Geometria Dinâmica com o iGeom: Algoritmos Geométricos, Autoria e Avaliação Automática de Exercícios

`http://www.matematica.br/igeom`

Sumário

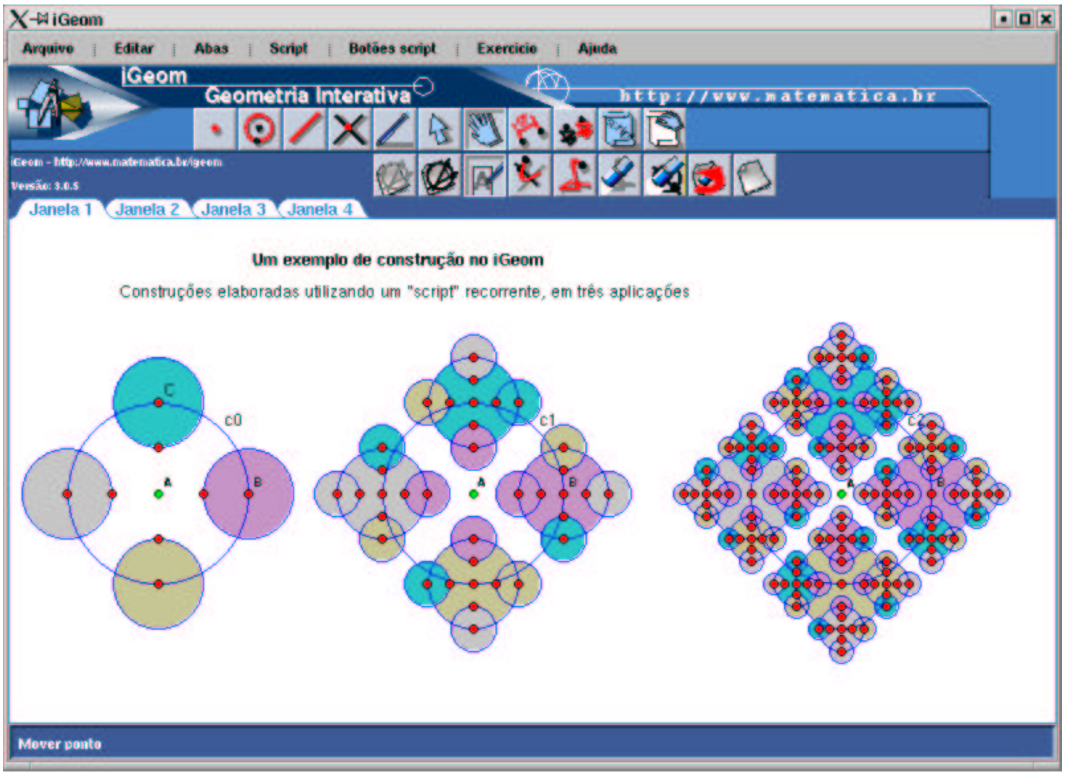
Geometria Dinâmica com o iGeom: Algoritmos Geométricos, Autoria e Avaliação Automática de Exercícios

Leônidas de Oliveira Brandão, Seiji Isotani e Janine Gomes Moura

Departamento de Ciência da Computação
Instituto de Matemática e Estatística
Universidade de São Paulo
Caixa Postal 66.281 – 05315-970 – São Paulo, SP
http://www.ime.usp.br/˜{leo, isotani, janine}

{leo, isotani, janine}@ime.usp.br

Resumo. *O propósito deste texto é apresentar alguns recursos avançados do programa de geometria dinâmica iGeom, além daquelas conhecidas vantagens que a geometria dinâmica pode trazer ao aprendizado de geometria (em geral). Mostramos como é possível explorar algoritmos geométricos com laços (utilizando recorrência), o que possibilita o uso do iGeom na introdução de conceitos de programação, além de explorações matemáticas a partir de fractais geométricos. Também apresentamos os recursos de autoria e avaliação automática de exercícios, além de mostrar como estes podem ser integrados a sistemas gerenciadores de cursos na Web, utilizando um mecanismo de comunicação padrão. Deste modo, se integrado a um sistema gerenciador, os recursos de autoria e avaliação automática de exercícios podem ser utilizados para facilitar a tarefa do professor, que não precisará avaliar pessoalmente as respostas dos alunos, e também para que o aluno saiba de imediato se sua solução está dentro do esperado pelo professor.*



1. Introdução

Neste texto apresentamos o programa para ensino/aprendizagem de geometria, **iGeom - Geometria Interativa na Internet**, destacando dois de seus recursos mais avançados: roteiros de construção (**script**) com recorrência e autoria e avaliação automática de exercícios. Além destes, o iGeom ainda apresenta facilidades para comunicação com servidores podendo, deste modo, eliminar o trabalho de exame de soluções de exercícios enviadas por alunos. Estes recursos aparecem em poucos programas equivalentes ao iGeom, sendo que no momento ele é o único a reuní-los num só programa. O iGeom pode ser descarregado gratuitamente a partir do endereço *http://www.matematica.br/igeom*. A pronuncia de iGeom, deve ser como *igeôm*, para demarcar a origem do termo *geometria* em seu nome.

Atualmente este tipo de programa é bastante conhecido e a geometria que ele possibilita é usualmente denominada **geometria dinâmica**, que abreviaremos por **GD**. Resumidamente, um programa de GD é a implementação computacional da régua e do compasso. Os programas de GD podem trazer várias vantagens ao aprendizado. Uma que podemos destacar é permitir que o aluno experimente, que procure ele próprio elaborar e testar conjecturas [King and Shattschneider, 1997, Gravina, 1996].

Na GD, um "script" pode ser entendido como um roteiro para realizar uma construção, podendo ser formalizado como uma função matemática tendo como domínio e como imagens objetos geométricos (como pontos, circunferências ou retas). Por exemplo, poder-se-ia agrupar a sequência de passos para construir uma mediatriz na forma de "script", tendo como parâmetros um par de pontos e produzindo como imagem a reta mediatriz dos pontos dados. Deste modo, o papel desempenhado pela recorrência é o mesmo nos "scripts" e nas funções[1].

Os "scripts" recorrentes permitem vários usos didáticos. Neste texto destacaremos dois deles: (1) estudo de contagem e convergência associados à fractais geométricos e (2) uma introdução ao conceito de algoritmo e à programação. O primeiro pode ser empregado com alunos do início do ensino médio e o segundo com alunos do final deste ciclo ou em início de graduação.

Em relação ao estudo de contagem/convergência, pode-se utilizar "scripts" previamente preparados para que os alunos procurem descobrir regras de formação (número de objetos, geralmente descritos por progressões geométricas), numa abordagem do tipo "caixa-preta" [Brandão, 2002].

Quanto ao uso dos "scripts" para introduzir algoritmos e programação, destacamos a eliminação da necessidade de aprendizado de uma linguagem formal, como ocorreria, por exemplo, com linguagens de programação como Lisp, Java, Pascal ou C. Nesta abordagem o aluno não precisa codificar os passos de construção de modo descritivo, bastando utilizar símbolos gráficos (ícones) e operações "concretas" definidas a partir dos ícones e dos objetos geométricos. Desde modo, o tipo de programação proporcionado pelos "scripts" na GD é mais intuitivo e concreto que aqueles da linguagem **Logo**, proposta por Seymour Papert para o aprendizado de geometria e de programação [Papert, 1985]. O próprio Papert observa posteriormente que o concreto, em oposição ao "formal-abstrato", é importante para o aprendizado [Papert, 1994, Papert, 1999].

O uso de "scripts" para programação é bastante interessante porque esconde do aluno detalhes de implementação, permitindo que ele se concentre em encontrar os pontos de recorrência. Neste sentido, este paradigma de programação fica mais próximo à programação funcional [Brandão, 2004].

Os recursos de autoria e avaliação automática de exercícios podem ser empregados de modo "aberto" na **Web**[2]. O professor pode produzir os exercícios e publicá-los em qualquer servidor Web,

[1] Um exemplo ilustrativo de função recorrente é a função fatorial, $f(n) = \langle 1, \text{ se } n = 0; n \times f(n-1) \text{ se } n > 0 \rangle$. A recorrência é caracterizada pela aplicação da própria função ($f(n-1)$) em sua definição.

[2] A *World Wide Web*, ou apenas Web ou WWW, é um sistema de armazenamento, recuperação e troca de informação pela Internet, originado no *European Organization for Nuclear Research* (CERN), no início dos anos 90. A novidade trazida pela Internet é permitir a conexão de redes heterogêneas - diferentes tipos de Computadores, trabalhando com diferentes sistemas operacionais, como PC com Linux, PC com Windows, Macintosh com MacOS, Sun com Solaris OS, etc - , de maneira completamente distribuída (sem a necessidade de um "Computador central") e a novidade da Web são os recursos gráficos e um sistema baseado em *hipertexto*: pode-se colocar apontadores, "links",

deixando-os disponíveis para qualquer pessoa que deseja testá-lo. O "internauta" pode tentar resolvê-los e obter, praticamente de imediato, uma avaliação do tipo "acertou/errou". Um exemplo deste tipo de aplicação foi produzida por uma aluna do curso de licenciatura do IME-USP, em uma disciplina obrigatória. O resultado pode ser conferido a partir do URL *http://www.matematica.br/igeom/docs/exemplo1/index.html*.

Entretanto, se o sistema ainda oferece serviços de comunicação, estes recursos podem ser muito melhor aproveitados, principalmente do ponto de vista do professor. A tríade "autoria/avaliação/comunicação" simplifica sobremaneira o trabalho do professor quando a turma de alunos é grande. Outra vantagem óbvia de tal aplicação é oferecer ao aluno uma imediata avaliação de desempenho, evitando algumas frustrações destacadas por [Hara and Kling, 1999].

O método de comunicação implementado no iGeom é um padrão da Web, o método POST [W3C, 2004]. Deste modo, é fácil integrá-lo com sistemas gerenciadores de cursos pela Web. No primeiro semestre de 2004, iniciamos o desenvolvimento de sistema gerenciador, utilizando tecnologias abertas (PHP+MySQL [PHP, 2004, Mysql, 2004]). Este sistema, chamado **Sistema de Aprendizado pela Web** (**SAW**), foi utilizado numa disciplina obrigatória em três turmas, com aproximadamente 150 alunos. O SAW também será em breve disponibilizado gratuitamente pela Web [Brandão et al., 2004].

Os recursos aqui discutidos, disponíveis no iGeom, não aparecem simultaneamente em nenhum outro programa de GD. Por exemplo, além do iGeom, temos conhecimento de apenas um outro programa de GD que dispõe de "scripts" recorrente, que é o Geometer's Sketchpad [Jackiw, 1995]. Do mesmo modo, recursos associados à produção e resolução de exercícios também são raros nos programas de GD. Além do iGeom, apenas o Cinderella [Richter-Gebert and Kortenkamp, 1999] e a versão Java do C.a.R [Grothman, 1999] apresentam ferramentas para isso.

Uma introdução mais completa ao conceito de geometria dinâmica (GD) será feita na seção 2. Na seção 3 discutimos o uso de "scripts" recorrentes do iGeom para a exploração de fractais geométricos e também seu uso para introduzir os conceitos de algoritmo e de programação. Na seção 4 apresentamos os recursos de autoria e avaliação de exercícios do iGeom, e na seção 5 como é possível utilizá-los em sistemas gerenciadores de cursos na Web. Na seção 6 concluimos os tópicos propostos no texto.

2. Introdução à Geometria Dinâmica

O uso do computador pode trazer grandes benefícios ao ensino de matemática, mas para isso é necessário escolher programas (*software*) adequados e uma metodologia que tire proveito das características positivas do computador, como boas representações gráficas e rapidez em cálculos.

Um bom exemplo desta agilização no estudo de matemática é o uso (adequado) de programas de Geometria Dinâmica (**GD**) [Brandão and Isotani, 2003, Richter-Gebert and Kortenkamp, 1999, Laborde and Bellemain, 1997, Jackiw, 1995]. Podemos dizer que a GD é Geometria de "régua e compasso" implementada no Computador, na qual o aluno pode, a partir de uma construção inicial, mover com o *mouse* algum dos objetos iniciais e o programa se encarrega de redesenhar toda sua construção, de modo aparentemente contínuo. Daí a origem do termo dinâmico na GD, em oposição a característica estática da geometria tradicional, da régua-e-compasso.

Deste modo, um programa de GD possibilita ao aprendiz, a partir de uma única construção, efetuar um número arbitrário de testes (para procurar ou verificar uma conjectura), o que seria praticamente impossível com régua e compasso. Por isso podemos dizer que a GD é do tipo 1**-construção,** N**-testes** enquanto a geometria de "régua e compasso" é do tipo 1-construção, 1-teste [Brandão, 2002].

Um exemplo simples pode ilustrar o *dinamismo* desta geometria: a construção da mediatriz de dois pontos dados, A e B. Para construir a mediatriz basta encontrarmos dois pontos distintos que equidistem de A e de B e por eles traçar a reta resposta (mediatriz).

---

que ao ser "clicado" com o "mouse" leva a outros textos, imagens ou outros recursos - para informações detalhadas sobre a histórica da Internet e da WWW, na forma de linha do tempo, consulte por exemplo, as páginas Web *http://www.w3.org/History.html*[W3C, 2004] e *http://www.zakon.org/robert/internet/timeline*.

Exemplo 1 *Dados dois pontos, A e B, construir sua mediatriz (lugar geométrico dos pontos que equidistam dos dois pontos dados).*

Construção 1 *Dados: ponto A, ponto B (resposta: reta r)*
1. *construir a circunferência $C0$, com centro no ponto A e contendo B,*
2. *construir a circunferência $C1$, com centro no ponto B e contendo A,*
3. *construir a reta r definida pelos pontos de interseção entre $C0$ e $C1$, C e D.*

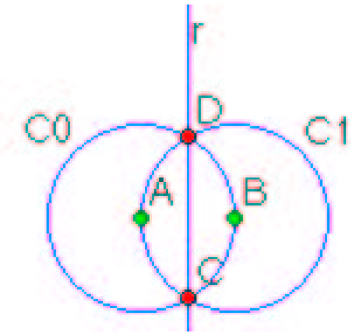


Figura 1: mediatriz de A e B

A maneira como se constróem os objetos geométricos em programas de GD varia, entretanto todos eles dispõem de recursos que desempenham o papel da régua e do compasso. Mais importante, todos eles apresentam o "dinamismo": uma vez pronta uma construção, como a anterior, podemos arrastar o ponto A (ou B) pela área de desenho e o programa, automaticamente, redesenha toda a figura preservando as propriedades iniciais (p.e., r continuará a ser a mediatriz dos pontos rotulados com A e B, nas novas configurações).

