação de padrão aberto, visando a aprendizagem apoiada por computador.

Procedimento 1: Seleção e organização das situações de ensino. Com relação à seleção das situações de ensino, a estratégia utilizada foi a de priorizar situações cujos conceitos são de difícil compreensão, seja por limitações no imaginário do aluno, seja por limitações da sua experiência, ou mesmo, limitações devido à complexidade matemática para a representação do modelo. As situações selecionadas foram organizadas segundo uma estratégia que contempla grupos de conceitos relacionados entre si. Neste trabalho, os conceitos contemplados foram campo (vetorial e escalar), força e potencial.

Procedimento 2: Análise exploratória das tecnologias de informação de padrão aberto. A estratégia utilizada na identificação das potencialidades das tecnologias abertas para a construção de simuladores para o ensino das ciências físicas baseou-se nas categorias de análise portabilidade, disponibilidade, facilidade de codificação, interatividade, reusabilidade e flexibilidade de um simulador. Os recursos tecnológicos elegidos por contemplar tais características foram SVG, XML, XHTML e ECMAScript.

Procedimento 3: Implementação dos simuladores, utilizando as tecnologias elegidas. Etapas utilizadas: (a) implementação de um documento XHTML contendo hipertexto, elementos interativos para entrada de dados (caixas de texto e botões) e referência aos objetos embutidos (arquivos externos) no formato SVG; (b) no arquivo SVG: implementação de elementos gráficos (círculos, vetores, gráficos, etc.) reutilizáveis, descritos no formato SVG; (c) implementação de *scripts* na linguagem ECMAScript, os quais se comunicam com elementos gráficos SVG e elementos descritos no documento XHTML, executando ações (simulações) em resposta à interação com o aluno (em geral, por meio de teclado e *mouse*). O modelo é implementado nos *scripts*, e estes, em resposta à interação com o aluno, alteram parâmetros do modelo. A estratégia utilizada para operacionalizar as etapas (a), (b) e (c) foi baseada em simulação interativa mediada por computador, a qual permite a interação do aluno com as situações em um contexto de ensino. Os recursos tecnológicos utilizados para implementar os simuladores foram *software (browser)*, módulos de *software (plugin)* e internet.

POTENCIALIDADES DAS TECNOLOGIAS SVG/XML/XHTML/ECMASCRIT NA CONSTRUÇÃO DE SIMULADORES

Um conjunto representativo de recursos computacionais das tecnologias SVG/XML/XHTML/ECMAScript foi utilizado para avaliar (a) a aplicabilidade destas tecnologias na construção de simuladores e (b) a interação do usuário com o simulador.

Análise da Aplicabilidade das Tecnologias SVG/XML/XHTML/ECMAScript na Construção de Simuladores

Foram utilizadas as categoriais de análise: portabilidade, disponibilidade, facilidade de condificação, interatividade, reusabilidade e flexibilidade.

Os simuladores construídos com as tecnologias citadas apresentam excelente portabilidade e disponibilidade, que são intrínscas às tecnologias utilizadas. Atualmente, qualquer navegador processa corretamente códigos utilizando as tecnologias XML, XHTML e ECMAScript. Quanto à tecnologia SVG, a disponibilidade de *plugins* para processar arquivos no formato SVG ainda não é ampla para todos os navegadores devido ao fato desta tecnologia ser recente. Entretanto, novas soluções de *softwares* para a integração da tecnologia SVG aos navegadores estão aparecendo continuamente, podendo-se citar o caso do projeto Mozilla que recentemente incorporou uma implementação nativa do SVG no navegador Firefox. Para o caso do navegador Internet Explorer, que é utilizado pelo maior número de usuários atualmente, existe a disponibilidade de um *plugin* distribuído gratuitamente pela Adobe.

Quanto à facilidade de codificação, avaliamos de modo positivo o conjunto de tecnologias citadas. A linguagem ECMAScript tem codificação simplificada em comparação com outras linguagens de programação, tais como o Java e o C++, apesar de manter um nível de recursos que avaliamos como suficiente para o desenvolvimento de praticamente qualquer tipo de simulador que não exija recursos computacionais elevados em termos de tempo computacional para execução de cálculos numéricos. No entanto, verificamos que, para o tipo de simulador que se pretende (desenvolvimento de conceitos), a linguagem ECMAScript pode ser utilizada sem comprometimento do tempo de resposta do simulador às ações do usuário.

