

# Uma ferramenta para ensino de Geometria Dinâmica na Internet: iGeom

Leônidas de Oliveira Brandão<sup>1</sup>, Seiji Isotani<sup>1</sup>

<sup>1</sup>Instituto Matemática e Estatística – Universidade de São Paulo (USP)  
Caixa Postal 66.281 – 05315-970 – São Paulo – SP – Brazil

{leo,isotani}@ime.usp.br

***Abstract.** In this work we discuss the central role played by Dynamic Geometry (DG) in the context of teaching Mathematics and Geometry aided by computer. We also present a program of DG, the iGeom, and some guidelines of how iGeom can contribute to this effort. One of its contribution is that iGeom can be freely downloaded from Web. Other contribution that we discuss is how geometric solutions can be viewed as algorithms, and how geometric algorithms, with repetitive steps, can be implemented in an automatic fashion. It is important to note that iGeom is already in use as supporting tool for disciplines and in the Institute of Mathematics and Statistics of the University of São Paulo.*

***Resumo.** Neste artigo discutiremos o papel de destaque que a Geometria Dinâmica (GD) tem adquirido no contexto do ensino da Matemática e da Geometria assistida por computador e apresentaremos um programa de GD, o iGeom, e como este programa pode contribuir neste esforço. A primeira contribuição é a sua forma de distribuição, o iGeom pode ser descarregado gratuitamente da Web. Outra contribuição que discutiremos é como soluções geométricas podem ser vistas como algoritmos e como é possível implementar algoritmos geométricos com laços repetitivos, de modo automático, no iGeom. Vale ressaltar que o iGeom já está sendo utilizado em cursos e disciplinas do Instituto de Matemática e Estatística da Universidade de São Paulo.*

## 1. Introdução

O nome “**Geometria Dinâmica**” (GD) hoje é largamente utilizado para especificar a Geometria implementada em computador, a qual permite que objetos sejam movidos mantendo-se todos os vínculos estabelecidos inicialmente na construção. Este nome pode ser melhor entendido como oposição à geometria tradicional de régua e compasso, que é “estática”, pois após o aprendiz realizar uma construção, se ele desejar analisá-la com alguns dos objetos em outra disposição terá que construir um novo desenho.

Em função desta possibilidade de alterar objetos preservando-se a construção, podemos dizer que a GD é uma geometria do tipo **1-construção, N-testes**, enquanto a tradicional de régua e compasso é do tipo 1-construção, 1-teste. Esta é, para nós, a grande vantagem da GD sobre a geometria tradicional, pois permite que o aluno teste conjecturas e procure descobrir propriedades.

Nesta situação, o aspecto dinâmico desencadeia um processo desafiador e interessante de ensino e aprendizagem [Isotani e Brandão 2001]. As explorações e

estratégias que vão se delineando ao longo das tentativas de solucionar o problema são similares às que acontecem no ambiente de pesquisa de um matemático profissional. Esta postura investigativa contribui para a formação de uma concepção sobre matemática diferente daquela construída, usualmente, ao longo da vida escolar [Gravina 1996].

## 2. Geometria Dinâmica

A discussão sobre vantagens/desvantagens pedagógicas entre as duas formas de se “fazer” geometria pode ser conduzida sob diferentes pontos de vista. Nesta seção abordaremos apenas a interatividade e visualização como mecanismos facilitadores da aprendizagem na GD.

O uso da GD no ensino da Geometria traz boas possibilidades de mudança em uma área que vem sendo relegada ao segundo plano no ensino fundamental e médio, quando ensinada. Como observa o professor Hygino Domingues na apresentação de [Lindquist e Shulte 1998]: “Que não são só os professores de Matemática de países subdesenvolvidos, por exemplo, que fogem da geometria; que o temor da geometria também aflige o aluno dos países ricos...”.

De um lado notamos problemas na forma tradicional de se ensinar geometria, como nota [Gravina 1996]: “Os livros escolares iniciam o ensino de Geometria com definições, nem sempre claras, acompanhadas de desenhos bem particulares, os ditos desenhos prototípicos. Por exemplo, quadrados com lados paralelos às bordas da folha de papel, retângulos sempre com dois lados diferentes, altura em triângulos sempre acutângulos, entre outros. Isto leva os alunos a não reconhecerem desenhos destes mesmos objetos quando em outra situação. E mais, os alunos passam a acreditar que a posição relativa do desenho ou seu traçado particular façam parte das características do objeto, o que os leva a estabelecer desequilíbrios na formação dos conceitos. O aspecto de construção de objetos geométricos raramente é abordado. Dificilmente encontramos no livro escolar a instrução “*construa*”, e no entanto, esta é uma das atividades que leva o aluno ao domínio de conceitos geométricos.”

