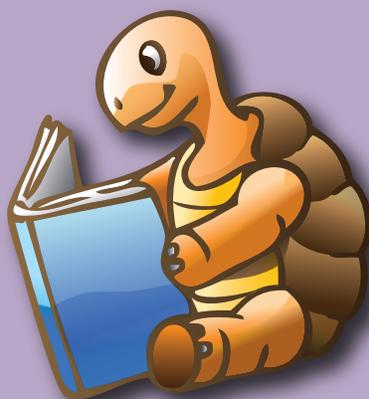


# Softwares Livres Eduacionais



**Dr Geo**  
Versão 1.1  
Geometria Interativa  
Volume 1

# Softwares Livres Educativos



## Dr. Geo

Versão 1.1

Geometria Interativa

Volume 1





SECRETARIA DE ESTADO DA EDUCAÇÃO  
SUPERINTENDÊNCIA DA EDUCAÇÃO  
DIRETORIA DE TECNOLOGIA EDUCACIONAL  
COORDENAÇÃO DE MULTIMEIOS

**Dr. Geo**  
**Versão 1.1**  
**Geometria Interativa**  
**Volume 1**

CURITIBA  
SEED-PR  
2010

É permitida a reprodução total ou parcial desta obra, desde que seja citada a fonte. Disponível também em: <<http://www.diaadia.pr.gov.br/multimeios>>

**Coordenação de Multimeios**

**Tradução e adaptação do Espanhol para o Português**

Elciana Goedert

**Coordenação de Mídia Impressa e Web**

**Revisão ortográfica**

Aquias da Silva Valasco

Bárbara Reis Chaves Alvim

Orly Marion Webber Milani

Tatiane Valéria Rogério de Carvalho

**Coordenação de Multimeios**

**Produção**

Eziquiel Menta

Ricardo Mendonça Petracca

**Capa**

Jocelin José Vianna da Silva

Rafael Cadilhe David

**Projeto Gráfico**

Juliana Gomes de Souza Dias

**Diagramação**

Hellen Falco Yokowo

Taisa Delazzeri Burtet

**CATALOGAÇÃO NA FONTE - CEDITEC-SEED-PR**

Paraná. Secretaria de Estado da Educação. Superintendência da Educação. Diretoria de Tecnologias Educacionais. P111

Dr. Geo , versão 1.1 : geometria interativa ; v. 1, tradução e adaptação / Secretaria de Estado da Educação. Superintendência da Educação. Diretoria de Tecnologias Educacionais. – Curitiba : SEED – Pr., 2010. – V. 1 ; 60 p – ISBN978-85-8015-009-4

1. Dr. Geo. 2. Geometria interativa. 3. Script sheme. 4. Educação-Paraná.

I. Goedert, Elciana, trad. II. Título.

CDD600

CDU 6+37(816.2)

**SECRETARIA DE ESTADO DA EDUCAÇÃO**

Superintendência da Educação

Diretoria de Tecnologia Educacional

Coordenação de Multimeios

Rua Salvador Ferrante, 1.651 – Boqueirão

CEP 81670-390 – Curitiba – Paraná

[www.diaadia.pr.gov.br/multimeios](http://www.diaadia.pr.gov.br/multimeios)

IMPRESSO NO BRASIL  
DISTRIBUIÇÃO GRATUITA

## APRESENTAÇÃO

A Diretoria de Tecnologia Educacional (Ditec), da Secretaria de Estado da Educação do Paraná, viabiliza ações que possibilitam não apenas o acesso operacional aos equipamentos existentes na escola, mas também a pesquisa, a produção e a veiculação de conteúdos educacionais de forma compatível com os avanços tecnológicos.