Outras duas características interessantes de programas de GD são: introduzir a necessidade de melhor formalização das construções e, devido ao dinamismo, facilitar a verificação de validade da mesma. A formalização é necessária devido ao computador não admitir ambiguidades[3] e movendo os objetos iniciais podemos verificar mais facilmente se a construção preserva as propriedades esperadas. No exemplo acima, se o usuário não construiu corretamente a mediatriz, ao arrastarmos o ponto A pela área de desenho aparecerá algum erro.

2.1. GD em página Web

Os conteúdos na Web são apresentados a partir de uma linguagem de marcação denominada **HTML** (**HyperText Markup Language**) [W3C, 2004]. Esta linguagem permite estruturar o documento apresentado e permite fazer ligações com outros documentos (*hyperlink*). Apesar do HTML não prover suporte direto para animações/interatividade, ele dispõe de "tags" para inserir outras linguagens mais poderosas. Estas linguagens precisam de interpretadores, embutidos no navegador ("Web browser"), que são conhecidos como **plug-ins**.

Uma primeira linguagem a ser utilizada com este propósito foi a linguagem **Java** [Arnold and Gosling, 1996] definida pela **Sun**. A própria Sun disponibiliza gratuitamente "plug-ins" Java para maioria das plataformas (computador+sistema operacional), a partir do endereço *http://www.javasoft.sun.org*. Estes interpretadores Java são denominado *Máquinas Virtuais Java* (JVM).

O iGeom foi desenvolvido em Java em função desta "portabilidade", podendo ser utilizado em qualquer plataforma e em qualquer navegador com interpretador Java.

Existem vários outros programas de GD, que funcionam em diferentes plataformas (Linux, Windows, Mac), com diferentes licenças e que possibilitam ou não seu uso na Web. Na tabela 1 destacamos alguns programas de GD, dentre aqueles que apresentam um bom número de recursos e que estão estão entre mais conhecidos ou que sejam gratuitos.

[3] Um exemplo de ambiguidade seria "construir uma circunferência definida pelos pontos A e B", enquanto uma versão não ambígua poderia ser "construir a circunferência de centro A, que passa pelo ponto B (ou o contém)".

Tabela 1: Tabela com alguns dos programas de GD.

Programa	Plataforma	Exporta Java	Licença
Cabri	Windows, Mac	sim	comercial
C.a.R[4]	Windows	não	gratuito
Cinderella	todas (Java)	sim	comercial
Dr. Genius	Linux, Windows	não	gratuito
Euklid	Windows	não	shareware
GSP	Windows, Mac	sim	comercial
iGeom	todas (Java)	sim	gratuito

Na tabela 1, apenas o Cinderella e o iGeom são implementados diretamente em Java. Os demais precisam de interpretadores feito em Java para permitirem visualizar suas construções em páginas Web. A coluna "exporta Java" indica a existência de um interpretador Java e algum mecanismo para disponibilizar na Web construções feitas no programa. Entretanto, normalmente por razões comerciais, os programas que não são implementados em Java, não permitem que a construção "exportada" seja alterada pelo usuário, ou seja, uma vez "exportada" a construção o usuário só poderá mover seus objetos.

Ainda existe um outro programa de GD desenvolvido no Brasil, também utilizando Java, que é o **Tabullae** [Guimarães et al., 2002]. Ele foi desenvolvido por um grupo do IM - UFRJ, coordenado por Luiz Carlos Guimarães, mas ainda não dispõe de um sistema de distribuição ou venda, nem pela Internet.

2.2. O iGeom

O iGeom [Brandão and Isotani, 2003] é um sistema em contínuo desenvolvimento. O grupo de desenvolvedores, liderado pelo primeiro autor, contou até o momento com a participação ativa de quatro desenvolvedores: o primeiro autor desde sua concepção e primeira implementação, Ricardo Hideo Sahara e Fabiana Piesigilli em projetos de iniciação científica e Seiji Isotani em seu projeto de mestrado no IME-USP.

O iGeom começou a ser desenvolvido no segundo semestre de 2000, por Ricardo Hideo Sahara e pelo primeiro autor [Sahara, 2001]. No primeiro semestre de 2001, Fabiana Piesigilli iniciou seu trabalho, participando principalmente da primeira versão do gravador e interpretador de "scripts" recorrentes [Piesigilli, 2002]. No início de 2003, Seiji Isotani iniciou seu trabalho com o iGeom, colaborando principalmente na construção do sistema de autoria e avaliação de exercícios e na comunicação do iGeom com o servidor Web.

A escolha da linguagem **Java** [Java, 2004] para implementar o iGeom foi para que ele funcionasse em qualquer computador e também pudesse ser utilizado em páginas Web. Por isso, a interface do programa foi desenvolvida com o intuito que o mesmo funcione nas formas **aplicativo** e **applet**. A versão "aplicativo" é a mais geral, permitindo gravar e recuperar arquivos nos vários formatos previstos, enquanto a versão "mini-aplicativo", ou "applet", pode ser utilizada em qualquer navegador com interpretador Java.

O iGeom está disponível para uso na Internet e para ser descarregado a partir do sítio do **iMática**, endereço *http://www.matematica.br* [Isotani and Brandão, 2001]. Lá também será encontrado um pequeno roteiro para o uso do programa e exemplos, a partir do endereço *http://www.matematica.br/igeom*.

2.2.1. A interface Principal

A interface do iGeom, na versão aplicativo, é estruturada a partir de uma barra de menus, duas barras de ferramentas compostas por botões (ícones), a área de desenho e uma barra de mensagens. Na figura 2 apresentamos a interface principal do iGeom, utilizando sua versão 3.0.5. A interface da versão *applet* é análoga, mas não dispõe da barra de menus[5].

[4] Atualmente existe uma nova versão do C.a.R. em Java.
[5] Por razões de segurança não é permitido a um *applet* acessar o disco rígido do micro-cliente.

A **barra de menus** contém opções para edição de arquivos (gravar/recuperar/exportar construções), criação de "scripts" e exercícios, dentre outras opções.

As barras de ferramentas são divididas em dois níveis: na fileira superior está **barra principal** e na inferior a **barra secundária**. A estrutura destas barras pode ser entendida como uma versão gráfica dos sistemas do tipo menus, como a barra de menus acima citada. Um botão da barra principal pode ter ou não uma lista de botões associados na barra secundária. Caso tenha, este botão não terá qualquer ação correspondente, a não ser servir para abrir sua a lista secundária.

Por exemplo, o botão de criação de retas na barra principal, ao ser clicado, fará aparecer na barra secundária todas as opções relacionadas com retas. Na versão atual do iGeom, a lista secundária conta com 7 botões, como indicado na figura 2.

Todos os recursos do iGeom, a menos daqueles relativos à gravação/leitura em arquivos, estão diponíveis a partir das barras de ferramentas. Podendo, portanto, serem utilizados na versão *applet*.

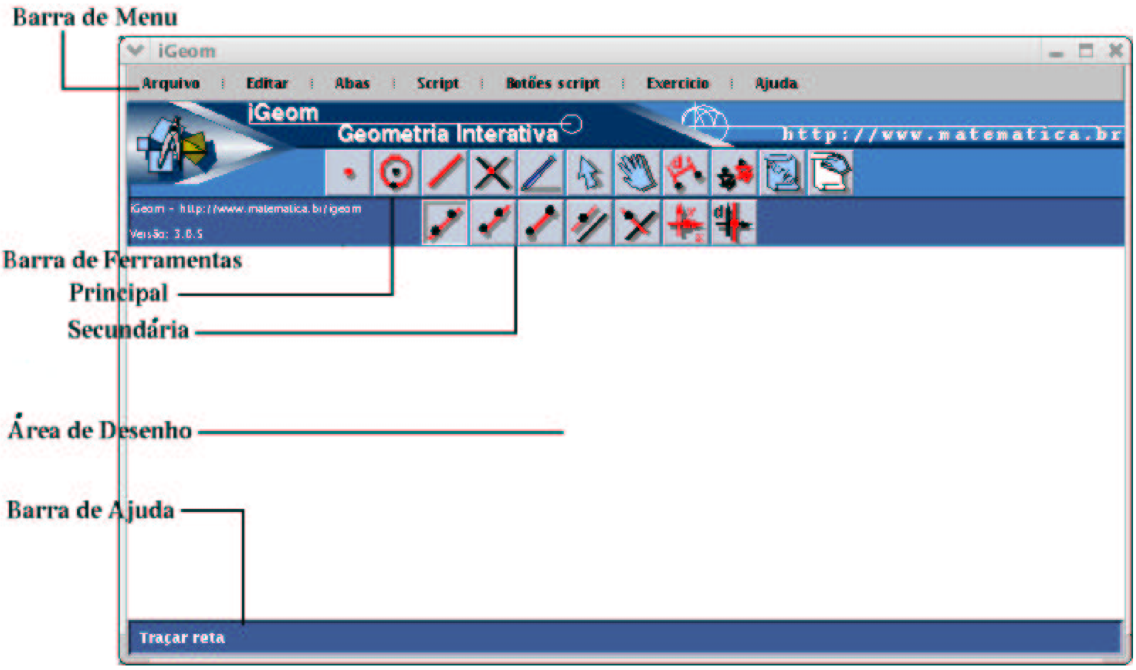


Figura 2: Janela Principal do iGeom

A **área de desenho** funciona com uma folha de papel, na qual é possível desenhar os objetos geométricos e interagir com eles. Na parte inferior da interface está a **barra de ajuda** que irá apresentar dicas de construção e de falhas de operação.

2.2.2. As Barras de Ferramentas e Algumas Operações Básicas

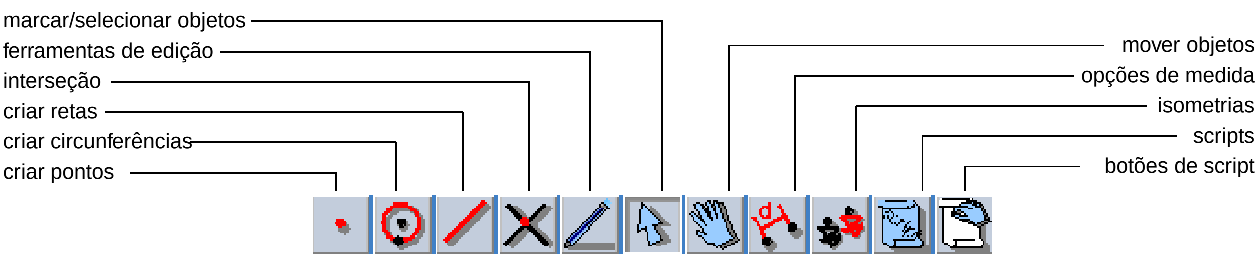


Figura 3: Barra de ferramentas principal

O sistema das barras de ferramentas foi desenvolvido para que o iGeom seja mais portável. Ele e o restante do sistema estão baseados em AWT (Abstract Windowing Tookit) [Java, 2004], podendo ser rodado em máquinas com interpretadores Java desde a versão 1.1 até o atual Java 1.4.

A barra de ferramentas principal contém atualmente 11 botões, a maioria deles dispondo de uma lista de botões secundários. Na figura 3 aparece um resumo descritivo desta barra.

De modo geral, os botões associados a alguma construção geométrica utilizam as cores vermelho e preto. Os objetos em preto devem ser entendidos como as bases para obter os objetos em vermelho. Por exemplo, o segundo botão da barra secundária, associado ao primeiro da barra principal, indica que será produzido um ponto a partir de outros dois. Este botão aparece na figura 4.

O modo como as construções são feitas nos programas de GD, podem ser resumidas em dois tipo, que podemos abreviar por: tipo *seleção+ação* ou do tipo *ação+seleção*. No tipo **seleção+ação**, primeiro são selecionados (ou marcados) os objetos base (ou entradas) e depois é aplicado sobre eles a ação de construção. No tipo **ação+seleção**, primeiro é escolhida a ação e depois são selecionados (marcados) os objetos sobre os quais será aplicada a ação.

Os programas representantes destes modos de operação são o *Geometer's Sketchpad* [Jackiw, 1995], que implementa o tipo "seleção+ação", e o *Cabri* [Laborde and Bellemain, 1997], que implenta o tipo "ação+seleção". O iGeom implementa ambos os tipos de operação, ficando a cargo do usuário escolher a forma mais conveniente de cada operação.

O modo de operação "seleção+ação" é um pouco mais simples de ser implementado, e explicado, em função de não haver mudança de contexto no momento da seleção dos objetos. Enquanto o modo "ação+seleção" naturalmente demanda mudança de contexto, como por exemplo, na utilização do botão "ponto médio": após clicar no referido botão, pode-se avisar ao usuário que agora ele deve clicar num primeiro ponto e após isso que é esperado o segundo ponto para que, finalmente, o ponto médio seja construído. Devido a este maior escopo de variantes do modo "ação+seleção", ele será utilizado na maior parte das explicações de operação a seguir.

A sequência de apresentação das operações segue uma ordem lexicográfica $\prec$, definida assim:

- $(i, 0)$ é o i-ésimo botão primário (da esquerda para a direita);
- (i, j) é o j-ésimo botão secundário associado ao i-iésimo primário (da esquerda para a direita);
- $(i_1, j_1) \prec (i_2, j_2) \Leftrightarrow i_1 < i_2$ ou $(i_1 = i_2$ e $j_1 < j_2)$.