Por outro lado temos o potencial interativo e aberto de um programa de GD. Como nota Valente [Valente 2001], discutindo a linguagem **Logo** [Papert 1999], uma “característica relevante da visão do computador como ferramenta é o ambiente aberto, ou seja, o aprendiz é livre para propor e resolver qualquer projeto que tenha interesse.” Do ponto de vista do aprendizado, também podemos notar vantagem da GD sobre a estática. Usando o modelo de aprendizado de Geometria proposto pelos van Hiele [Lindquist e Shulte 1998]<sup>1</sup>, que classificam os níveis cognitivos de Geometria em cinco, visualização, análise, dedução informal, dedução formal e rigor. Notamos que a Geometria Dinâmica pode ser bem empregada nos três primeiros níveis. Nestes níveis iniciais o estudante está começando a abstrair conceitos matemáticos e deste modo a experimentação pode contribuir muito. Como observa [Gravina 1996], a Geometria Dinâmica evidencia uma nova abordagem ao aprendizado geométrico, onde conjecturas

---

<sup>1</sup> Este modelo é anterior ao surgimento da GD, que aparece em 1987, enquanto o trabalho principal dos van Hiele é da década de 50 [Veloso 1998, Lindquist e Shulte 1998].

são feitas a partir da experimentação e criação de objetos geométricos. Deste modo, podemos introduzir o conceito matemático dos objetos a partir do retorno gráfico oferecido pelo programa de GD, surgindo naturalmente daí o processo de argumentação e dedução.

No ensino tradicional (no qual o aluno apenas “ouve”), o aprendiz não é incentivado a ter uma postura investigativa (ativa), não sendo desafiado a construir seu próprio conhecimento. Em uma aula de matemática tradicional o professor enuncia conceitos, definições e propriedades que, muitas vezes, são apenas memorizados e futuramente reproduzidos pelo aluno sem sua devida compreensão. Segundo [Melo et al 2000], se o aluno agir ativamente, modificando características de vários objetos matemáticos, ele aprenderá pesquisando, relacionando as modificações feitas, analisando e verificando o que ocorre genericamente.

Neste contexto, a geometria dinâmica nos oferece uma nova proposta que visa explorar os mesmos conceitos da geometria clássica, porém, através de um programa interativo [Rodrigues 2002]. Assim é possível disponibilizar representações gráficas de objetos geométricos que aproximam o objeto material da tela do computador (desenho) ao objeto teórico (figura), favorecendo o desenvolvimento de uma leitura geométrica dos desenhos por parte do aprendiz, contornando, assim, uma das dificuldades do ensino da Geometria [Braviano e Rodrigues 2002].

Segundo [King e Schattshneider 1997], destacam-se como principais benefícios e aplicações de um sistema computacional de Geometria Dinâmica: a prova de teoremas, a precisão e visualização, a explorações e descobertas, as transformações e lugares geométricos e, por fim, a simulação de micromundos:

- **Prova de teoremas:** embora a Geometria Dinâmica não possa provar teoremas, a capacidade de experimentação de hipóteses pode motivar a busca pela prova de um teorema, pois induz à convicção de sua validade. Da mesma forma, pode ajudar e sugerir caminhos para a prova formal.
- **Precisão e visualização:** a construção da geometria é feita pelo estabelecimento de relações geométricas entre os elementos (perpendicularismo, paralelismo, pertinência, ângulo, etc). Pode-se medir ângulos e distâncias e calcular relações com precisão, permitindo facilmente a verificação empírica de hipóteses e teoremas. Os conceitos de um teorema podem ser compreendidos por visualização. Adicionalmente, a precisão também é importante porque construções imprecisas podem conduzir o aluno a conclusões errôneas.
- **Exploração e descoberta:** a manipulação de construções permite que se explore a Geometria e que “novas” relações e propriedades sejam descobertas. Muitas vezes, os próprios alunos “re-descobrem” teoremas em sala de aula.
- **Transformações e lugares geométricos:** pela sua capacidade de realizar transformações em figuras geométricas, programas de Geometria Dinâmica são ideais para o estudo de isometrias, similaridades e outras funções. Animando figuras e traçando lugares geométricos de pontos pré-definidos, estes aplicativos também podem explicitar problemas e propriedades normalmente não abordadas na literatura por sua inerente dificuldade.

- **Simulação e micromundos:** indo muito além da abstração da Geometria, as simulações que podem ser construídas com programas de Geometria Dinâmica permitem ilustrar conceitos de cinemática e óptica, entre outros. Por outro lado, oferecem também a possibilidade de criação de micromundos geométricos, a exemplo daqueles concebidos no âmbito da linguagem **Logo** [Papert 1999]. Neles, o aluno pode vivenciar experiências geométricas, algumas pré-concebidas pelo professor e muitas outras descobertas ao acaso, através da exploração interativa e de sua criatividade.