Nossa proposta consiste na integração e articulação das mídias com o mundo moderno por meio de ações desenvolvidas pela TV Paulo Freire, pelo Multimeios e pelo Portal Dia-a-dia Educação. Além do acesso à tecnologia, entendemos que é necessária a orientação para o seu uso. Essa é a tarefa da equipe dos assessores da Coordenação Regional de Tecnologia na Educação (CRTE) presente nos Núcleos Regionais de Educação e que atendem aos professores das escolas da Rede Pública de Ensino do Estado do Paraná.

Para complementar a ação de produção e disponibilização de materiais didáticos de apoio ao uso de tecnologia, o Multimeios elaborou tutoriais de alguns dos *softwares* educacionais instalados no Paraná Digital para serem disponibilizados aos usuários dos laboratórios. Entre eles encontra-se o tutorial do Dr. Geo, um *software* de construção em Geometria que pode ser utilizado por professores do Ensino Fundamental e Médio, em disciplinas como Matemática e Física. Ele permite ao aluno explorar, de maneira interativa, noções como óptica, cinemática e relações trigonométricas além de permitir fazer cálculo de ângulos, interseção e equações de retas. Consideramos que o uso deste aplicativo pode contribuir com o processo de aprendizagem.

O conteúdo deste tutorial foi traduzido e adaptado do sítio oficial do *software* Dr. Geo e autorizado para uso da Secretaria de Estado da Educação do Paraná de acordo com a realidade da Rede Pública Estadual de Ensino.

Elizabete dos Santos  
**Diretora de Tecnologia Educacional**

Eziquiel Menta  
**Coordenador de Multimeios**

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>8</b>
1.1 PRELIMINARES .....	8
1.2 DR. GEO NA INTERNET .....	10
<b>2 FUNÇÕES BÁSICAS .....</b>	<b>10</b>
2.1 FERRAMENTAS DE CONSTRUÇÃO .....	10
2.1.1 Ferramentas de pontos .....	11
2.1.2 Ferramentas de linhas .....	12
2.1.3 Ferramentas de transformação .....	13
2.1.4 Ferramentas numéricas .....	15
2.1.5 Ferramentas para macros de construções .....	16
2.2 OUTRAS FERRAMENTAS .....	16
2.2.1. Árvore lógica de construção .....	16
2.2.2 Mover a figura .....	17
2.2.3 Mover um objeto .....	17
2.2.4 Apagar um objeto .....	17
2.2.5 Editar a aparência de um objeto .....	17
2.2.6 Editar propriedades de objetos .....	19
2.2.7 Ativar uma grade .....	20
2.3 PREFERÊNCIAS DE USUÁRIO .....	20
2.3.1 Comportamento predeterminado .....	20
2.3.2 Outras preferências .....	21
<b>3 CARACTERÍSTICAS AVANÇADAS .....</b>	<b>21</b>
3.1 MACROS DE CONSTRUÇÕES .....	22
3.1.1 Criar uma macro .....	24
3.1.2 Executar uma macro .....	26
3.1.2.1 Executar macro através da janela de diálogo .....	26
3.1.2.2 Executar macro com a ajuda do menu Macros .....	27
3.2 <i>SCRIPT SCHEME</i> DO DR. GEO .....	27
3.2.1 <b>DGS</b> por meio de exemplos .....	28
3.2.1.1 <b>DGS</b> sem parâmetros de entrada .....	29
3.2.1.2 Cálculo de algumas constantes comuns .....	30
3.2.1.3 <b>DGS</b> com ao menos um parâmetro de entrada .....	30
3.2.2 Métodos de referência para os <i>scripts</i> do Dr. Geo .....	34
3.3 <i>FIGURAS SCHEME</i> DO DR. GEO .....	37

3.3.1 Alguns exemplos .....	38
3.3.2 Métodos de referência para as figuras <i>Scheme</i> do Dr. Geo.....	41
3.3.2.1 Comandos gerais.....	41
3.3.2.2 Definição de objetos em uma figura .....	41
3.3.2.3 As transformações geométricas .....	46
3.3.2.4 Modificação dos atributos (aparências) dos objetos ..	51
3.3.3 Sinônimos dos comandos das figuras <i>Scheme</i> do Dr. Geo.....	52
3.3.4 Galeria de exemplos.....	57
<b>REFERÊNCIAS .....</b>	<b>60</b>