Assim, começaremos pelos botões secundários associados à construção de pontos (figura 4), seguido pelos botões de construção de circunferências (figura 5). Não detalharemos as possibilidade de criação de retas, que desmembra-se em várias outras.

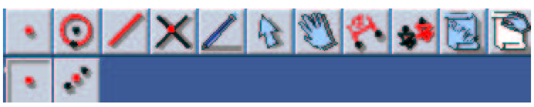


Figura 4: Opções de ponto.

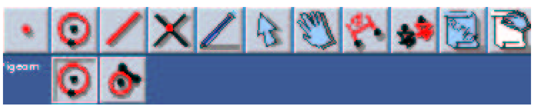
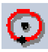


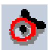
Figura 5: Opções de Circunferência.

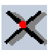
1.  **Sobre a criação de pontos**
   Botão: $(1, 1)$
   Ao clicar no botão de criação de pontos na barra principal, surgem duas opções na barra secundária, como indicado na figura 4. Utilizando a forma de construção "ação+seleção", o primeiro botão desta lista secundária, permite a construção de três categorias de pontos:
   (a) *Criar ponto completamente solto*: Estando selecionado o primeiro botão de pontos da barra secundária, ao clicar sobre a área de desenho, será criado um ponto (originalmente com cor verde), que pode ser movido livremente.
   (b) *Criar ponto sobre objeto*: Também com o primeiro botão de pontos selecionado, ao clicar sobre um único objeto do tipo reta (podendo ser semi-reta ou segmento) ou circunferência, será criado um ponto (também verde), que ficará sempre sobre o referido objeto.

   (c) *Criar ponto de interseção*: Ainda com o primeiro botão de pontos selecionado, ao clicar na vizinhança de dois objetos do tipo reta ou circunferência, será criado o ponto representando a interseção destes dois objetos.
   Vale ressaltar que se um dos objetos for uma circunferência, podem exitir dois pontos de interseção, sendo gerado aquele mais próximo da região em que ocorreu o clique.

2.  **Sobre a criação de pontos médios**
   Botão: $(1, 2)$
   Estando selecionado o segundo botão de pontos da barra secundária, deve-se clicar sobre um ponto A existente na área de desenho e depois pode-se clicar em qualquer posição da área de desenho. Se o segundo clique for:
   (a) sobre um ponto B existente, será criado o ponto médio entre os pontos A e B.
   (b) sobre um objeto do tipo reta ou circunferência, primeiro será criado um ponto B, sobre o referido objeto, e depois C, o ponto médio entre A e B.
   (c) na vizinhança de dois objetos do tipo reta ou circunferência, será criado o ponto B de interseção entre eles, como no item 1c acima, e depois o ponto C, ponto médio entre A e B.

3.  **Sobre a criação de circunferência definida por um par de pontos**
   Botão: $(2, 1)$
   Esta opção corresponde ao primeiro botão de criação de circunferências, na barra secundária associada ao segundo botão da principal, e pode ser acionada de várias formas distintas. Entretanto, o que não muda é que: deve ser providenciado um par ordenado de pontos, A e B, sendo construída a circunferência com centro A e raio definido pela distância $||A - B||$. Portanto, no sentido matemático, o ponto B pertencerá à circunferência c definida por: c é o lugar geométrico dos pontos que distam $||A - B||$ unidades de A.
   Tanto o primeiro clique, que define A, quanto o segundo, que define B, podem ser como nos itens 1a, 1b e 1c, relativos à criação de pontos.

4.  **Sobre a criação de circunferência definida por pontos e segmento**
   Botão: $(2, 2)$
   Nesta opção pode-se criar uma circunferência de centro dado por um ponto A e raio definido pelo comprimento de um segmento s, também previamente existente. A definição do centro A da circunferência pode ser feita como em uma das três opções de criação de pontos, indicadas anteriormente nos itens 1a, 1b e 1c. O segmento s que define o raio tem que ser um já existente na área de desenho.

5.  **Sobre a criação de pontos de interseção entre dois objetos ("interseção")**
   Botão: $(4, 0)$
   Com este botão selecionado, ao clicar na vizinhança de dois objetos cria-se o ponto de interseção entre ambos. Havendo mais de um ponto de interseção entre eles, será criado aquele mais próxima à região do clique.
   No modo de operação "seleção+ação", se ao clicar neste botão já existirem dois objetos marcados (veja como marcar no item 11), o botão "interseção" não será selecionado (não será abaixado), mas será criada a interseção entre os objetos marcados.

6.  **Esconde objetos ("esconder")**
   Botão: $(5, 1)$
   Com este botão selecionado, ao clicar sobre um objeto, este não mais aparecerá na área de desenho, o que não significa ser removido da construção.
   No modo de operação "seleção+ação", se ao clicar neste botão já existirem objetos marcados (com o marcador do item 11), o botão "esconder" não será selecionado (não será abaixado), mas todos os objetos marcados serão escondidos.

7.  **Apresenta todos os objetos escondidos ("mostrar")**
   Botão: $(5, 2)$

Ao clicar neste botão, todos os objetos escondidos aparecerão marcados (em cor amarela). Um segundo clique neste botão fará com que todos os objetos fiquem visíveis, na cor original dele. Se o usuário desejar que fique visível apenas alguns destes objetos, deve: com o botão "marcador" selecionado, clicar naqueles que deseja que voltem a ficar visíveis e depois clicar novamente no botão "esconder", para esconder os demais.

8. **Edita atributos de objetos ou cria textos ("texto")**
   Botão: $(5, 3)$
   Este botão agrupa as característica de edição de texto, incluindo aí rótulos e medidas dinâmicas. Se ele estiver selecionado (abaixado), todos os objetos geométricos criados nascerão com rótulos, em caso contrário, eles nascem sem rótulo. Além desta funcionalidade, com ele ainda:
   (a) Pode-se editar atributos de cores dos objetos geométricos ou os vários atributos de rótulos ou textos, bastando dar um duplo clique sobre o objeto alvo.
   Se o alvo for um texto (incluindo aí rótulo e medidas dinâmicas), será aberta uma janela,

Figura 6: Janela para modificar o rótulo

   como na figura 6. Nela o usuário pode trocar a cor do texto, pode trocar a fonte ou o seu tamanho e ainda pode deixá-la em negrito ou itálico.
   Se o alvo for um objeto geométrico abre-se uma janela parecida com a de edição de texto, mas com opção apenas de alteração de cor.
   (b) Pode-se criar textos na área de desenho, para isso deve-se:
   - clicar no botão primário de edição ( ), de modo que sua barra secundária fique visível;
   - clicar no ponto em que deseja o texto, manter o botão pressionado, arrastar o *mouse* pela área de desenho de modo a aparece algum retânculo na tela (de qualquer tamanho) e soltar o botão do *mouse*;
   - clicar no botão de "rótulo", será aberta uma janela como na figura 7 para digitar seu texto (atualmente em uma linha) - ao finalizar, clique no botão **OK**.

9. **Remove objetos na área de desenho ("remover")**
   Botão: $(5, 7)$
   Do mesmo modo que os itens 5, 6 e 7 anteriores, existem dois tipos de operação dependendo de já haver ou não objetos marcados ao clicar no "remover": não havendo objetos marcados ("ação+seleção"), o botão é selecionado (ficando abaixado), podendo-se clicar nos objetos que deseja remover da construção; havendo objetos marcados ("seleção+ação"), o botão "remove" não é selecionado, e os objetos marcados são todos removidos.
   Note que todos os objetos que dependem de um dos removidos também será removido. Por exemplo, ao remover um ponto A que é extremo do segmento s, o segmento também será removido.

10. **Remove todos os objetos na área de desenho ("limpar")**
   Botão: $(5, 9)$

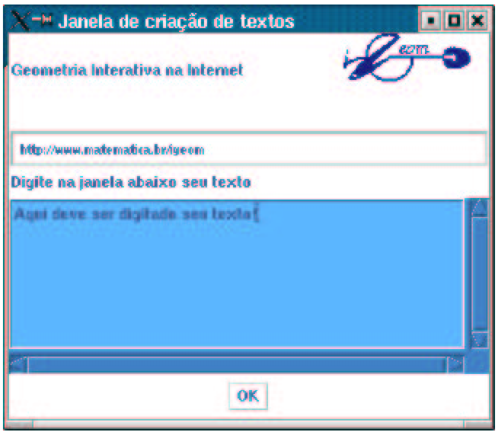


Figura 7: Janela para criar um texto

Ao clicar no botão "limpar" todos os objetos da área de desenho serão removidos. É possível recuperá-lo se imediatamente depois clicar o botão à esquerda dele (botão $(8, 8)$, ).

11.  **Marca/seleciona objetos ("marcar")**
Botão: $(6, 0)$
Este botão é utilizado para preparar operações do tipo "seleção+ação". Com ele selecionado é possível marcar objetos para que seja, sobre eles, realizada alguma operação, dentre às várias existentes. Por exemplo, se forem marcados os pontos A e B de uma construção e depois: clicar no botão "ponto médio" (item 2), será criado o ponto médio entre A e B; ou clicar no botão "esconder", ambos os pontos ficarão "invisíveis"; ou clicar no botão "remover", ambos serão removidos da área de desenho, juntamente com todos os objetos que dependem de A ou de B.
Também é possível marcar vários objetos de uma vez, clicando em algum ponto da área de desenho (em que não haja algum objeto) e, com o botão do *mouse* presssionado, movê-lo para outro ponto e soltar o botão do *mouse*. Todos os objetos que ficarem dentro da área retângular formada ficarão marcados. Qualquer operação feita a seguir, fará com que este retângulo desapareça.

12.  **Move pontos (e re-escalona objetos) ("mover")**
Botão: $(7, 0)$
Com este botão selecionado, deve-se clicar sobre um ponto *alvo* (o botão do *mouse* não deve ficar pressionado) e depois mover o *mouse*. O ponto alvo acompanhará a movimentação do *mouse* até o próximo clique. Todos os objetos dependentes do ponto alvo também serão atualizados e se houver alguma medida dinâmica, que dependa deste ponto ou de algum dependente, também ela será atualizada.

13.  **Cria expressões aritméticas dinâmicas (envolvendo medidas e expressões)**
Botão: $(8, 1)$
Este botão abre uma calculadora que permite construir expressões aritméticas básicas com qualquer medida dinâmica, o que inclui outra expressões. A calculadora é apresentada na figura 8.

Este recurso é bastante útil para que o aluno procure descobrir relações ou propriedades. Uma área da matemática elementar que pode ser explorada com ele, é a **trigonometria**. Por exemplo, pode-se realizar um experimento em que o aluno perceba que um comprimento de arco dividido pelo raio é uma constante desde que o ângulo não mude, como ilustrado na figura 9.

14.  **Cria medidas (distâncias) dinâmicas ("medir")**
Botão: $(8, 2)$
Atualmente esta opção só está implementada para o modo de operação do tipo "seleção+ação". Com dois objetos marcados (via "selecionador"), ao clicar em "medir" será criada uma "medida dinâmica" associada aos dois objetos. Por exemplo, se os objetos forem: ponto e ponto a medida

Figura 8: Calculadora para expressões dinâmicas

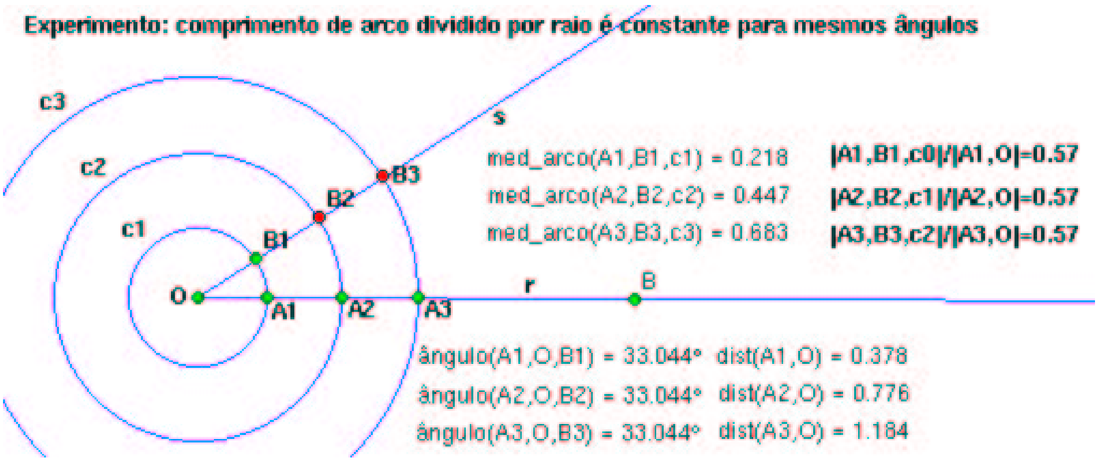


Figura 9: Descobrindo uma propriedade de comprimento de arco, ângulo e raio

será a distância (euclideana) entre ambos; ponto e reta a medida corresponderá à distância entre o ponto e a reta; ponto e circunferência a medida corresponderá à distância entre o ponto e a circunferência.
A medida criada é "dinâmica" no sentido de, ao mover um dos objetos que a definem, ela será automaticamente recalculada para a nova configuração.

Inicialmente os objetos criados possuem alguns atributos padrões. Por exemplo, os rótulos dos pontos são iniciados com letras maiúsculas, enquanto os rótulos para retas são grafados em letras minúsculas.

Após criados os objetos geométricos na área de desenho, é possível gravá-los em arquivo para sua posterior recuperação. Para isso, seleciona-se a opção *Arquivo* da barra de menus e dentro dela seleciona-se *Gravar construções...*. Será aberta uma janela na qual deve-se escolher o diretório e o nome do arquivo a ser gravado e clicar no botão **OK** desta janela. Se já existir algum arquivo (ou diretório) com o mesmo nome, o iGeom irá perguntar se deseja escrever sobre o arquivo existente ou cancelar a gravação. Pode-se utilizar a opção *Gravar como...* para gravar o arquivo com outro nome.

Além das funcionalidades apresentadas nesta seção, o iGeom possui várias outras, como por exemplo, uso de múltiplas abas, via opção *Abas* na barra de menus. Com elas pode-se manter diferentes construções num mesma instância do iGeom.