Vale observar que o comentário de [King e Schattshneider 1997] sobre “prova de teoremas” na GD, pode começar a ser explorado com aprendizes que estejam desde o segundo nível de compreensão dos propostos por van Hiele, a análise. Quando o aprendiz está neste nível, começa a analisar os conceitos geométricos envolvidos, por exemplo, podendo usar a experimentação para discernir características dos objetos (um exemplo simples de experimentação no segundo nível é o aluno perceber que um quadrado também é um retângulo por conter todas as propriedades exigidas deste último).

Ainda sobre a “prova de teoremas”, podemos acrescentar outra razão para o uso da GD: os contra-exemplos. Com a GD o estudante pode mais facilmente encontrar uma configuração que sirva de contra-exemplo a uma conjectura em estudo. Esta observação também pode servir como resposta a uma das críticas mais comuns contra a GD: a visualização dispensaria, ou desestimularia, a necessidade de prova matemática. Esta crítica pode ser encontrada, por exemplo, em [Munzer 1996].

Em resumo, como a GD possibilita visualizar uma mesma construção de diversas formas, e dessa maneira, facilita a compreensão do comportamento geométrico dos elementos envolvidos [Rodrigues 2002], podemos utilizar um programa de GD para revelar relações geométricas intrínsecas que poderiam passar despercebidas numa representação estática. Com isso, podemos incentivar o espírito investigativo do estudante, solicitando ao final uma justificativa para as relações encontradas (a prova matemática).

## 2.1. Geometria Dinâmica no ensino: Geometria Euclidiana e Trigonometria

Como observado inicialmente, a Geometria Dinâmica pode ser utilizada no estudo da geometria de “régua e compasso”, Geometria Euclidiana, e devido a seus recursos adicionais, a GD também pode ser empregada em modalidades de geometria que envolvam medida, como por exemplo na Trigonometria.

O nome **Geometria Euclidiana** é devido ao matemático grego Euclides de Alexandria, que viveu entre os anos de 330 a 277 a.C, aproximadamente. Em sua obra intitulada *Elementos*, composta de postulados, axiomas e teoremas, Euclides procurou sistematizar o saber geométrico da época, baseando-se em trabalhos de outros matemáticos famosos, como Hipócrates de Chios, Platão e Pitágoras [Boyer 1974].

A origem e o desenvolvimento da **trigonometria** estão bastante ligados à astronomia. Os astrônomos babilônicos dos séculos IV e V a.C. obtiveram várias informações que foram transmitidas aos gregos, e Hiparco de Nicéia, na segunda metade do século II a.C escreveu um tratado de doze livros relativo ao que hoje

conhecemos como trigonometria (neste trabalho existe uma extensão para trigonometria esférica).

A Geometria Dinâmica pode ser empregada também em outros tópicos de Geometria, além da geometria construída com régua e compasso. Um exemplo interessante é a “descoberta” das funções trigonométricas: utilizando recursos de medidas (distâncias entre objeto) de GD e fazendo experimentações, o aprendiz pode perceber as relações métricas num triângulo retângulo. Este exemplo é ilustrado figura 1.

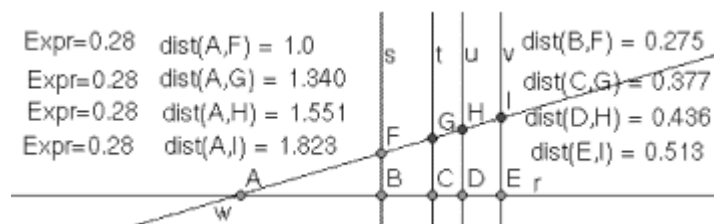


Figura 1. Semelhanças entre triângulos retângulos, razão entre comprimentos dos catetos “verticais” com as hipotenusas no iGeom.

## 2.2. Introduzindo conceitos de algoritmo via programação geométrica

Informalmente podemos dizer que um **algoritmo** é uma seqüência finita de passos, que aplicada a um conjunto de **dados de entrada** produz um conjunto de **dados de saída** (ou resposta). Além disso, a menos de uma classe particular de algoritmos, um algoritmo deve ser **determinístico**, ou seja, sempre que for aplicado sobre um mesmo conjunto de entradas, deve produzir o mesmo conjunto resposta. Observe que, se o conjunto de entrada for vazio, a saída do algoritmo será sempre a mesma.