Os simuladores por nós desenvolvidos e apresentados neste trabalho incorporam um conjunto de módulos de *softwares* escritos na linguagem ECMAScript que são reutilizáveis, e que, portanto, permitem o desenvolvimento de novos simuladores, minimizando a quantidade de condificação necessária. Isso também ocorreu quanto aos objetos gráficos SVG. Além disso, em um mesmo simulador, um determinado objeto gráfico SVG, tal como a representação de um vetor na forma de uma linha e uma seta, pode ser reutilizável, criando-se quantas instâncias de tal objeto quantas forem necessárias, minimizando a complexidade e a quantidade de memória necessárias.

O fato das tecnologias utilizadas serem de padrão aberto confere flexibilidade aos projetos desenvolvidos com tais tecnologias, não sendo necessária a utilização de *softwares* específicos, e geralmente proprietários, para a edição de arquivos. No caso de arquivos SVG, é possível a edição utilizando qualquer editor de texto, ao contrário de arquivos em formato proprietário, tal como o Flash. A utilização de tecnologias abertas também é vantajosa no sentido de prolongar o tempo de vida dos códigos de simuladores desenvolvidos, uma vez que tais tecnologias não são vinculadas a nenhuma empresa específica, mas sim suportadas por uma comunidade composta por um grande número de usuários, desenvolvedores e empresas.

Análise da Interação do Usuário com o Simulador

Foram utilizadas duas categoriais de análise: consistência da interface gráfica com o sistema operacional e tempo de resposta do sistema com as ações do usuário.

A interface gráfica não deve ser um recurso dificultador mas sim facilitador da interação do usuário com o simulador. Utilizando o padrão XHTML na construção dos simuladores, foi possível introduzir elementos de interface gráfica com o usuário que possuem consistência com a interface gráfica utilizada no sistema operacional em que o simulador é executado. Isso permite uma utilização do simulador por parte do usuário mais intuitiva, uma vez que usualmente este aluno já possui familiaridade com a interface gráfica do sistema operacional.

O tempo de resposta é dependente da capacidade do processamento do computador cliente que roda o simulador, bem como da complexidade do simulador em questão. Entretanto, com a utilização do padrão SVG na apresentação gráfica, constatamos que foi possível a atualização em tempo real da janela gráfica dos simuladores desenvolvidos, utilizando computadores com capacidade média de processamento disponíveis atualmente. O padrão SVG é centrado na definição de objetos gráficos reutilizáveis, os quais podem ser selecionados individualmente pela ação do usuário, gerando eventos que podem ser processados no programa ECMAScript. Isso permite expandir o grau de interatividade do simulador com o usuário. Nos simuladores desenvolvidos, o tipo de interação mais sensível ao tempo de resposta à ação do usuário foi a seleção e arrasto de objetos gráficos, ação esta que altera as coordenadas espaciais dos objetos gráficos na janela do simulador. Outros elementos de interação que são mobilizados pela ação do usuário com o simulador, e que configura em uma boa interação do usuário com a interface gráfica, são a entrada de dados numéricos e a confirmação através de botões. Os recursos tecnológicos utilizados para implementar os simuladores foram *software (browser)*, módulos de *software (plugin)* e internet.

EXEMPLOS DE SITUAÇÕES DE ENSINO BASEADAS EM SIMULAÇÃO INTERATIVA APOIADA POR COMPUTADOR COM RECURSOS DAS TECNOLOGIAS SVG/XML/XHTML/ECMASCRIPT

As Figuras 1, 2 e 3 apresentam imagens estáticas bidimensionais de simulações de duas situações de ensino. Para as telas apresentadas nessas figuras, foi utilizado o navegador Internet Explorer juntamente com o *plugin* Adobe SVG Viewer. As situações apresentadas nas Figuras 1 e 2 foram construídas para mobilizar no aluno invariantes e representações simbólicas dos conceitos campo vetorial, campo elétrico e força elétrica. Na Figura 3, os conceitos envolvidos são campo escalar e campo de temperaturas.