Outras funcionalidade avançadas do iGeom aparecem na seções 3 e 4. A seção 3 é sobre a utilização de "scripts" recorrentes, para explorar algoritmos e fractais. A seção 4 é sobre autoria e avaliação automática de exercícios.

3. Algoritmos, programação e fractais com o iGeom

Uma construção geométrica, ou uma solução para um problema geométrico que implique em construção, pode ser verbalizada com instruções do tipo "construa uma circunferência com centro em A, passando por B, como no exemplo 1, da mediatriz. Além disso estas instruções precisam ser bem definidas, sem ambiguidades, e deste modo podemos agrupá-las na forma de função, cujos parâmetros são objetos sobre os quais efetuamos a construção.

No exemplo 1 os parâmetros de entrada foram os dois pontos, A e B, e o objetos resposta (ou saída) foi a reta r. Entretanto, podemos imaginar algumas "funções geométricas" que não têm claramente uma resposta.

Podemos agrupar a solução do exemplo 1 como **mediatriz(Ponto P_1, Ponto P_2)** e depois aplicá-la em quaisquer pares de pontos, obtendo a correspondente mediatriz. No exemplo 1 a chamada da função seria "mediatriz(A,B)".

A este agrupamento de comandos damos o nome de **"script"** (**macro-construções** ou apenas **macro**). Na construção da mediatriz a ordem dos pontos iniciais pode ser invertida sem qualquer alteração do resultado final, entretanto existem casos em que esta ordem é relevante. Um exemplo simples de algoritmos em que a ordem é essencial, é a divisão de um número por outro. Abaixo apresentamos um exemplo geométrico em que a alteração da ordem dos parâmetros de entrada produz resultados diferentes.

Exemplo 2 *Sejam $d(P_1, P_2)$ a distância entre os pontos P_1 e P_2 e $\vec{P_1P_2}$ a semi-reta com origem em P_1 e passando por P_2. Dados os pontos, A e B, definir o ponto F tal que: $F \in \vec{AB}$ e $d(A, F) = \frac{3 \times d(A,B)}{2}$ $\left(\Rightarrow d(B, F) = \frac{d(A,B)}{2}\right)$.*

Construção 2 *(dist_metade) Entradas: ponto A, ponto B*
Saída: ponto F tal que $F \in \vec{AB}$ e $d(A, F) = \frac{3 \times d(A,B)}{2}$

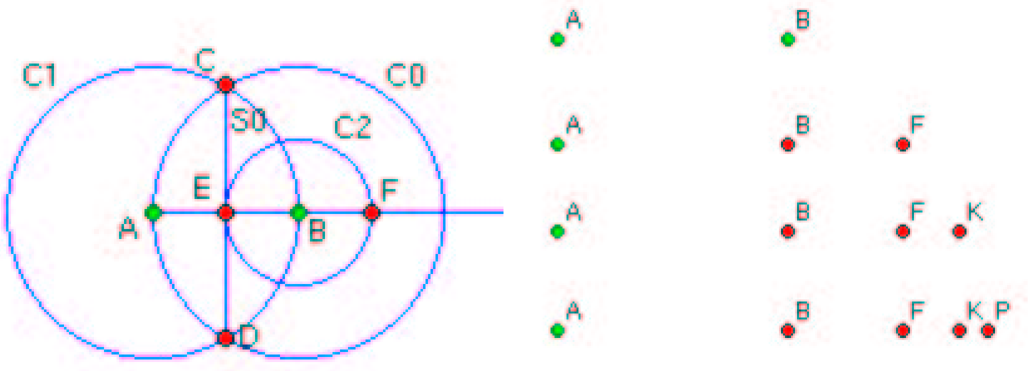
1.	$r \leftarrow semi\_reta(A, B)$	*semi-reta começando em A, passando por B*
3.	$C0 \leftarrow circ(B, A)$	*circunferência centrada em B, passando por A*
2.	$C1 \leftarrow circ(A, B)$	*circunferência centrada em A, passando por B*
4.	$C \leftarrow int_s(C0, C1)$	*C é interseção "superior"[6]entre as circunferências $C0$ e $C1$*
5.	$D \leftarrow int_i(C0, C1)$	*D é interseção "inferior" entre as circunferências $C0$ e $C1$*
6.	$S1 \leftarrow segm(C, D)$	*$S1$ é o segmento ligando C à D*
7.	$E \leftarrow int(r, S1)$	*E é interseção entre r e $S1$*
8.	$C2 \leftarrow circ(B, E)$	*circunferência centrada em B, passando por E*
4.	$F \leftarrow int_d(C2, r)$	*F é interseção "direita" entre a circunferência $C2$ e a semi-reta r*

Se a função dist_metade(.,.) for aplicada invertendo-se as entradas, ou seja, com dist_metade(B, A), mantendo as configurações de A e B como na figura, o ponto F ficará à esquerda de A e não à direita de B. Isso ocorre porque as interseções envolvendo circunferências podem produzir dois resultados distintos, dependendo da orientação dos pontos que definem as retas e circunferências. Na construção distinguimos as interseções com os conceitos de "superior", "inferior", "direito" e "esquerdo", que claramente depende do referencial.

Na figura 10, em (a) estão representados todos os objetos geométricos da construção 2. Em (b) estão representadas seguidas aplicações da macro: $F \leftarrow$dist_metade(A, B), depois $K \leftarrow$dist_metade(B, F) e então $P \leftarrow$dist_metade(F, K).

Da lista de programas da tabela 1, na página 6, os que dispõem de agrupamento na forma de "script" são o Cabri, Dr. Genius, GSP e iGeom.

[6] Para a representação computacional da geometria euclideana é necessário distinguir os dois (eventualmente um) pontos de intereseção, novamente para evitar ambiguidade - o mesmo vale para interseção entre retas (e análogos) e circunferência.



(a) representação gráfica para o exemplo 2 (b) aplicações para (A, B), (B, F) e (F, K)

Figura 10: aplicações da função/"script" dist_metade(.,.).

3.1. Algoritmos geométricos

Nos exemplos anteriores notamos claramente uma forma de algoritmo: para cada "script", existe uma sequência de entradas (em ambos, os pontos iniciais, A e B), uma sequência de passos (indicado nas contruções) e eventualmente uma sequência de saídas (no exemplo 1 é a reta mediatriz e no exemplos 2 o ponto F). Vale observar que este tipo de algoritmo ("função geométrica" ou "algoritmo geométrico") é determinístico, pois sempre que o aplicarmos sobre os mesmos objetos iniciais (pares de pontos), obteremos sempre os mesmos objetos finais (mediatriz no exemplo 1 e ponto no 2).

Podemos formalizar a função do exemplo 2 da seguinte forma (mais compacta),

dist_metade: ponto $\times$ ponto $\longrightarrow$ ponto

dist_metade$(A, B) = F$: $F \in \vec{AB}$ e $d(A, F) = 3 \times d(A, B)/2$.

Esta função pode ser utilizada para simular um problema clássico da matemática, o **paradoxo de Zenão** [Ávila, 1999], enunciado no exemplo abaixo.

Exemplo 3 *Aquiles e uma tartaruga apostam uma corrida, sendo que Aquiles tem o dobro da velocidade da tartaruga e, por isso, a tartaruga larga à frente.*

Podemos simular esta hipotética corrida a partir de eventos discretos, definindo duas sequências, uma de tempo $\{t_k\}_{k \in I\!N}$ e uma de posição $\{P_k\}_{k \in I\!N}$. Sendo t_0 o instante de largada e P_0 a posição inicial da tartaruga, podemos definir as sequências, para $k > 0$, da seguinte forma:

t_k : o instante em que Aquiles atinge a posição P_{k-1}

P_k : a posição ocupada pela tartaruga no instante t_{k+1}.

Deste modo, a construção 2 serve para gerar os pontos P_k, como esquematizado abaixo.

Sendo A e B as posições iniciais, respectivamente, de Aquiles e da tartaruga, então:

1. $P_0 = B$ é a posição inicial da tartaruga e A é a posição inicial de Aquiles;
2. $P_1 \leftarrow$ dist_metade(A, P_0) quando Aquiles chegar à posição P_0 a tartaruga estará em P_1 $\left(\text{da construção, } P_1 \in \vec{AP_0} \text{ e } d(P_0, P_1) = \frac{d(A,B)}{2}\right)$;
3. $P_2 \leftarrow$dist_metade(P_0, P_1) quando Aquiles chegar à posição P_1 a tartaruga estará em P_2 $\left(P_2 \in \vec{P_0P_1} \text{ e } d(P_1, P_2) = \frac{d(P_0,P_1)}{2} = \frac{d(A,B)}{2^2}\right)$,

e assim por diante.

Zenão argumentava que a metade de um número positivo (distância) é um número positivo e, deste modo, sendo o tempo e o espaço contínuos, Aquiles jamais alcançaria a tartaruga[7]. Uma apresentação

[7] Note que o problema deste argumento é supor que a soma de infinitas parcelas positivas será sempre $+\infty$, o que não é verdade. No exemplo, temos uma soma de p.g. infinita, de razão menor que 1, $1/2 + (1/2)^2 + (1/2)^3 + \cdots = 1$.

didática deste problema aparece na Revista do Professor de Matemática, num artigo de Geraldo Ávila [Ávila, 1999].

O aparente paradoxo divulgado por Zenão ilustra a dificuldade que os matemáticos tinham, antes do surgimento do Cálculo Diferencial e Integral, com o conceito infinito e infinitésimo.

Notamos na simulação anterior uma outra característica de algoritmos que é o **bloco de repetição** ou **laço**[8]. O passo geral do algoritmo acima é

$P_{k+1} \leftarrow \text{dist\_metade}(P_{k-1}, P_k)$, para $k > 1$, sendo $P_0 = B$ e $P_1 = dist\_metade(A, B)$.

Assim, para efetuar a simulação utilizando dist_metade$(.,.)$ é necessário aplicá-la seguidas vezes. Entretanto, se na própria definição deste algoritmo incorporarmos uma chamada recorrente, a própria recorrência controla as múltiplas aplicações. Portanto, é natural imaginar que tal recurso também possa ser incorporado aos "scripts" na GD. A partir do "script" dist_metade podemos produzir um novo "script", de nome aquiles, anotando a repetição através de uma **recorrência** (ou **recursão**). A recorrência ocorre sobre os pontos B e F (nesta ordem), produzindo uma solução mais completa para o exemplo 3. Usando a construção 2, o "script" aquiles é definido da seguinte forma:

Construção 3 *aquiles(A, B): ponto A, ponto B*

1. $F \leftarrow$ *dist_metade(A, B)* *de acordo com a construção 2*
2. $G \leftarrow$ *aquiles(B, F)* *recorrência, aplicada aos pontos B e F*

Nesta construção, como não definimos uma condição para a aplicação da recorrência, é necessário definir de outra forma o número de vezes que a recorrência será aplicada. A solução adotada no iGeom é a mesma da versão 3 do GSP: quando o usuário for aplicar um "script" recorrente, aparece uma janela para que ele defina a profundidade de recorrência (conforme figura 11).

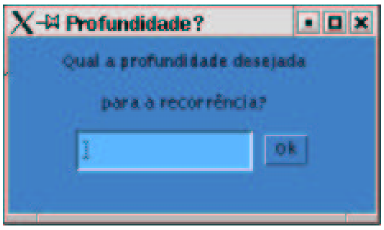


Figura 11: Janela disparada por "scripts" recorrentes

Para definir claramente o que é profundidade, precisamos de alguns outros conceitos associados à árvores. Resumidamente, uma **árvore** pode ser entendida como uma estrutura hierárquica, composta por nós (ou vértices) e arestas. Uma aresta tem como extremos um par de nós, e liga um nó superior a um de seus **filhos**. Um nó que não tem filho é denominado **folha** e o primeiro nó da árvore é sua **raiz** [Lucchesi et al., 1979, Szwarcfiter, 1984]. Na figura 12 aparece uma árvore binária, todos os nós tem no máximo 2 filhos, com 7 nós, 6 arestas[9] e 4 folhas.

Dizemos que um nó está no **nível** k da árvore, se existem k arestas ligando ele à raiz da árvore. Sendo H uma árvore com raiz r e tendo como filhos as raizes das árvores $\{H_i\}_{i \in I}$, então

$$nivel(H) = 1 + max\{nivel(H_i) : i \in I\}.$$

A **profundidade** de um nó é o caminho desde a raiz até este nó. Mas para computar o número de chamadas recorrentes, devemos examinar a profundidade de uma folha.

[8] Sequência de passos repetitivos, que constitui a parte central dos algoritmos, como por exemplo, a soma do "vai um" com os dígitos correspondentes na soma de dois números: $k = n+m$, com $n = n_3 n_2 n_1 n_0$ e $m = m_2 m_1 m_0$, então o laço é $k_i \leftarrow \text{resto\_inteiro\_de}\,\{(n_i + m_i + vai\_um)/10\}\,; vai\_um \leftarrow \text{quociente\_de}\,\{(n_i + m_i + vai\_um)/10\}$.

[9] Sendo VG e AG os conjuntos, respectivamente, dos nós e das arestas de uma árvore, então $\#VG = \#AG + 1$ (vide p.e., [Lucchesi et al., 1979]).

Como as recorrências no iGeom são uniformes (se existem k chamadas, nenhuma delas ou todas elas serão efetuadas), a árvore associada às recorrências é uniforme: cada nó interno (não folha) tem precisamente o mesmo número de filhos. Portanto, a profundidade de todas suas folhas são iguais, correspondendo ao nível em que se encontram.

Para ilustrar esta associação é melhor utilizar um "script" com mais de uma chamada recorrente, como no "script" abaixo.

Construção 4 *rec2*(A, B)*: ponto* A*, ponto* B

1. $F1 \leftarrow$ dist_metade(A, B)
2. $F2 \leftarrow$ dist_metade(B, A)
3. $G1 \leftarrow$ rec2$(B, F1)$ *(primeira recorrência)*
4. $G2 \leftarrow$ rec2$(A, F2)$ *(segunda recorrência)*

Na figura 12 ilustramos a aplicação do "script" recorrente rec2, que apresenta representados duas chamadas recursivas e resulta numa árvore binária. A aplicação representada é com profundidade 2.