Uma característica importante de um algoritmo é que ele resolve uma classe de problemas e não uma instância. Por exemplo, um algoritmo de ordenação para  $N$  números inteiros ( $N < 10^8$ ), ordena qualquer conjunto finito (com no máximo  $10^8$  elementos), em qualquer configuração (i.e., qualquer que seja a permutação dentre as  $N!$  possíveis). A aplicação do algoritmo sobre um particular conjunto de inteiros, constitui a resolução de uma instância do problema. O limitante do tipo “ $N < 10^8$ ” é imposto pelo computador onde é implementado o algoritmo de ordenação, do ponto de vista matemático o algoritmo funciona para qualquer  $N$  natural.

Esta observação permite entendermos melhor a diferença entre a GD, que é do tipo 1- $N$  e a geometria estática, tipo 1-1: uma solução geométrica implementada em GD na prática constitui um algoritmo, enquanto a correspondente solução estática equivale a uma aplicação do algoritmo geométrico sobre um conjunto fixado de dados (estáticos e, portanto, únicos).

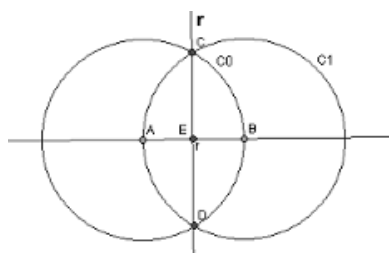
A caracterização de soluções geométricas como algoritmos, pode ser melhor percebida por um aprendiz quando trabalhando com a GD, pois ao finalizar uma construção e testá-la com outras configurações (de entrada), fica claro que a mesma pode ser aplicada a qualquer outro conjunto de entradas (dentre os possíveis). Por outro lado, como observa [Rodrigues 2002], as soluções obtidas pela geometria da régua e compasso são estáticas e particulares, pois não podem ser alteradas e nenhuma delas garante o significado genérico de sua definição (“o desenho de um círculo, por exemplo,

possui um centro e raio, ambos fixos, mas o conceito de círculo não depende de valores arbitrários”).

Um exemplo simples deste processo, usado por [Brandão 2002] para introduzir o conceito de **algoritmo geométrico**, é a obtenção da mediatriz de dois pontos dados,  $A$  e  $B$ : uma vez efetuada a construção (solução), o aprendiz pode mover  $A$  ou  $B$  (entradas do problema), ficando óbvio que sua solução pode ser aplicada a quaisquer pares de pontos!

<pre> Mediatriz(A,B):   C0 := Circ(A,B);   C1 := Circ(B,A);   ln := Inters(C0,C1,n);   ls := inters(C0,C1,s);   r := Reta(ln.ls);   Resposta r </pre>
---

**Tabela 1. Exemplo de Construção de uma Mediatriz.**



**Figura 2b. Visualização da Mediatriz utilizando o iGeom.**

Na tabela 1 acima, ilustramos uma seqüência de passos que determina a mediatriz de dois pontos quaisquer, rotulados  $A$  e  $B$ . Os comandos *Circ*, *Inters* e *Reta* podem ser entendidos como primitivas (funções) geométricas: *Circ*( $X, Y$ ) é a circunferência centrada em  $X$  e passando pelo ponto  $Y$ ; *Inters*( $X, Y, p$ ) é um dos pontos de interseção entre os objetos  $X$  e  $Y$  (podem existir dois, neste caso definidos por "norte", se  $p=n$ , ou "sul", se  $p=s$ ); e *Reta*( $X, Y$ ) é a reta que contém os pontos  $X$  e  $Y$  (se  $X=Y$ , a reta não será única).

Deste modo é natural esperar que programas de GD permitam que construções geométricas sejam armazenadas explicitamente na forma de **funções**, como o fazem o iGeom, GSP e Cabri. Estas funções geométricas, por agruparem seqüências de comandos, recebem o nome de **scripts** (como nas versões atuais do iGeom e do GSP) ou **macros** (como no Cabri I e II). Uma vez armazenada uma função, o usuário/aprendiz pode: marcar os objetos de entrada (na ordem correta) e depois selecionar a função desejada (como no iGeom e GSP) ou selecionar a macro e depois marcar os objetos de entrada (Cabri). A primeira forma pode ser dita tipo "**seleção+ação**", enquanto a segunda "**ação+seleção**", como descrito em [Braviano e Rodrigues 2002].

A possibilidade de armazenar algoritmos geométricos como funções é bastante útil do ponto de vista didático, pois permite que o aprendiz armazene em funções as construções que utiliza mais freqüentemente e com isso possa concentrar sua energia nas tarefas novas. Os programas citados permitem até que uma função, já armazenada, seja invocada durante a "geração" de uma nova função.

Exemplos nos quais o uso de funções geométricas são muito úteis são aqueles com processo de repetição, nos quais é necessário aplicar várias vezes a mesma seqüência de passos, como no exemplo abaixo, apresentado em [Brandão 2002].