Exemplo 1: Simulação de uma Configuração de Cargas Elétricas Pontuais

A Figura 1 apresenta uma tela do navegador contendo o simulador de uma configuração de duas cargas elétricas pontuais de módulos iguais, representadas pelos círculos azul e vermelho. Na Figura 2, é apresentada uma nova janela do simulador da configuração de cargas em que o valor da carga representada pelo círculo azul foi alterado pela ação do usuário, e por conseguinte, alterada a intensidade do campo elétrico (gradiente de cores) gerado por esta nova carga representada pelo círculo azul. A imagem apresentada nesta mesma Figura 2 também mostra que novas alterações foram feitas pelo usuário, uma vez que as cargas estão em nova posição na janela do simulador, o que reflete um novo conjunto de coordenadas espaciais para as cargas. Essa dupla modificação conduz a novos valores para o módulo, a direção e o sentido do vetor força elétrica mútua entre as cargas, representado por diferente comprimento do segmento do vetor e diferente orientação do vetor na janela do simulador em relação a situação antes de ocorrer a interação do usuário com o simulador (Figura 1).

Para que essas ações fossem praticadas pelo usuário, a seguinte operacionalização foi necessária: em um documento XHTML foi embutido um arquivo no formato SVG contendo os elementos gráficos e *scripts* na linguagem ECMAScript. À esquerda da janela gráfica, nas Figuras 1 e 2, os elementos gráficos SVG (círculos) foram disponibilizados para serem

manipulados dinamicamente pela ação do usuário, e em tempo real, por meio da seleção e arrasto com o *mouse*. Isso permite que coordenadas espaciais das cargas pontuais sejam alteradas e visualizados os novos resultados nesta mesma janela. Utilizamos como elementos gráficos: (a) círculos, para representar cargas pontuais; (b) vetores (construídos com retas e polígonos), para representar o módulo, a direção e o sentido do vetor força que atua sobre uma carga devido a outra; (c) gradiente de cores, para representar a intensidade do campo elétrico gerado por uma carga pontual em um certo ponto do espaço em torno da mesma; e, (d) texto, para a apresentação de dados numéricos calculados em tempo real, à medida que o usuário interage com o simulador.

O simulador foi construído de modo que a interatividade permite que o usuário altere: (a) o valor de cada uma das cargas, que pode ser feito utilizando a caixa de texto à direita da janela gráfica e introduzindo manualmente o novo valor através do teclado e *mouse*, seguido do *click* no botão “Alterar dados”; e, (b) as coordenadas espaciais das cargas, com a atualização em tempo real dos vetores força, bem como dos dados numéricos apresentados na parte inferior da janela do simulador.

A Tabela 1 sintetiza os elementos de interação presentes no simulador, a ação esperada do usuário, a resposta do simulador à ação do usuário, as representações simbólicas e os invariantes operatórios para este exemplo.