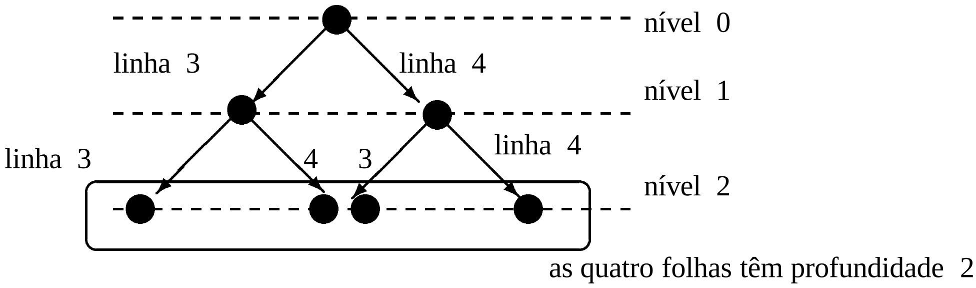


Figura 12: Árvore de recorrência, com dois níveis de chamada (profundidade 2)

Atualmente existem ao menos dois programas de GD que permitem "scripts" recorrentes, o iGeom e o GSP.

3.2. Algoritmos: Fractais construídos com programas de Geometria Dinâmica

O uso de recorrência em "scripts", nos programas de GD, permite uma elegante introdução ao conceito de algoritmo, sem a necessidade de explicar variáveis ou comandos tipo *for*. Além disso, o uso de recorrência agiliza a construção de **fractais**[10] geométricos [Mandelbrot, 1983, Kappraff, 1991], com os quais podemos explorar conceitos de progressões geométricas, somatórios e até de convergência. O exemplo 3, da corrida entre Aquiles e a tartaruga, pode ser utilizado para iniciar tal atividade, seguindo-se de algum exemplo graficamente mais interessante, como o fractal baseado em circunferências, descrito no exemplo 4.

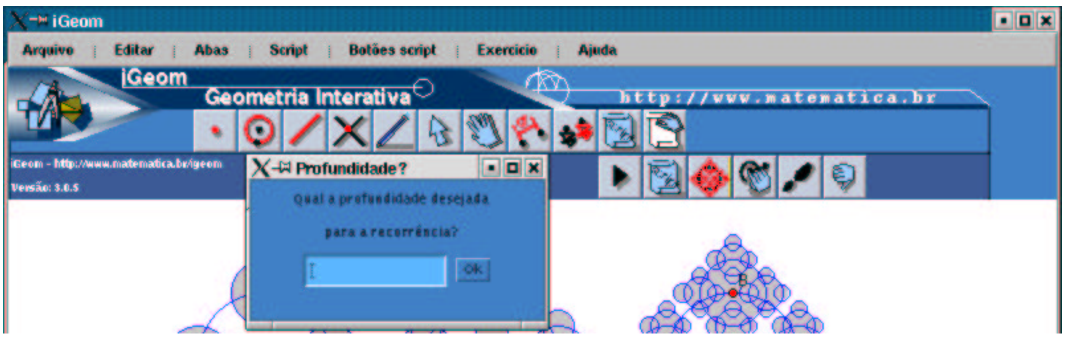


Figura 13: Interface do iGeom com a janela para profundidade em "scripts" recorrentes

Na figura 13 mostramos a tela do iGeom, destacando os botões associados à "scripts" (*executar*, *começar*, *recorrência*, *enviar*, *passo* e *cancelar*) e mostrando a janela em que o usuário escolhe a profundidade de recorrência a ser aplicada.

[10] O nome fractal é devido à Benoit Mandelbrot, derivada do latim *fractus*, que significa quebrado, partido ou irregular. Apesar de não existir um definição universalmente aceita para fractal, uma característica comumente aceita é a **auto-similaridade**, que significa que podemos reconhecer o todo da figura olhando apenas uma parte da mesma.

Exemplo 4 *Dados dois pontos A e B e uma circunferência $C0$ com centro A e passando por B, construir quatro novas circunferências centradas nos "pólos" de $C0$ e com raios equivalentes a metade do raio de $C0$. Utilizando cada uma das quatro novas circunferências, repetir o processo (recorrência).*

Construção 5 frac_circ$(A, B, C0)$
Entradas: ponto A, ponto B e circunferência $C0$, $C0$ com centro A e passando por B
Saídas: fractal a partir dos "pólos", com circunferência de raio metade

1.	$r \leftarrow reta(A, B)$	*reta passando por A e por B*
2.	$s \leftarrow reta\_perp(A, r)$	*reta perpendicular a r passando por A*
3.	$(C, D) \leftarrow int(C0, r)$	*C e D interseções entre $C0$ e r*
4.	$(E, F) \leftarrow int(C0, s)$	*E e F interseção entre $C0$ e s*
5.	$M_1 \leftarrow pto\_med(A, C)$	*M_1 ponto médio do segmento AC (implementado numa macro)*
6.	$M_2 \leftarrow pto\_med(A, F)$	*M_2 ponto médio do segmento AF*
7.	$M_3 \leftarrow pto\_med(A, D)$	*M_3 ponto médio do segmento AD*
8.	$M_4 \leftarrow pto\_med(A, E)$	*M_4 ponto médio do segmento AE*
9.	$C1 \leftarrow circ(B, M_1)$	*circunferência centrada em B, passando por M_1*
10.	$C2 \leftarrow circ(F, M_2)$	*circunferência centrada em F, passando por M_2*
11.	$C3 \leftarrow circ(D, M_3)$	*circunferência centrada em D, passando por M_3*
12.	$C4 \leftarrow circ(E, M_4)$	*circunferência centrada em E, passando por M_4*
13.	frac_circ$(B, M_1, C1)$	*recorrência aplicada à circunferência $C1$*
14.	frac_circ$(F, M_2, C2)$	*recorrência aplicada à circunferência $C2$*
15.	frac_circ$(D, M_3, C3)$	*recorrência aplicada à circunferência $C3$*
16.	frac_circ$(E, M_4, C4)$	*recorrência aplicada à circunferência $C4$*

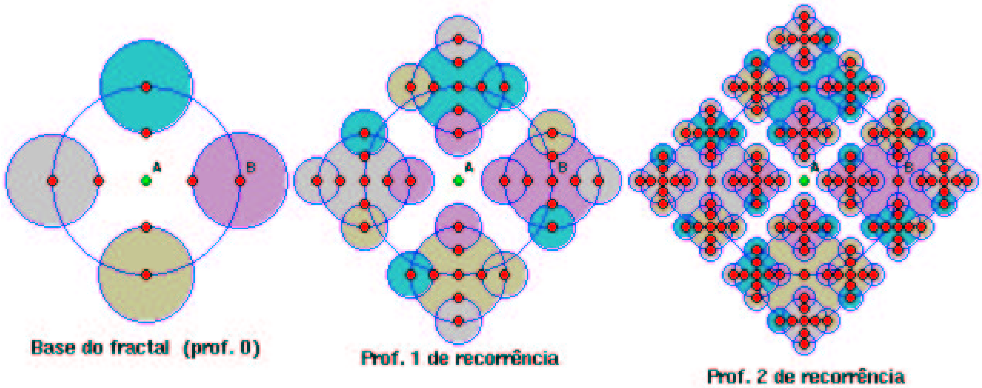


Figura 14: Fractal baseado em circunferências (nível de recorrência 0, 1 e 2)

Na figura 14 podemos observar três fases da construção do fractal produzido pelo "script" frac_circ: na esquerda aparece a base do fractal (profundidade de recorrência 0), no centro está o resultado de uma aplicação com profundidade 1 (nível 1: aplica-se uma vez para cada uma das 4^1 circunferências do nível 0); e na direita, a aplicação de frac_circ com profundidade 2 (nível 2: aplica uma vez para cada uma das 4^2 circunferências do nível 1).

Uma possível utilização destas figuras no ensino médio é trabalhar as macros como **caixas pretas**, sem explorar como elas foram obtidas, e discutir propriedades nas figuras geradas. No caso do "script" frac_circ, o professor pode fornecê-la ao aluno e solicitar:

1. *procurar contabilizar o número de circunferências em cada nível*;
   (termos de p.g.)
2. *procurar uma equação para, dado n, determinar o número de circunferências que se obtém com uma aplicação de* frac_circ *com profundidade n*;
   (soma de p.g.);
3. *procurar saber se a figura formada com níveis de aplicação cada vez maiores é limitada ou não*;
   (série convergente de p.g. de razão menor que 1)

4. *uma vez que a figura é limitada, procurar determinar para que figura converge*;
   (para um quadrado);
5. *procurar formalizar o porquê de um quadrado conter o fractal.*
   (provando que: é quadrado pois seus diâmetros encontram-se em ângulo de 90^0 e ambos tem mesmo comprimento - sendo R o raio de $C0$, o diâmetro do fractal é definido pela série convergente $4\sum_{i=0}^{\infty}(\frac{1}{2})^i R = 4R$)

Pode-se fazer o mesmo tipo de atividade com vários fractais conhecidos, como por exemplo, a **curva de Kock** (ou floco de neve) e o **triângulo de Sierpinski**, respectivamente representados nas figuras 15 e 16.

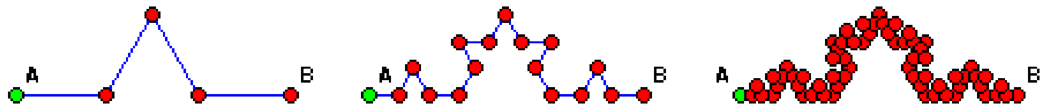


Figura 15: Três etapas de construção para a "curva de Kock" (nível de recorrência 0, 1 e 2)

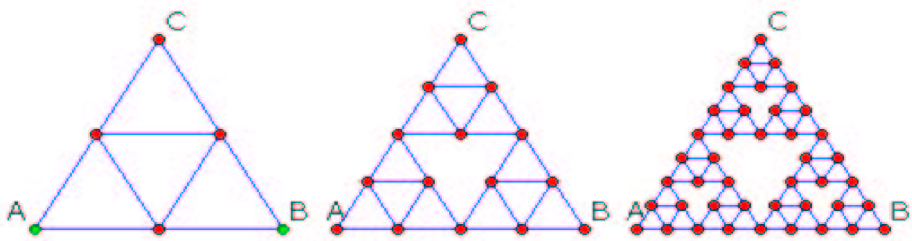


Figura 16: Três passos para construção do fractal "triângulo de Sierpinski" (níveis 0, 1 e 2)

Exemplo 5 *(Curva de Kock) Dados dois pontos A e B e um segmento $s0$, trocar o terço central do mesmo por dois lados de um triângulo equilátero e aplicar a recorrência nos quatro novos segmentos de comprimento igual a $1/3$ do original.*

Exemplo 6 *(Triângulo de Sierpinski) Dado um triângulo formado pelos pontos A, B e C construir M_1, M_2 e M_3, respectivamente, os pontos médios de $\overline{AB}$, $\overline{AC}$ e $\overline{BC}$. Construir os segmento ligando $\overline{M_1M_2}$, $\overline{M_1M_3}$ e $\overline{M_2M_3}$. Aplicar a recorrência aos novos triângulos AM_1M_2, BM_1M_3 e CM_2M_3.*

Note que no exemplo 5 existe um problema inicial que é dividir um segmento em três partes iguais. Para isso podemos utilizar uma técnica de desenho geométrico, tomando um cuidado adicional para que o "script" dependa apenas de dois pontos A e B, como na construção a seguir.

Construção 6 div_segmento(A, B)
Entradas: ponto A, ponto B
Saídas: pontos F e G tais que $F \in \overline{AB}$, $G \in \overline{AB}$ e $d(A,F) = d(F,G) = d(G,B) = \frac{d(A,B)}{3}$.

1.	$r \leftarrow reta(A, B)$	*reta passando por A e por B*
2.	$r0 \leftarrow reta\_perp(A, r)$	*reta perpendicular a r passando por A*
3.	$C0 \leftarrow circ(A, B)$	*circunferência centrada em A, passando por B*
4.	$C \leftarrow int_s(C0, r0)$	*C interseção "superior" entre $C0$ e $r0$*
5.	$C1 \leftarrow circ(C, A)$	*circunferência centrada em C, passando por A*
6.	$D \leftarrow int_s(C1, r0)$	*D interseção "superior" entre $C1$ e $r0$*
7.	$C2 \leftarrow circ(D, C)$	*circunferência centrada em D, passando por C*
8.	$E \leftarrow int_s(C2, r0)$	*E interseção "superior" entre $C2$ e $r0$*
9.	$s0 \leftarrow segm(E, B)$	*segmento entre E e B*
10.	$r1 \leftarrow reta\_par(s0, D)$	*reta paralela a $s0$ passando por D*
11.	$r2 \leftarrow reta\_par(s0, C)$	*reta paralela a $s0$ passando por C*
12.	$F \leftarrow int(r2, r)$	*F interseção entre $r2$ e r*
13.	$G \leftarrow int(r1, r)$	*G interseção entre $r1$ e r*

Existe uma infinidade de possíveis fractais para serem explorados, seja para o aluno tentar construir um “script” que o descreva ou descobrir propriedades usando a abordagem tipo caixa preta. Nas figuras 17 e 18 apresentamos dois tipos baseados em segmentos AB, abrindo ramos de comprimento igual a metade do original (árvores binárias). Neste tipo de exemplo (linhas conexas) um exercício interessante é encontrar o comprimento da curva no nível de recorrência k (soma de p.g.).