**Exemplo 1:** *Aquiles e a Tartaruga (ou "paradoxo de Zenão")*

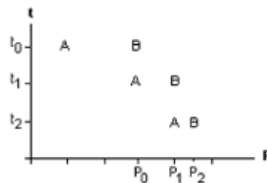
*Aquiles e uma tartaruga apostam uma corrida, sendo que Aquiles tem o dobro da velocidade da tartaruga, e por isso, a tartaruga larga à frente.*

Para simular geometricamente este modelo podemos definir uma função (tabela 2) e a empregar seguidamente.

**Tabela 2: construção de uma função para o exemplo 1.**

<b>Dist_metade</b>	
Entrada: ponto A, ponto B	
Saída: ponto F tal que F pertence semi-reta(A,B) e $d(B,F)=d(A,B)/2$ .	
1. r :=	semi_reta(A,B) semi-reta começando em A, passando por B
3. C0:=	circ(B,A) circunferência centrada em B, passando por A
2. C1:=	circ(A,B) circunferência centrada em A, passando por B
4. C :=	inters(C0,C1,n) C é interseção "superior" (norte) entre as circunferências C0 e C1
5. D :=	inters(C0,C1,s) D é interseção "inferior" (sul) entre as circunferências C0 e C1
6. S1 :=	segm(C,D) S1 é o segmento ligando C à D
7. E :=	inters(r,S1) E é interseção entre r e S1
8. C2 :=	circ(B,E) circunferência centrada em B, passando por E
9. F :=	inters(C2,r,n) F é interseção "superior" (norte) entre C2 e r

Sendo A a posição inicial de Aquiles e B a posição inicial da tartaruga, definindo  $t_0=0$  o instante inicial e  $P_0=B$  a posição inicial da tartaruga, denotaremos por  $t_{k+1}$  o instante em que Aquiles alcança a posição  $P_k$  ocupada pela tartaruga no instante  $t_k$ .(figura 3).



**Figura 3. Posições de Aquiles e da tartaruga no instante  $t_k$**

Neste caso a simulação é obtida aplicando-se  $dist\_metade(A,B)$ , para definir o ponto  $P_1$ , depois aplica-se a função sobre o par  $(B,P_1)$  para obter  $P_2$  e assim por diante.

**Tabela 3. Versão recorrente, com contador, para a simulação do exemplo 1.**

```

aquiles_tartaruga(A, B, cont) {
  se (cont=FIM) retorne // termine laço
  Aux := dist_metade(A, B);
  repete_dist_metade(B, Aux, cont+1); // recorrência
}

```

Note que o algoritmo geométrico acima apresenta um **bloco de repetição** ou **laço**, comum na programação usual (em linguagens como C, Pascal ou Java), do tipo “para  $i=1$  até 5 { Aux := dist\_metade(A,B); A:=B; B:=Aux;”, como mostra a tabela 3.

Os programas de GD não implementam este tipo de linguagem, principalmente no que concerne a comandos do tipo “se *CONDIÇÃO* então *COMANDO*”. Entretanto o iGeom e o GSP (e só eles - até onde sabemos) permitem que na definição de uma função existam chamadas recorrentes (como na função fatorial  $f(n)=n!$ :  $f(n) = \{ 1, se n=0; n * f(n-1), em caso contrário \}$ ) e com isso podemos armazenar uma função com o mesmo significado (semântica) do algoritmo 1(tabela 4).

**Tabela 4. Versão recursiva da função Aquiles\_tartaruga**

<b>aquiles_tartaruga(A,B)</b> Entrada: ponto A, ponto B 1. $F := \text{dist\_metade}(A,B)$ //repete os passo 1 a 9 da construção 1 2. <b>aquiles_tartaruga(B,F)</b> //recorrência aplicada aos pontos B e F
--

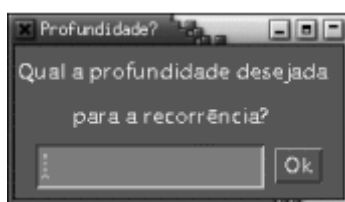
O número de vezes que a recorrência será aplicada é controlado na chamada da função, o usuário especifica este número como na figura 3. A seguir apresentamos mais um exemplo que usamos em [Brandão 2002], que ilustra o uso do iGeom para explorações de fractais geométricos.