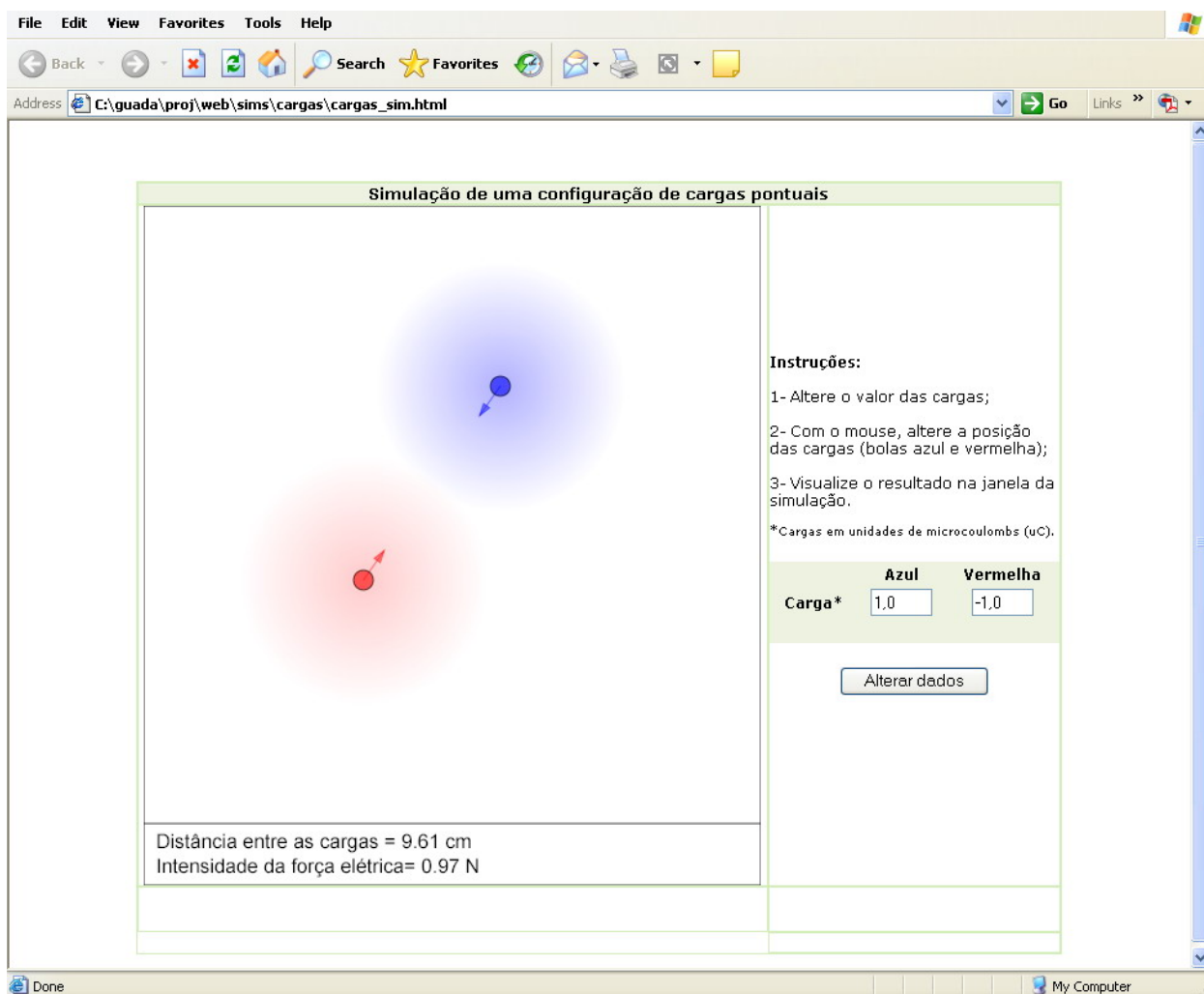


Figura 1: Janela do simulador de uma configuração de cargas elétricas pontuais de módulos iguais.