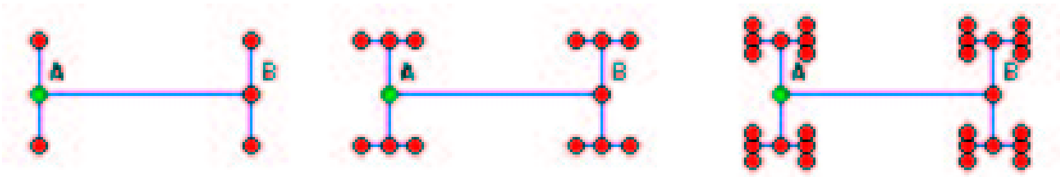


Figura 17: Um fractal tipo árvore binária, com ângulo de 90^o, níveis de recorrência 0, 1 e 2

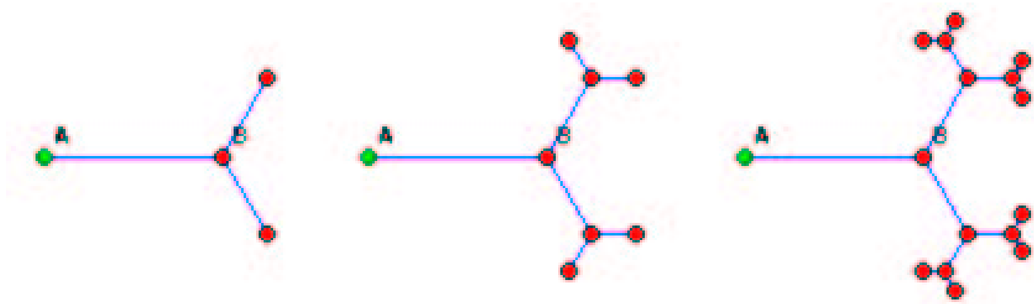
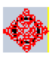


Figura 18: Um fractal tipo árvore binária, com ângulo de 120^o, níveis de recorrência 0, 1 e 2

3.3. “Scripts” recorrentes no iGeom

Em termos práticos, para construir um “script” deve-se utilizar os botões da barra secundária associada ao primário *Script* . Na figura 13 aparecem os atuais seis botões associados aos “scripts”, sendo eles:

1.  aplica um “script”, gravado como arquivo, usualmente com extensão `scr`;
2.  dispara o gravador de “script”, que a partir deste momento passa a anotar no “script” todas as construções realizadas (vide figura 19);
3.  durante a geração de um “script”, pode-se clicar neste botão para produzir uma chamanda recorrente (antes deve-se marcar os objetos em que a recorrência será aplicada);
4.  botão para disparar o avaliador automático de exercícios, posteriormente enviando os dados para o servidor adequado (vide seção 4);
5.  botão para anotar comentários a serem mostrados durante a aplicação do “script” (no ponto em que foi inserido);
6.  cancela a geração do “script”, fechando a janela da figura 19.

Lembrando que um “script” pode ser entendido como um algoritmo, este pode ter ou não parâmetros. Caso tenha parâmetros, para que possa ser aplicado, será necessário passar corretamente esta lista de parâmetros. Caso não tenha parâmetros, sempre que for dispardo ele produzirá sempre o mesmo resultado (e nesse sentido seria equivalente a abrir um arquivo do iGeom, com extensão `geo`).

Na geração do “script” o iGeom deduz os parâmetros a medida que as construção são anotadas, anotando-os no topo da janela de “script”, sob o título *Parâmetros:*.

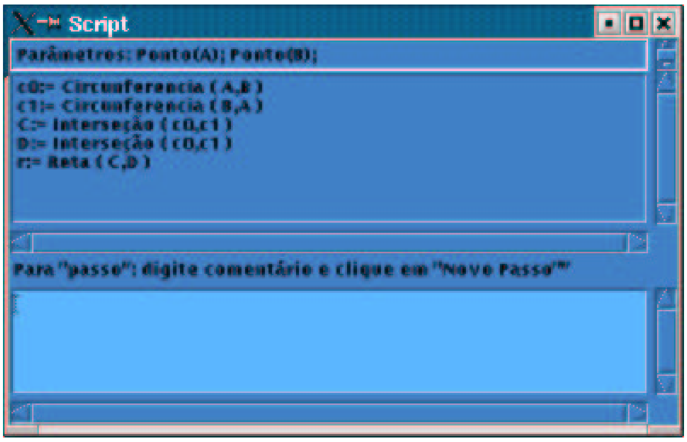


Figura 19: Janela de registro de “script” com um construtor de mediatriz

Para gerar um “script” deve-se, primeiro disparar o gravador de “scripts” com o botão do item 2  acima. A partir deste momento, todas as construções realizadas são anotadas no novo “script”, aparecendo na janela de “scripts”, ilustrada na figura 19. A dedução dos parâmetros funciona da seguinte forma: ao usar um objeto que não foi construído durante a geração do “script”, automaticamente este será inserido como parâmetro.

Para finalizar a geração, e gravar efetivamente o “script”, deve-se clicar novamente no mesmo botão que o disparou, apresentado no item 2 acima .

A recorrência pode ser obtida em qualquer “script” do mesmo modo que é obtida em funções matemáticas: durante a geração de um “script”,

- deve-se marcar os objetos sobre os quais deseja-se aplicar a recorrência
  (estes devem ser de mesmo tipo, e na mesma sequência, que os parâmetros do “script”, indicados como *parâmetros* na janela de “script”);
- clicar no botão “recorrência” (item 3 acima).

Se não houver erro na seleção dos parâmetros, ao clicar no botão de recorrência, será gerada uma instrução do tipo `Recorrência ( A,D )` na janela do “script”, onde `A` e `D` são os objetos marcados (compatíveis com os parâmetros deduzidos para o “script”).

Durante a geração de um “script”, o iGeom permite que seja disparado algum “script” já gravado. Nesse caso, todos os passos executados pelo segundo “script” será anotado no primeiro.

4. Autoria e avaliação automática de exercícios com o iGeom

Existem várias características que são importantes em sistemas de apoio ao ensino. Destacamos aqui duas delas, uma relacionada aos professores e outra relacionada aos alunos.

Para o professor, é importante que o sistema simplifique, de alguma forma, seu trabalho. Esta simplificação pode motivá-lo a enfrentar o aprendizado do sistema, que idealmente deve ser fácil.

Em relação ao aluno, além de outras questões didáticas, é importante que ele receba rapidamente o resultado de sua avaliação em cada atividade desenvolvida. Como observam [Hara and Kling, 1999, Kirby, 1999], a principal frustração do aluno nos cursos de educação a distância é a limitação ou a falta de uma avaliação (*feedback*) imediata.

Nesta seção apresentamos os recursos do iGeom para ambas problemas: a geração e publicação de exercícios pelo professor e a resolução e verificação imediata dos exercícios feitos pelos alunos. Além disso, apresentamos o recurso de comunicação do iGeom e como, a partir dele, é possível utilizar o iGeom em sistemas Web de ensino.

4.1. Autoria de Exercícios

O processo de autoria de exercício no iGeom possui cinco etapas: construção da solução, seleção dos objetos resposta, seleção dos objetos de entrada, desabilitar ferramentas e gravação do exercício.

O esquema de autoria de um exercício começa com a construção geométrica desejada, que servirá como **gabarito**. Isso é feito do mesmo modo que todas as outras construções. Uma vez pronta a construção o professor deve anotar o que o aluno receberá como enunciado e o que deverá marcar como sua resposta, utilizando a interface de autoria de exercícios.

A interface de autoria de exercício é bastante simples, contando com uma só janela (figura 20). Nesta janela o professor deve:

- marcar quais são os **objetos de entrada** (aqueles que o aluno vai receber como enunciado do exercício);
- marcar quais são os **objetos de saída** (aqueles que serão comparados aos correspondentes da resposta do aluno);
- determinar quais os botões que ficarão disponíveis para o aluno/usuário resolver o exercício.

Como a construção do exercício utiliza a mesma área de desenho que será utilizada em sua resolução, o professor pode deixar os objetos nas posições onde gostaria que o aluno os visualizasse, agilizando sua formatação e publicação.

Outro recurso do iGeom útil à confecção dos exercícios é a ferramenta para inserir textos. Este textos podem servir para introduzir comentários ou conter o próprio enunciado do exercício. Este recurso é bastante útil, mesmo que o sistema Web também disponha de uma área específica para expor o enunciado (como no SAW), pois assim armazena-se o enunciado do exercício no mesmo arquivo (facilitando seu uso futuro).

Vamos especificar melhor a forma de criação de um exercício a partir do exemplo da mediatriz.

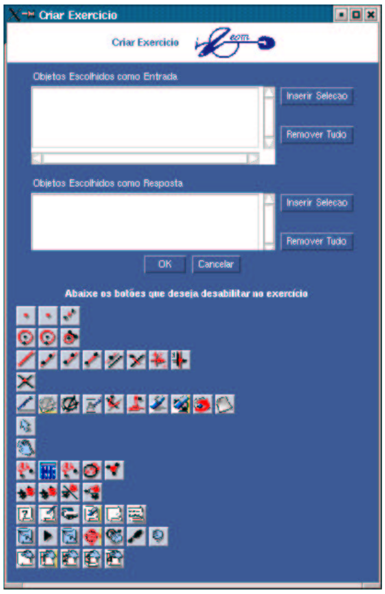


Figura 20: Janela para criar exercício

Exemplo 7 *Dados os pontos A e B, construir a reta r mediatriz de ambos.*

1. *Construir o gabarito*
   Fazer a construção da mediatriz de dois pontos, criando os pontos A, B, C e D, as circunferências $C0$ e $C1$ e a reta r (mediatriz), conforme a figura 21. Acrescentar um texto com enunciado e explicações adicionais.
2. *Anotar os objetos de entrada*
   Abrir a janela de criação de exercícios clicando inicialmente na opção *Exercício* da barra de menu, selecionando a sub-opção *Criar Exercício...*, como indicado na figura 23.

   Marcar os objetos de entrada, usando o ícone "marcardor" . Com este botão selecionado, clicar nos pontos A e B e também no texto de enunciado.
   Dentro da janela de criação de exercício, clicar no botão *Inserir Seleção* ao lado da área de "Objetos Escolhidos como Entrada".
3. *Anotar os objetos de resposta (saídas)*
   Marcar os objetos de saída de modo análogo. Com o ícone "marcardor" selecionado, clicar na reta r, correspondente à mediatriz.
   Dentro da janela de criação de exercício, clicar no botão *Inserir Seleção* ao lado da área de "Objetos Escolhidos como Resposta".

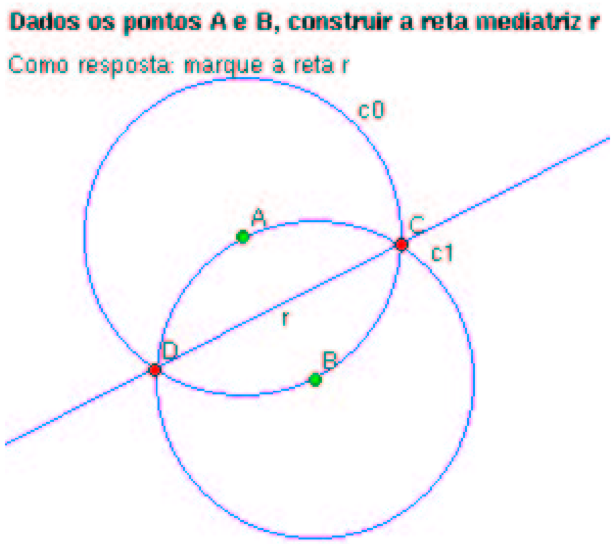


Figura 21: Gabarito para o exercício de mediatriz

4. *Selecionar os botões a serem disponibilizados ao aluno*
   Ainda na janela de criação de exercícios, pode-se clicar em todos os botões que não devem ser utilizados pelos aluno na resolução do exercício.
   Neste exemplo, pode-se eliminar os botões *Ponto Médio*, *Perpendicular* e *Paralela* para que o aluno utilize circunferências para gerar os pontos C e D, equidistantes à A e a B (figura 26). Com estes dois botões seria possível obter uma solução como na figura 27.
5. *Gravação do exercício*
   Clicar no botão *OK* da janela de exercício (figura 20. Ficarão visíveis na área de desenho apenas os objetos de entrada, sendo esta a forma que o aluno/internauta receberá o exercício.
   Pode-se gravar o exercício em dois formatos: formato legível pela versão "applet" do iGeom ou formato para a versão aplicativo. No primeiro caso utiliza-se a opção de menu *Arquivo*, seguida de *Gerar applet...*, gravando um arquivo com extensão `HTML`, legível por navegadores Web. O aluno, ao abrir este exercício via Web, verá apenas os objetos de entrada, não sendo possível visualizar os outros objetos construídos pelo professor. No segundo caso utiliza-se a opção de menu *Arquivo* seguida de *Gravar construções...* (ou de *Gravar como...*), gerando um arquivo com extensão `GEO`. Neste caso pode-se voltar a editar o arquivo na versão aplicativa do iGeom.

Na figura 22 está apresentada a interface do iGeom, com o menu *Criar Exercício...* em destaque.

Figura 22: Menu para criar exercício

Na figura 23, sobre a janela principal do iGeom, aparece a janela de criação de exercícios, com o gabarito do exemplo 7. Na janela de exercícios deve-se notar, de cima para baixo, a área em que estão listados os objetos de entrada selecionados, a área do objeto de resposta/saída (só a reta mediatriz) e no canto inferior, os botões que podem ser "abaixados" para que não apareçam no enunciado que o aluno receberá.

Na figura 24 aparece a barra de botões principal, com a opção de retas na barra secundária. Nela aparecem todos os botões relativos à retas atualmente disponíveis no iGeom. A figura 25 ilustra como

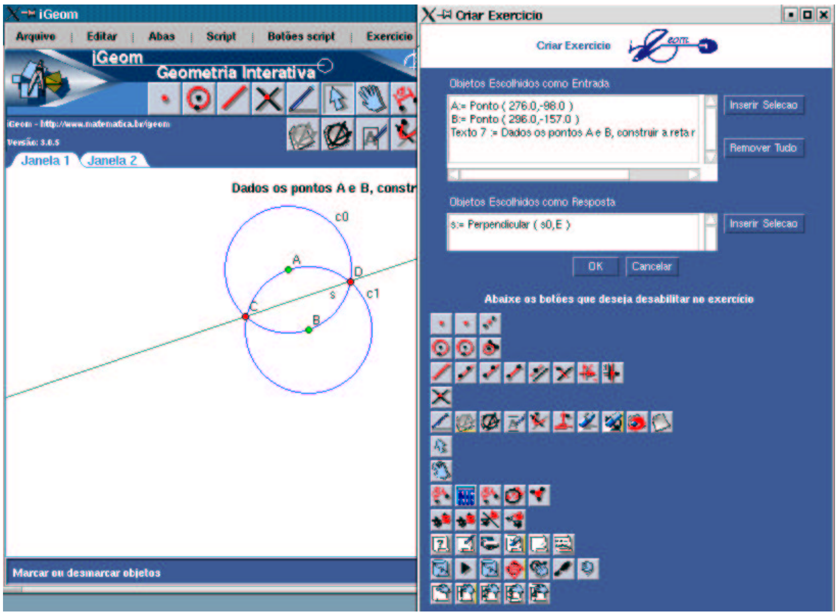


Figura 23: Janela para criação de exercício.