**Exemplo 2:** *Fractal de circunferências definidas por polos e pontos médios*

*Dados dois pontos A e B, construir a circunferência C0 com centro A e passando por B, os quatro "polos" de C0 e seus pontos médios com o centro A.*

**Tabela 5. construção de uma função para o exemplo 2.**

<b>frac_circ(A,B,C0)</b> Entradas: ponto A, ponto B Saídas: circunferências centradas em polos de outra, com metade do raio 1. $C0 := \text{circ}(A,B)$ circunferência centrada em A, passando por B 2. $r := \text{reta}(A,B)$ reta passando por A e por B 3. $s := \text{reta\_perp}(A,r)$ reta perpendicular a r passando por A 4. $(C,D) := \text{int}(C0,r)$ C e D interseções entre C0 e r 5. $(E,F) := \text{int}(C0,s)$ E e F interseção entre C0 e s 6. $M1 := \text{pto\_med}(A,C)$ M1 ponto médio do segmento AC (implementado em macro) 7. $M2 := \text{pto\_med}(A,F)$ M2 ponto médio do segmento AF 8. $M3 := \text{pto\_med}(A,D)$ M3 ponto médio do segmento AD 9. $M4 := \text{pto\_med}(A,E)$ M4 ponto médio do segmento AE 10. <b>frac_circ(B,M1)</b> recorrência aplicada aos ponto B e M1 11. <b>frac_circ(F,M2)</b> recorrência aplicada aos ponto F e M2 12. <b>frac_circ(D,M3)</b> recorrência aplicada aos ponto D e M3 13. <b>frac_circ(E,M4)</b> recorrência aplicada aos ponto E e M4 14. <b>esconde(r,s,M1,M2,M3,M4)</b> esconde os objetos de rótulo listado (para efeito visual)
---



**Figura 4. Janela para que usuário defina profundidade para funções recorrentes no iGeom.**

A janela da figura 4 é aberta sempre que uma função com recorrência é invocada no iGeom. Na figura 5 mostramos também no *iGeom*, o resultado de três aplicações da função recorrente *frac\_circ*, na primeira foi digitado como profundidade o valor 0, na segunda o valor 1 e na última 2.



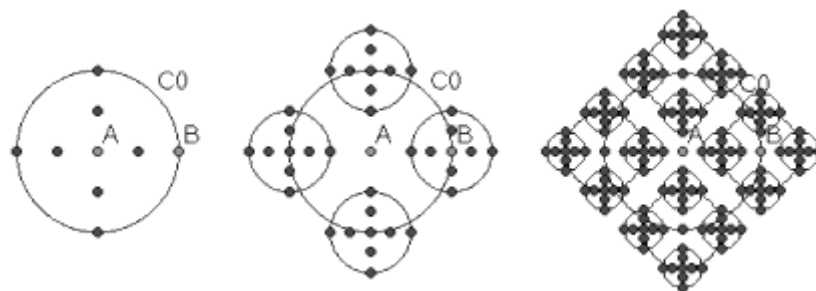


Figura 5. Exemplo de recursão no iGeom.

### 3. iGeom: Uma Plataforma de Ensino Geométrico e Matemático

O objetivo geral da plataforma *iGeom: Geometria Interativa na Internet*, é proporcionar recursos para facilitar o ensino e aprendizagem de Geometria, por um lado providenciando recursos que auxiliem o professor na produção de material didático e no acompanhamento de seus alunos, e por outro, trazendo facilidades para um aprendiz adquirir conhecimentos geométricos (seja na escola ou fora dela, de modo auto-didático).

Um objetivo específico ao iniciarmos o projeto *iGeom* foi “democratizar” o uso da Geometria Dinâmica, permitindo que qualquer estudante ou professor com acesso a um micro-computador pudesse usufruir dos benefícios da Geometria Dinâmica. O iGeom é distribuído gratuitamente pela Internet, podendo ser testado e descarregado a partir do sítio **iMática**, <http://www.matematica.br>, ou mais precisamente da página <http://www.matematica.br/igeom>.

Para atingir estes objetivos, a plataforma iGeom, que está em constante desenvolvimento, é implementada na linguagem de programação Java [Java 2001], por esta permitir grande portabilidade (possibilidade de uso em diferentes computadores/sistemas operacionais) e possibilitar seu uso diretamente em páginas Internet, na forma de “applet” (programa Java especialmente projetado para a Web).

A plataforma iGeom foi iniciada no segundo semestre de 2000, a partir de projetos de iniciações científicas, com Ricardo Hideo Sehara (segundo semestre de 2000 até início de 2002), com Fabiana Piesigilli (segundo semestre de 2001 até segundo semestre de 2002) e Renata Teixeira Carneiro (ano de 2002), todos sob orientação do primeiro autor.

Atualmente o *iGeom* está sendo utilizado como principal ferramenta em uma disciplina obrigatória para alunos do curso de licenciatura no Instituto de Matemática e Estatística (IME) da Universidade de São Paulo e também nos cursos do **LEM** (Laboratório de Ensino de Matemática: <http://www.ime.usp.br/lem>).