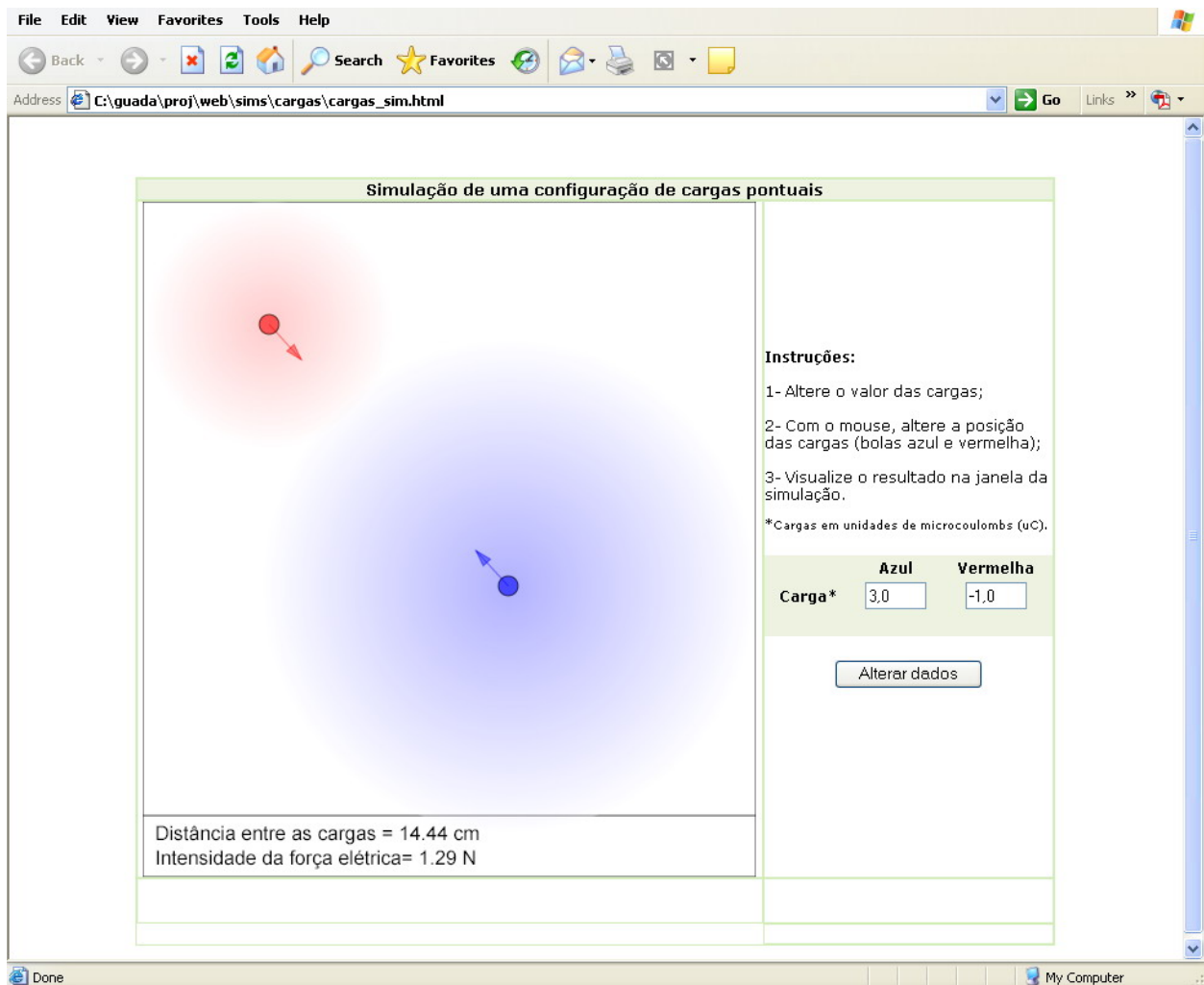


Figura 2: Janela do simulador de uma configuração de cargas elétricas pontuais de módulos diferentes.

Exemplo 2: Simulação de um Campo de Temperaturas

A Figura 3 apresenta uma simulação de um campo de temperaturas bidimensional radial em torno de um material metálico em que o usuário pode obter o valor da temperatura em um dado ponto do plano da janela do simulador posicionando o cursor do *mouse* sobre o plano neste ponto. Nesse caso, o valor da temperatura é indicada na parte inferior da janela da simulação, e é atualizada em tempo real a cada nova intervenção do usuário.

Neste simulador, foi utilizado um gradiente de cores representando a função que descreve a variação radial da temperatura. Outros campos de temperatura arbitrários podem ser implementados de maneira semelhante, fornecendo representações de intensidades de campos escalares baseados em gradiente de cor. No caso específico do simulador da Figura 3, altas temperaturas são representadas por cores de tonalidades vermelhas à alaranjadas; e baixas temperaturas, por cores de tonalidades verdes à azuis.

Tabela 1: Elementos de interação presentes no simulador, ação esperada do usuário, resposta do simulador à ação do usuário, representações simbólicas e invariantes operatórios para o simulador do Exemplo 1.

Elemento de interação (“objeto”)	Ação do usuário	Resposta do simulador	Representações simbólicas	Invariantes operatórios
Círculos azul e vermelho	Seleção do objeto e arrasto com o <i>mouse</i>	Altera coordenadas espaciais das cargas	Círculos azul e vermelho	Cargas elétricas pontuais
Caixa de texto (<i>mouse</i> e teclado) e botão de confirmação (“Alterar dados” com <i>mouse</i>)	Altera o valor da carga geradora do campo	Altera a representação do gradiente de cores	Gradiente de cores	Intensidade do vetor campo elétrico
Círculos azul e vermelho	Seleção do objeto e arrasto com o <i>mouse</i>	Altera o comprimento do segmento representativo do vetor; Altera coordenadas espaciais da carga arrastada; Recalcula a distância de separação entre as cargas	Comprimento do segmento representativo do vetor	Intensidade do vetor força elétrica mútua
Círculos azul e vermelho	Seleção do objeto e arrasto com o <i>mouse</i>	Altera a orientação espacial do segmento representativo do vetor	Orientação espacial do segmento representativo do vetor	Direção do vetor força elétrica mútua
Círculos azul e vermelho	Seleção do objeto e arrasto com o <i>mouse</i>	Altera o sentido da seta na extremidade do segmento representativo do vetor	Sentido da seta localizada na extremidade do segmento representativo do vetor	Sentido do vetor força elétrica mútua