**No volume 2 estão disponíveis os seguintes conteúdos:**

- **Arquivos e documentos**
- **Aplicações didáticas**
- **Livro de receitas**

# 1 INTRODUÇÃO

## 1.1 PRELIMINARES

O Dr. Geo é um programa tanto de geometria interativa como de programação em linguagem *Scheme* que permite criar figuras geométricas, assim como manipulá-las interativamente, respeitando suas restrições geométricas. Ele oferece igualmente a possibilidade de o aluno ser introduzido gradualmente na programação, sendo útil para o ensino de estudantes de nível básico ou superior.

A interface de usuário do Dr. Geo foi concebida para dar ergonomia e recursos avançados, dentro de um conjunto ao mesmo tempo harmônico e simples na sua operação. Assim, ela permite ao usuário iniciante a familiarização com as funções básicas do programa. Posteriormente, à medida que ele esteja mais ambientado, conhecerá os aspectos mais avançados da interface e do funcionamento do Dr. Geo: múltiplos modos para a construção de objetos,<sup>1</sup> macros de construções, seções, adaptabilidade da interface, uso de *script* e de figuras *Scheme* do Dr. Geo. Ou seja, Dr. Geo é programado de dois modos distintos. Essas funções avançadas geram pouquíssima sobrecarga à interface, e por isso ele é utilizado com entusiasmo no ensino básico, o que não invalida sua utilização a partir do nível médio.

Inicialmente, serão apresentadas as ferramentas básicas e, posteriormente, as funções avançadas.

Na interface há a barra de menu, com **Arquivo, Editar, Macros, Janelas, Ajuda**, e a de ações, utilizada para criar uma nova figura ou um texto explicativo. Contém também as ferramentas de **Fazer/desfazer** e a grade.

Para criar uma figura geométrica nova, clique no primeiro botão da barra de ferramentas. Além desse procedimento, uma nova figura pode ser criada, utilizando o elemento da

---

<sup>1</sup> Trata-se de poder, a partir de um mesmo comando, criar um tipo de objeto segundo se deseja. Por exemplo, a partir do comando para a construção de um círculo, é possível criar um círculo a partir de seu centro e dele um ponto, um comprimento, um segmento, e assim por diante. Contudo, esse comando é representado por um só ícone, e Dr. Geo se antecipa para oferecer ao usuário a construção desejada. O efeito imediato é que existe uma redução do tempo de aprendizagem da interface, ao propor um número importante de modos de construção.

barra de menu **Arquivo / Novo / Figura**. Quando uma nova figura é criada, aparece uma barra formada por seis ícones, a qual termina com um menu que permite ajustar a escala da figura geométrica atual.

Ademais, uma barra de ícones vertical, à esquerda da figura, oferece um rápido acesso às ferramentas mais utilizadas. Os seis ícones da barra são entradas de menu gerais chamadas de funções específicas, e serão descritas no próximo capítulo.



Figura 1–Tela de boas-vindas do Dr. Geo

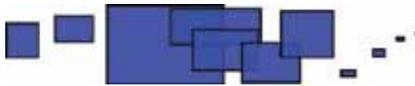


Figura 2 – Figura geométrica vazia

Para cada figura, uma árvore panorâmica (ou árvore de construção) está disponível. Inicialmente, este painel está oculto no extremo esquerdo, de modo que só a representação da figura é visível. Em qualquer momento, é possível empurrar o painel para a direita, como segue: mova o *mouse* para o lado esquerdo da figura; quando o ponteiro do *mouse* mudar de forma, pressione o botão e, mantendo-o pressionado, mova o *mouse* para a direita. Com ele, será

visível a descrição da figura, formada por uma árvore com todos os elementos dessa figura.