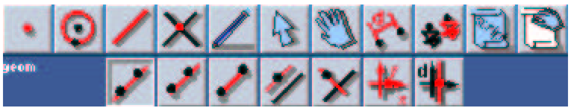


Figura 24: Menu de botões completos

Figura 25: Menu sem alguns botões

apareceria a barra secundária de retas num "applet" com o exercício do exemplo 7. Nela não aparecem os botões *Perpendicular* e *Paralela*, que forma eliminados no passo 4 do referido exemplo.

Na figura 26 aparece a construção do exemplo 7, com o gabarito do professor. A figura 27 apresenta a construção não desejada que utiliza os botões *Ponto Médio* e *Perpendicular*.

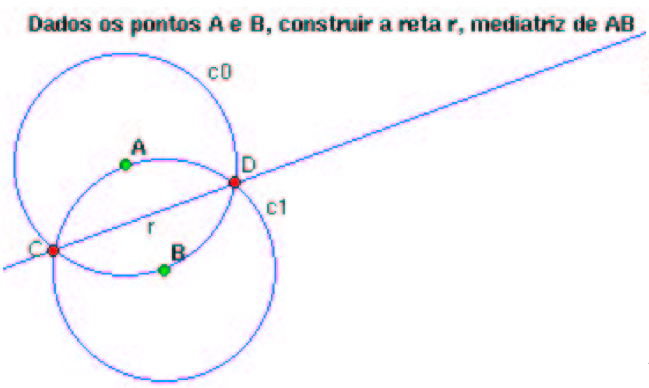


Figura 26: Gabarito para o exercício

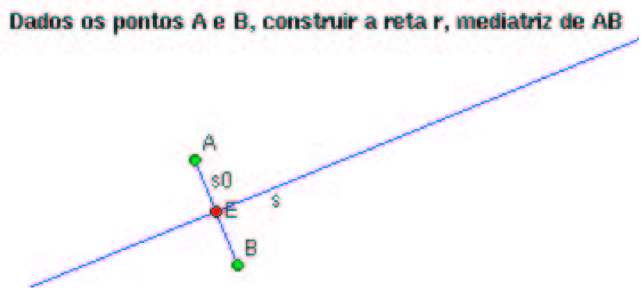


Figura 27: Mediatriz construída com "ponto médio"

4.2. Avaliação Automática

Associado ao recurso de autoria de exercício, foi também desenvolvido um recurso para avaliação automática da solução do aluno. A técnica empregada nesta avaliação permite:

1. detectar como corretas quaisquer soluções, mesmo que diferentes daquelas imaginadas pelo professor (no gabarito);

2. apresentar de imediato o resultado da avaliação;
3. fornecer um contra-exemplo se a resposta do aluno foi classificada como incorreta.

A solução que adotamos no iGeom é fortemente baseada na estrutura "dinâmica" da GD e parte do gabarito que o professor deve fornecer na geração do exercício. O aluno recebe este gabarito, mas só pode visualizar os objetos iniciais que descrevem o problema e não a resposta do professor. A avaliação é efetuada comparando-se os objetos-resposta do aluno com os correspondentes do gabarito do professor. Esta comparação é feita a partir de um critério de distância entre os objetos.

Quando o aluno abre uma página com o exercício produzido no exemplo 7, ele verá apenas os pontos A e B e mensagem `Dados os pontos A e B, construir a reta r, mediatriz de AB`. Após terminar sua construção o aluno deve marcar com o selecionador a reta que supõe resolver o problema e clicar no botão de envio de resposta , dentro das opções de "script".

Quando forem selecionados e enviados os objetos-resposta, o algoritmo de avaliação automática será iniciado. Caso o aluno selecione algum objeto de família geométrica diferente daquelas existentes no gabarito, o aluno receberá uma mensagem de erro específica para cada caso. Em não havendo este tipo de erro, o algoritmo de avaliação será disparado retornando: correto ou incorreto. Nos casos que o algoritmo retornar incorreto o aluno poderá ver um contra-exemplo, ou seja, uma configuração de sua construção em que ficar óbvio seu erro (de modo mais formal: as distâncias entre suas respostas e as respostas do gabarito do professor são maiores que um limite de tolerância).

O resultados poderá ser apresentados ao usuário para notificá-lo de qual foi a avaliação de seu exercício, viabilizando o uso desta ferramenta de forma auto-didata, pois o aluno pode fazer e refazer exercícios, obtendo a avaliação deste processo.

Um exemplo de curso introdutório de geometria, utilizando exercícios no iGeom, foi produzido por Sandra Cairolli como trabalho final da disciplina *Noções de ensino de matemática usando o computador* no IME-USP, no primeiro semestre de 2004. Este exemplo está no endereço http://www.matematica.br/igeom/docs/exemplo1.

4.3. Comunicação

O iGeom dispõe de recursos de comunicação para permitir seu uso em cursos "fechados", com turmas de alunos matriculados e com um servidor. A comunicação é feita através de uma conexão HTTP direta entre o *applet* e o servidor, sendo a troca de mensagens baseada no método POST [W3C, 2004]. A escolha dos protocolos HTTP e POST é devido à sua popularidade, estando disponível em qualquer plataforma.

Este recurso, em conjunto com os anteriormente apresentados, proporciona a um aluno, conectado ao servidor, descarregar um exercício criado pelo professor, resolvê-lo e solicitar sua avaliação. Ao fazer esta requisição, o iGeom avalia o exercício (localmente) e o envia resultado ao servidor (juntamente com mais algumas informações, por exemplo, com o instante de envio). Dentre as vantagens dessa implementação pode-se listar:

1. um professor pode criar/editar um exercício em sua própria máquina e depois enviá-lo para o servidor ou pode fazê-lo pela Internet através de um navegador com interpretador *Java*;
2. a possibilidade de adaptação da interface, habilitando ou desabilitando botões;
3. a correção automática dos exercícios, oferecendo contra-exemplos quando for detectado erro;
4. o envio da solução do aluno, sua avaliação e eventual contra-exemplo, para o servidor.

Além desses benefícios, a comunicação do iGeom com o servidor viabiliza análises quantitativas da interação do aluno com o programa. Por exemplo, é possível verificar quais as ferramentas mais utilizadas, os passos realizados para completar uma construção, as formas mais comuns na construção de objetos, e diversas outros dados de interação. Com estes dados é possível melhorar a interface do programa e fornecer um guia para a criação de novas funcionalidades.

5. Páginas Web Interativas com o iGeom

Os computadores aliados às tecnologias de comunicação produziram a Internet e A World Wide Web, gerando um mundo digital sem fronteiras. Estas tecnologias podem auxiliar o processo de ensino/aprendizagem, abrindo portas para a criação de novas metodologias no ensino. A presença destas tecnologias em instituições de ensino e sua utilização como ferramenta de auxílio e de pesquisa é bastante estudada. Por exemplo, em [Castro (1996)], são apresentados vários fatores positivos da Internet, entre eles podemos destacar o acesso imediato a informação atualizada e a interface gráfica facilmente reconhecível e maleável surgida com a criação da World-Wide Web.

Hoje existem vários sistemas complexos para gerenciar cursos pela Web, inclusive alguns gratuitos, como o Teleduc [Rocha, 2004] e o Moodle [Dougiamas, 2004]. Entretanto estes ambientes são desprovidos de recursos específicos para a produção e utilização de aprendizado de conteúdos específicos, como a geometria.

Nesta seção destacaremos a utilização do iGeom como recurso para incrementar a interatividade para o aluno, em páginas Web, e como facilitador de geração de conteúdo por parte do professor. Com ele podemos criar conteúdo interativo, nos quais objetos podem ser livremente manipulados ou exercícios podem ser realizados diretamente na Web. Além disso, o resultado desta interação pode ser enviada ao servidor sem a necessidade do usuário possuir a versão aplicativo do programa. Isto faz do iGeom uma ferramenta muito útil na preparação de material que pode ser utilizado em cursos à distância.

5.1. Desenvolvimento de Páginas Interativas

A exportação de construções para Web, incluindo exercícios, no iGeom é muito simples. Uma vez criada a construção/exercício no iGeom aplicativo, basta clicar em *Arquivo* na barra de menus e selecionar a opção *Gerar applet...*. Essa opção irá gerar uma página HTML contendo uma *tag* para importar o "applet" do iGeom (arquivo de nome `iGeom.jar`). Os dados relativos à construção estão codificados nos parâmetros desta *tag*.

Para publicar a página HTML na Web, é necessário colocar esta página em algum diretório do servidor Web, acessível pela Internet. Neste mesmo diretório deve-se colocar também uma cópia da versão "applet" do iGeom, seu arquivo `iGeom.jar` (que pode ser obtido em: *http://www.matematica.br/igeom/iGeom.jar*).

A página gerada pelo iGeom, ao ser visualizada por um navegador Web apresentará o formato padrão das páginas do sítio **iMática** (*http://www.matematica.br*), mas sem as figuras. Entretanto, não é difícil para o usuário conseguir transportar sua construção para uma outra página HTML qualquer. Basta transferir um pedaço de texto HTML gerado pelo iGeom, da seguinte forma:

1. Abrir o arquivo gerado pelo iGeom em um editor de texto como o **Emacs** no **Linux/Unix** ou **Notepad** (*Bloco de notas*) no **Windows**. Abra também o outro arquivo HTML no qual deseja inserir sua construção.
2. Identifique os comandos relativos ao iGeom, como apresentado na figura 28.
3. Copiar este código, que está entre `<applet ... </applet>` (incluindo estas *tags*), para o arquivo de destino.
4. Gravar o arquivo modificado.

5.2. A Comunicação do iGeom com o Servidor

Além dos recursos anteriores, o iGeom ainda dispõe de um recurso de comunicação que simplifica muito as atividades do professor. Se o iGeom estiver integrado a um gerenciador de cursos pela Web:

1. o professor pode produzir seus exercícios diretamente no sistema gerenciador, utilizando todos os outros recursos que este pode proporcionar, além do próprio iGeom;
2. o aluno, ao resolver os exercícios diretamente na Web, o gerenciador pode automaticamente arquivar sua solução e o resultado da avaliação automática (dentre outras informações que o iGeom ou o gerenciador pode produzir).

```
<applet codebase="." archive="iGeom.jar" code="IGeomApplet.class"
        WIDTH=720 HEIGHT=590 alt=" ...">
 <param name="BGCOLOR" value="#eeffee">
 <param name="igeom" value="# igeom: http://www.matematica.br!
[iGeom versão 3.0.5]!
[ versao: 3.0.5 ]!
[[21-Oct-04 2:13:21 PM; leo]]!
[0:1.1, 1:7, 2:0] - iGeom versão 3.0.5!
[21-Oct-04 2:13:21 PM; leo]!
{1:0, 0:0, 2:285.0 -210.0, 3:2 3, 4:A 3.141592653 0, 6:1, 5:-16711936, 7:0}!
{1:1, 0:0, 2:378.0 -241.0, 3:2 7 3, 4:B -0.0 0, 6:1, 5:-16711936, 7:0}!
{1:2, 0:3, 2:0 1, 3:4 7 5, 4:c0 2.3334711293507038 0, 6:1, 5:-16776961, 7:0}!
{1:3, 0:3, 2:1 0, 3:4 7 5, 4:c1 0.7853981633974483 0, 6:1, 5:-16776961, 7:0}!
{1:4, 0:1, 2:2 3 1, 3:7 7 6, 4:C 0.7853981633974483 0, 6:1, 5:-65536, 7:0}!
{1:5, 0:1, 2:2 3 2, 3:6, 4:D 0.7853981633974483 0, 6:1, 5:-65536, 7:0}!
{1:6, 0:4, 2:4 5, 3:, 4:r 0.5 0, 6:1, 5:-16776961, 7:0}!
"></applet>
```

Figura 28: Código gerado ao exportar a construção da mediatriz, na figura 26.

O professor ainda pode definir, para cada exercício, se o aluno vai receber de imediato o resultado de sua avaliação.

Atualmente, o iGeom tranfere a solução do aluno, o resultado da avaliação do exercício (certo/errado), um contra-exemplo (caso a solução esteja incorreta) e outros dados como a interação do aluno com as ferramentas de construção de objetos geométricos (pontos, retas, etc).

O método de transferência de dados do iGeom para o servidor é feita através de uma conexão HTTP utilizando o método POST. Para que o envio de mensagens ocorra corretamente é necessário inserir um parâmetro[11] na página HTML, definindo qual o endereço Web do servidor que receberá as respostas. O formato é:

```
<param name="enderecoPOST" value="http://aqui_vai_o_endereço">
```

Na figura 29 está indicado um exemplo do uso deste parâmtro, supondo que o tratamento será feito por um servidor que utiliza a linguagem PHP (como no sistema SAW [Brandão et al., 2004]).

```
<applet codebase="." archive="iGeom.jar" code="IGeomApplet.class"
        WIDTH=720 HEIGHT=590>
<param name="BGCOLOR" value="#eeffee">
<param name="enderecoPOST"
       value="http://milanesa.ime.usp.br/mac118/resolucao1.php?user=...">
<param name="igeom" value="# igeom: http://www.matematica.br!
[iGeom versão 3.0.5]!
[ versao: 3.0.5 ]!
[[21-Oct-04 2:46:33 PM; leo]]!
[0:1.1, 1:2, 2:0] - iGeom versão 3.0.5!
[21-Oct-04 2:46:33 PM; leo]!
{1:0, 0:0, 2:204.0 -147.0, 3:, 4:A 0.7853981633974483 0, 6:1, 5:-16711936, 7:0}!
{1:1, 0:0, 2:240.0 -163.0, 3:, 4:B 0.7853981633974483 0, 6:1, 5:-16711936, 7:0}!
"></applet>
```

Figura 29: Exemplo de "tag" para inserção do iGeom com o parâmetro *enderecoPOST*

Observe que neste caso o programa em PHP, localizado no servidor, deve ser o responsável pelo tratamento adequado dos dados recebidos. Para tanto, é necessário que o programa possua as seguintes variáveis:

[11] O parâmetro nada mais é que uma *tag* HTML que fornece algum dado para a applet.