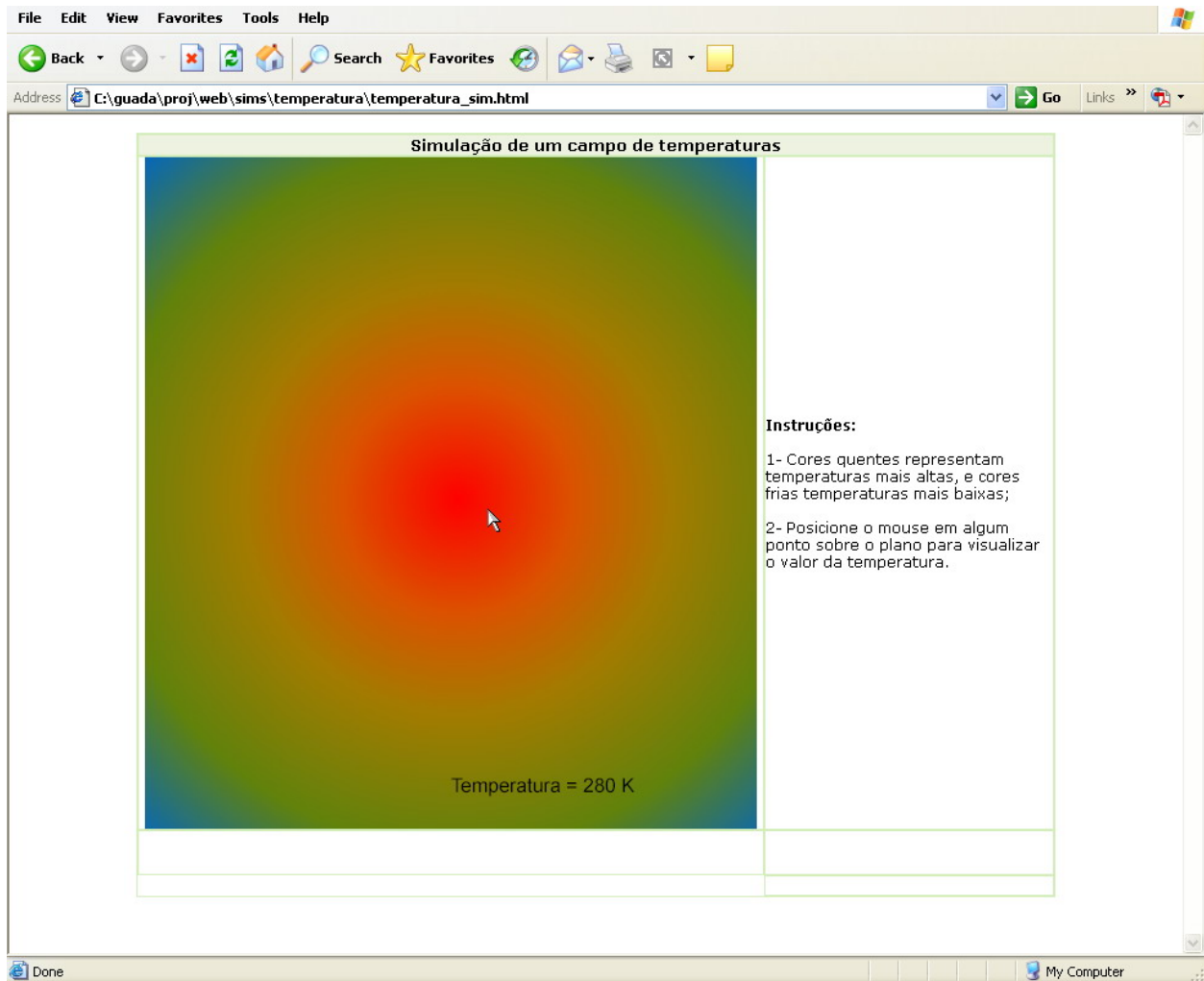


Figura 3: Janela do simulador de um campo de temperaturas.