Devido ao início do uso “prático” do iGeom, em janeiro de 2003, o primeiro autor tem introduzido novas opções ou feito correções com grande frequência (no mínimo uma a cada semana). A versão atual permite realizar todas as operações básicas de GD, como por exemplo: criar objetos geométricos (como pontos, retas, semi-retas, segmentos e circunferências); opções de edição (esconder/mostrar, remover ou desfazer remoção, rastrear objetos,...); e opções de gravação/recuperação de arquivos (em três possíveis formatos - explicados na próxima seção). Um recurso que implementamos no

*iGeom* e que é raro nos outros programas de GD é a possibilidade de fazer macro/script com recorrência, como abordado na seção anterior.

### 3.1. *iGeom*: algumas características e comparações

A interface do *iGeom* foi desenvolvida com o intuito que o mesmo pudesse ser utilizado nas formas aplicativa e “applet”. A menos da opção de gravação, que não é permitida em “applet” (portanto via Internet) por razões de segurança, todas as demais estão acessíveis a partir dos botões.

Além disso, estes botões foram construídos, assim como o restante do programa, usando uma quantidade pequena dos recursos hoje disponíveis na atual versão do Java, de modo a ampliar as possibilidades de portabilidade. Explicando de modo mais “técnico”, o *iGeom* está escrito em “Java puro”, usando apenas a classe gráfica **AWT** e deste modo pode ser interpretado por qualquer navegador que tenha embutido o **Java 1.1** (atualmente o Java está na versão 1.4). Apenas para ilustrar a desvantagem em se usar as novas versões desta linguagem, se a usássemos os recursos da “biblioteca” **Swing**, poderíamos muito mais facilmente implementar botões e menus, entretanto um usuário com navegador Netscape 4.77 não conseguiria rodar o *iGeom* (precisaria “atualizar” o navegador, eufemismo para “troque de navegador”!).

Em geral, programas de Geometria Dinâmica dispõem de recursos que permitem trabalhos além das possibilidades da Geometria Euclidiana, principalmente por poderem incorporar movimentos e medições (distâncias e áreas). Por exemplo, no *iGeom* e no *GSP* é possível construir uma reta perpendicular a um dos eixos cartesianos cuja distância até a origem é dada por uma medida qualquer (isso permite ao aprendiz construir gráficos trigonométricos, movendo um ponto sobre o traço de uma circunferência). Já *Cabri*, *Cinderella* e *Tabullae* permitem, por exemplo, construir uma cônica definida por cinco pontos.

As medidas também são implementadas de modo “dinâmico”, por exemplo, se o usuário colocar dois pontos soltos e medir a distância entre ambos, ao mover um dos pontos o *iGeom* atualiza automaticamente (de modo aparentemente contínuo) o valor da medida representada na tela.

Como citado anteriormente, no *iGeom* existem três tipos de armazenagem de construções: guardar o desenho para a versão aplicativa (GEO); guardá-lo na forma de “applet” (exportar para HTML+Java); ou guardar o algoritmo que realiza a construção, na forma de *script* (SCR). A primeira é utilizada na área de desenho do *iGeom*, a segunda é para ser incorporada em páginas HTML (é a mesma do GEO, com cabeçalhos para “applet”) e a última é uma extensão do formato GEO para permite execuções de funções (*scripts*).

Muitas dessas ferramentas de auxílio no estudo da Matemática e da Geometria apresentadas nas seções anteriores são comuns em programas de Geometria Dinâmica como *Cinderella*, *GSP*, *Cabri* e outros.

### 3.2. *iGeom*: Um software de auxílio à educação a distância

Nos dias atuais, com a globalização e a grande quantidade de informação disponível na Internet, boa parte do esforço exercido pelos educadores, interessados em usar a Internet, está direcionado para a criação de conteúdo de boa qualidade e adequados para

a aprendizagem do aluno. Deste modo, ferramentas de autoria para desenvolvimento e publicação de cursos à distância são necessárias para facilitar esta tarefa, e por isso, parte de nosso trabalho com o *iGeom* atualmente é para beneficiar estes educadores.

Um dos desafios que pretendemos vencer é a incorporação de recursos de comunicação no *iGeom* para permitir, por exemplo, que o professor obtenha informações dos exercícios que cada aluno já resolveu. Este exemplo está relacionado com um ambiente presencial de ensino, mas também podemos pensar neste processo para comunicação assíncrona, onde o aprendiz pode entrar no sistema a qualquer momento para estudar. Este é um objetivo ambicioso, pois para isso necessitaremos desenvolver mecanismos confiáveis para avaliação de respostas ao problema proposto pelo professor, eventualmente, analisando o “conhecimento” do aluno relativo ao assunto e sugerindo alternativas/dicas para ele vencer dificuldades encontradas na tentativa de resolução do exercício.