$envWebValor: Esta variável indica o resultado da avaliação do exercício. Caso seu valor seja igual a 0 (zero) então o exercício foi avaliado como incorreto, caso seja igual a 1 (um) o exercício foi avaliado como correto.

$envWebArquivo: Toda a construção realizada pelo usuário na resolução de um exercício é inserida nesta variável. Por questões de segurança as construção são enviadas criptografadas.

$envWebGeoResp: O valor desta variável indicará quais foram os objetos da construção selecionados como resposta para um determinado exercício.

$envWebGeoOuvidor: Nesta variável são inseridos dados da interação do usuário com a interface do iGeom.

Todas as variáveis apresentadas anteriormente são inicializadas quando o iGeom requisita uma chamada ao servidor, por exemplo, após a avaliação de um exercício. Após isso, o programa em PHP pode armazenar os dados recebidos em um banco de dados (BD) (por exemplo, usando o gerenciado de BD MySQL) e retornar uma nova página ao usuário utilizando o seguinte comando (em PHP):

```
print "http://aqui_vai_o_endereço";
```

Na seção seguinte, apresentaremos um sistema que aplica os recursos apresentados e pode ser usado no desenvolvimento de cursos interativos de geometria. O esse sistema faz algumas adaptações de apresentação ao usuário, a partir da análise das interações do usuário/aluno com os exercícios realizados no iGeom.

5.3. Utilizando o iGeom em Cursos à Distância

Com a crescente expansão da Internet e do uso de computadores no ensino existe uma grande demanda por cursos via Web, e conseqüentemente, por ferramentas que facilitem sua produção e reutilização. Além disso, algumas características como a flexibilidade e a interatividade do sistema são essenciais para viabilizar aulas mais dinâmicas e interessantes.

A partir destas idéias, no final de 2003 foi iniciado o desenvolvimento, no IME-USP, de um sistema de aprendizagem pela Web, **Sistema de Aprendizagem pela Web** (SAW). Este projeto utiliza uma arquitetura cliente/servidor e é coordenado pelo primeiro autor. Apesar do SAW ter sido iniciado utilizando exclusivamente o iGeom, sua arquitetura prevê o acoplamento de qualquer módulo de aprendizagem, do tipo "applet", seguindo as conveções de parâmetros acima expostas.

O SAW está sendo desenvolvido em *PHP*, utilizando o gerenciador de banco de dados *MySQL*. No momento ela já dispõe de um segundo módulo de aprendizagem, para cursos introdutórios de programação. Este "applet" ainda está em fase inicial, mas já tem uma versão disponível na Internet no endereço *http://www.matematica.br/programas/icg*).

Uma primeira versão do SAW foi usada no primeiro semestre de 2004, em uma disciplina obrigatória do curso de licenciatura em matemática do IME-USP, *Noções de Ensino de Matemática Usando Computador* (MAC118). O sistema foi usado por três turmas, com 2 professores, 3 monitores e mais de 150 alunos. Na figura 30 está a atual interface do SAW na parte em que o exercício é apresentado ao aluno, no caso um exercício usando o iGeom.

Em edições anteriores de MAC118, todos os exercícios eram realizados utilizando o programa iGeom, mas sua correção era feita manualmente pelos monitores e professores. Devido ao número de alunos, a correção consumia grande parte do tempo dos monitores e o resultado da correção do exercício era entregue ao aluno duas ou três semanas após a realização do mesmo. Eram aplicados cerca 20 exercícios por semestre. Com o desenvolvimento do SAW e o uso da ferramenta de avaliação automática do iGeom, além de reduzir o trabalho de professores e monitores, foi possível aplicar mais de 40 exercícios com a apresentação imediata da avaliação, além de permitir que os exercícios fossem realizados via Internet.

6. Conclusão

Neste texto mostramos que a geometria pode ser examinada do ponto de vista de algoritmo e com a ajuda dos programas de geometria dinâmica (GD) podemos introduzir confortavelmente este conceito.

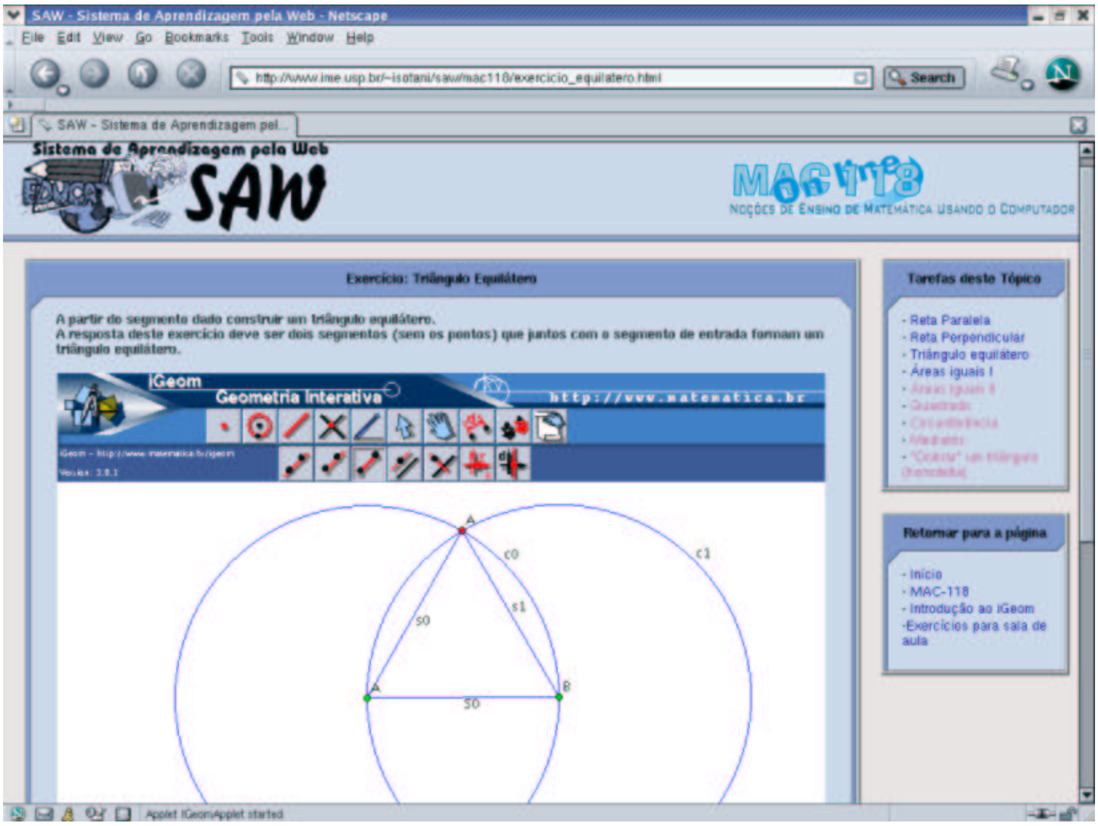


Figura 30: Resolvendo Exercícios no SAW+iGeom

Além disso, se empregarmos uma abordagem tipo caixa preta, podemos trabalhar com alunos do ensino médio, fornecendo-lhes as macros recorrentes prontas e solicitando tarefas exploratórias sobre número de objetos e, no caso de fractais, qual o formato do mesmo (para que figura converge). Foi mostrado como é possível descrever construções de fractais geométricos e também como pode-se efetivamente obter "scripts" construtores de fractais no iGeom.

Também discutimos aspectos relacionados a produtividade proporcionada por computadores e pela World Wide Web, particularmente com a geometria dinâmica. Mostramos como o iGeom pode ajudar professores e alunos em suas tarefas. Por exemplo, como o iGeom pode reduzir o trabalho os professor na preparação de exercícios para Web e, principalmente, se o iGeom estiver integrado a um sistema cliente/servidor, todo o trabalho de avaliação de exercícios é feito automaticamente. Nesta mesma aplicação, mostramos que também o aluno pode ter uma vantagem pois a avaliação é imediata, o que evita algumas frustrações.

Outro ponto importante aqui apresentado é que todos estes sistemas estarão a disposição na Web, para serem descarregado sem qualquer custo. O iGeom está disponível a mais de um ano.

Referências

[Arnold and Gosling, 1996] Arnold and Gosling (1996). *The Java Programming Language*. Addison-Wesley.

[Brandão, 2002] Brandão (2002). Algoritmos e fractais com programas de gd. *Revista do Professor de Matemática*, 49(1):27–32.

[Brandão, 2004] Brandão (2004). Programação geometrica: Uso da geometria dinâmica para programação. In *Anais do Segundo Colóquio de História e Tecnologia no Ensino de Matemática*, pages 191–202.

[Brandão and Isotani, 2003] Brandão and Isotani (2003). Uma ferramenta para ensino de geometria dinâmica na internet: igeom. In *Anais do Workshop sobre Informática na Escola*, pages 1476–1487. XXIII Congresso da Sociedade Brasileira de Computação.

[Brandão et al., 2004] Brandão, Isotani, and Moura (2004). A plug-in based adaptive system: Saaw. In *Intelligent Tutoring Systems*. Spinger-Verlag.

[Dougiamas, 2004] Dougiamas (2004). *Ambiente de Ensino a Distância*. http://moodle.com.

[Gravina, 1996] Gravina (1996). Geometria dinamica - uma nova abordagem para o aprendizado da geometria. In *Anais do VII Simposio Brasileiro de Informática na Educação*, pages 1–13.

[Grothman, 1999] Grothman (1999). *C.A.R - Compass And Rules*. http://mathsrv.ku-eichstaett.de/MGF/homes/grothmann/java/zirkel/.

[Guimarães et al., 2002] Guimarães, Barbastefano, and Belfort (2002). Tools for synchronous distance teaching in geometry. In *Proceedings of Second International Conference on the Teaching of Mathematics*, http://www.math.uoc.gr/ ictm2/Proceedings/pap385.pdf.

[Hara and Kling, 1999] Hara and Kling (1999). Student's frustrations with a web-based distance education course. *First Monday: Journal on the Internet*, 4(12). available at http://www.firstmonday.dk/issues/issue4_12/hara/.

[Isotani and Brandão, 2001] Isotani and Brandão (2001). imática: Ambiente interativo de apoio ao ensino de matematica via internet. In *Anais do Workshop sobre Informática na Escola*, pages 533–543. XXI Congresso da Sociedade Brasileira de Computação.

[Jackiw, 1995] Jackiw (1995). *The Geometer's Sketchpad v3.0*. Key Curriculum Press, Berkeley.

[Java, 2004] Java (2004). *Linguagem de Programação Java*. Sun Microsystems, http://java.sun.com/.

[Kappraff, 1991] Kappraff (1991). *Connections: the geometric bridge betwwen art and sciense*. Mc Graw-Hill.

[King and Shattschneider, 1997] King and Shattschneider (1997). *Geometry Turned On - Dynamic Software in Learning, Teaching and Research*. Mathematical Association of America, Washington.

[Kirby, 1999] Kirby (1999). Student's frustrations with a web-based distance. In *Technology and Teacher Education Annual*, pages 199–205.

[Laborde and Bellemain, 1997] Laborde and Bellemain (1997). *Cabri Geometry II*. Texas Instruments, Dallas.

[Lucchesi et al., 1979] Lucchesi, Simon, Simon, Simon, and Kowaltowsk (1979). *Aspectos Teóricos da Computação*. Projeto Euclides - IMPA, Rio de Janeiro.

[Mandelbrot, 1983] Mandelbrot (1983). *The fractal geometry of nature*. W.H. Freeman, New York.

[Mysql, 2004] Mysql (2004). *Banco de Dados Mysql*. Mysql AB, http://www.mysql.com.

[Papert, 1985] Papert (1985). *Logo: Computadores e Educação*. Brasiliense.

[Papert, 1994] Papert (1994). *A Máquina das Crianças: repensando a escola na era da informática*. Artes Médicas, Porto Alegre.

[Papert, 1999] Papert (1999). *Mindstorms: children, computer and powerful ideas*. Basic Books, New York, second edition edition.

[PHP, 2004] PHP (2004). *Linguagem de Programação PHP*. The PHP Group, http://www.php.net.

[Piesigilli, 2002] Piesigilli (2002). *Ferramentas para o ensino presencial e on-line de Geometria Descritiva*. http://www.matematica.br/igeom/historico/index.html.

[Richter-Gebert and Kortenkamp, 1999] Richter-Gebert and Kortenkamp (1999). *The Interactive Geometry Software Cinderella*. Springer-Verlag, Heidelberg.

[Rocha, 2004] Rocha (2004). *Ambiente de Ensino a Distância*. http://teleduc.nied.unicamp.br/teleduc/.

[Sahara, 2001] Sahara (2001). *Implementação de um Programa de Geometria Dinâmica em Java*. http://www.matematica.br/igeom/historico/index.html.

[Szwarcfiter, 1984] Szwarcfiter, J. L. (1984). *Grafos e algoritmos computacionais*. Editora Campus, Rio de Janeiro.

[Ávila, 1999] Ávila (1999). Os paradoxos de zenão. *Revista do Professor de Matemática*, 1(30).

[W3C, 2004] W3C (2004). *HyperText Markup Language (HTML)*. http://www.w3.org/MarkUp/.