CONSIDERAÇÕES FINAIS

Podemos considerar que os estudos aqui apresentados indicam que o conjunto de ferramentas de padrão aberto SVG, XML, XHTML e ECMAScript oferecem atributos vantajosos em aplicações computacionais relacionadas à aprendizagem das ciências por meio de experimentação por simulação interativa, e têm implicações para o desenvolvimento de ambientes virtuais de aprendizagem para o ensino presencial e à distância.

Para os simuladores apresentados, as tecnologias abertas de apoio à aprendizagem com a estratégia de simulação interativa mostrou potencial para auxiliar na exploração de situações de ensino relacionadas ao aprendizado dos domínios das representações e dos invariantes operatórios dos conceitos de campo e força. Para o simulador de uma configuração de cargas elétricas pontuais, exploramos os conceitos de campo elétrico e força elétrica, relacionando os elementos de interação (“objetos”) do simulador com as ações do usuário e a resposta do simulador a estas ações, bem como os invariantes operatórios explorados e as suas representações simbólicas relacionadas.

Estudos têm demonstrado que os simuladores, isoladamente, podem não ser suficientes para um bom aproveitamento das atividades de ensino (Mayer & Anderson *apud* Nascimento, 2005). É desejável, portanto, que os simuladores sejam incorporados em um contexto de ensino. Isso pode ser efetivado através da complementação de informações na forma de hipertexto ou outra forma de guia para o aluno.

A análise do desenvolvimento dos conceitos presentes nas situações simuladas consiste numa etapa posterior deste estudo, em que serão analisados invariantes e representações simbólicas mobilizadas pelo aluno quando da interação com estas situações.

REFERÊNCIAS

- Anido, Luis; Llamas, Martín; Fernández, Mael J.. Internet-based learning by doing. Disponível em <<http://www.ewh.ieee.org/soc/es/May2001/17/Begin.htm>> Acesso em 27 de julho de 2005.
- Bogaard, Daniel S.; Vullo, Ronald P.; Cascioli, Christopher D. SVG for educational simulations. In: CONFERENCE ON INFORMATION TECHNOLOGY EDUCATION, 5th, 2004, New York, **Proceedings of the 5th Conference on Information Technology Education**. New York: ACM Press, 2004. Páginas 43-49.
- Campbell, Katy. The Web: Design for Active Learning. Disponível em <<http://www.atl.ualberta.ca/documents/articles/activeLearning001.htm>> Acesso em: 30 de julho de 2005.
- Eisenberg, J. David. **SVG essentials**. 1st ed. Sebastopol: O'Reilly&Associates, Inc., 2002.
- Nascimento, Anna Christina de Azevedo. Princípios de design na elaboração de material multimídia para Web. Disponível em <<http://rived.proinfo.mec.gov.br/artigos/multimidia.pdf>> Acesso em 30 de julho de 2005.
- Neumann, A. **Use of SVG and ECMAScript technology for e-learning purposes**. In: ISPRS WORKSHOP COMMISSIONS VI/1–VI/2 TOOLS AND TECHNIQUES FOR E_LEARNING, Germany, 2005.
- Neumann, Andreas; Winter, Andréas M. **Time for SVG – Towards high quality interactive Web-Maps**. In: PROCEEDINGS OF THE 20th INTERNATIONAL CARTOGRAPHICS CONGRESS, Beijing, 2001. 9 páginas.
- Vergnaud, Gérard. The nature of mathematical concepts. In: Nunes, T.; Bryant, P. (Eds.) **Learning and teaching mathematics: an international perspective**. Hove (East Sussex): Psychology Press Ltd., 1997. Páginas 5-28.
- W3C. World Wide Web Consortium. Disponível em: <<http://www.w3.org>> Acesso em: 21 de agosto de 2005.