De modo geral, os recursos de autoria no *iGeom* deverão permitir que os professores possam criar e adaptar o material didático estático que normalmente é usado no ensino de geometria, trazendo benefícios ao aprendizado do aluno e permitindo uma abordagem mais construtiva com exemplos interativos que ilustram o relacionamento entre os objetos geométricos.

#### **4. Conclusões**

A Geometria é, sob nosso ponto de vista, a área da Matemática que mais se beneficiou com o uso do Computador e isso se deve à Geometria Dinâmica (GD). A razão chave desta nossa crença pode ser ilustrada por um antigo ditado atribuído a Confúcio: *o aluno ouve e esquece, vê e se lembra, mas só compreende quando faz*. Ou seja, para aprender é necessário *fazer* e a GD auxilia o *fazer*, permitindo que o aluno experimente e descubra por si só, relações. E como observado por [Fleming at al 2000], para que a “auto-aprendizagem” tenha sucesso é preciso que o aprendiz tenha acesso a materiais suficientemente compreensíveis e atrativos.

O *iGeom* é uma tentativa de popularizar mais a GD no Brasil, para isso adotamos o Java (e seus “applets”) como linguagem de desenvolvimento, para que pudéssemos disponibilizar um ambiente aberto de ensino/aprendizagem na Web. Mais que um sistema de Geometria, a GD (e o *iGeom*) é um rico ambiente que proporciona vários aprendizados matemáticos, como o conceito de algoritmos.

Entretanto o desenvolvimento do *iGeom* continua, pois esperamos transformá-lo em um sistema de produção de material didático (autoria), bem como de acompanhamento de aprendizado.

#### **Referências**

- Boyer, C. B. (1974) “História da Matemática”, Ed. Edgard Blucher.
- Brandão, L. O. (2002) "Algoritmos e Fractais com programas de Geometria Dinâmica", Revista do Professor de Matemática 49, p. 27-34.
- Braviano, G. e Rodrigues, M. H. W. L. (2002) "Geometria Dinâmica: uma nova Geometria?", Revista do Professor de Matemática 49, p. 22-26.

- Flemming, D. M., Luz, E. F. e Coelho, C. (2000) "Desenvolvimento de Material Didático para Educação a Distância no Contexto da Educação Matemática", Anais eletrônicos do VII Seminário Internacional de Educação a Distância.
- Gravina, M. A. (1996) "Geometria Dinâmica - Uma Nova Abordagem para o Aprendizado da Geometria", Anais do VII Simpósio Brasileiro de Informática na Educação, p. 1-13.
- Isotani, S. e Brandão, L. O. (2001) "iMática: ambiente interativo de apoio ao ensino de matemática via internet", anais do Workshop sobre Informática na Escola, XXI Congresso da Sociedade Brasileira de Computação, p. 533-543.
- Java (2001) - Linguagem de Programação Java. URL: <http://www.java.sun.com>.
- King, J. e Shattschneider, D. (1997) "Geometry Turned On - Dynamic Software in Learning, Teaching and Research", Washington : Mathematical Association of America.
- Lindquist, M. M. e A. P. Shulte editores (1998) "Aprendendo e ensinando Geometria", Ed. Atual (Crowley, M.L., "O modelo van Hiele de desenvolvimento do pensamento geométrico").
- Melo, L. B., Ferreira, J. M. e Pontes, J. D. A. (2000) "Um software Educacional para o descobrimento de Propriedades Matemáticas", Anais do XX Congresso da Sociedade Brasileira de Computação.
- Munzner T. (1996) "Mathematical visualization: standing at the crossroads", In: Proceedings of the conference on Visualization, p. 451-453.
- Papert, S. (1999) "Mindstorms: children, computer and powerful ideas", Second Edition New York: Basic Books.
- Rizzato, F. B. e Milies, F. C. P. (2001) "O início da Trigonometria", <http://www.matematica.br/historia/trigonometria.html>, Dezembro.
- Rodrigues, D. W. L. (2002) "Uma Avaliação Comparativa de Interfaces Homem-Computador em Programas de Geometria Dinâmica", Tese (mestrado) em Engenharia de Produção da Universidade Federal de Santa Catarina.
- Santos, E. T. (2000) "Novas Tecnologias no Ensino de Desenho e Geometria", Anais do Encontro Regional do Vale do Paraíba de Profissionais do Ensino de Área de Expressão Gráfica, p. 71-81.
- Valente, J. A. (org.) (2001) "O Computador na sociedade do conhecimento", URL: [http://www.seed.pr.gov.br/doc\\_mec/livro02.pdf](http://www.seed.pr.gov.br/doc_mec/livro02.pdf), descarregado em 28/05/2001.