Os elementos que dependem de outros podem ser revelados ao clicar sobre o símbolo +. Isso mostrará os elementos a que estes deram origem (conduzem a uma linha que pode ser dois pontos, embora possa ter outras origens. Também podem ser chamados elementos antecedentes ou elementos-pai).



Figura 3 – Figura do Dr. Geo e sua descrição

## 1.2 DR. GEO NA INTERNET

O Dr. Geo dispõe de seu próprio espaço na Internet, no sítio **Ofset.org** (<http://www.ofset.org/drgeo>).

Nesse espaço, você encontrará:

- informações para obter o Dr. Geo;
- a documentação do programa;
- indicações para envolvimento no projeto Dr. Geo;
- referências para o uso pedagógico do programa.

## 2 FUNÇÕES BÁSICAS

Este capítulo descreve as ferramentas utilizadas para construir figuras geométricas. Ao final do capítulo, também é apresentada a configuração das preferências do usuário.

### 2.1 FERRAMENTAS DE CONSTRUÇÃO

Essas ferramentas estão separadas em seis grupos disponíveis a partir da segunda barra de ferramentas do Dr. Geo.



Figura 4 –  
Categorias de  
ferramentas do  
Dr. Geo e suas  
descrições

Ao clicar sobre um dos ícones decorados com um pequeno triângulo verde, uma nova barra vertical (de ícones) aparece imediatamente. Nela se agrupam funções de uma mesma família.

Da esquerda para a direita, temos acesso às barras de ferramentas verticais para: construir pontos, construir linhas, realizar transformações geométricas, calcular valores numéricos, gerar macros de construções e utilizar as ferramentas de edição.

**Outras ferramentas** – estas funções também aparecem dentro do menu contextual de uma figura, que surge ao se clicar com o botão direito no fundo da figura.

### 2.1.1 Ferramentas de pontos

**Ponto livre** – cria um ponto livre na área ou um ponto livre em um objeto unidimensional (segmento, semirreta, reta, arco de círculo, círculo, lugar geométrico). No primeiro caso, o ponto criado pode mover-se para qualquer lugar do fundo da figura. Para construí-lo, clique em qualquer parte do fundo. No segundo caso, o ponto está limitado a mover-se no objeto unidimensional (linha ou círculo), ou seja, está “colado” ao objeto. Para construir esse tipo de ponto, clique sobre uma linha, que pode ser uma reta, uma semirreta, um segmento, um círculo, um arco de círculo, etc.

Para localizar um ponto a partir de suas coordenadas, a solução mais simples consiste em colocar um ponto livre e, em seguida, editar suas propriedades (veja a seção 2.2.6, “Editar propriedades de um objeto”, na página 19), ajustando livremente suas coordenadas.

Outra possibilidade, menos flexível para esse tipo de figura, consiste em colocar dois valores livres na figura (veja “Ferramentas numéricas”, na seção 2.1.4, página 15) e, em

seguida, construir um ponto com coordenadas dadas por esses dois valores (veja ferramenta **ponto definido por suas coordenadas**, nesta página). Esse procedimento tem uma vantagem sobre o anterior: o ponto construído não pode ser movido com o *mouse*, uma vez que está inexoravelmente atado à sua posição.

 **Ponto médio** – cria o ponto médio de um par de pontos ou a metade de um segmento. No primeiro caso, selecionamos dois pontos, no segundo, um segmento.

 **Interseção** – cria o(s) ponto(s) de interseção de duas linhas (objetos unidimensionais, ou seja, reta, semirreta, segmento, arco do círculo, círculo ou lugar geométrico). É necessário selecionar duas linhas (arco do círculo e semirreta).

 **Ponto definido por suas coordenadas** – cria um ponto definido por suas coordenadas. É necessário selecionar dois números: primeiro, o da abcissa e, segundo, o da ordenada.

Como colocar um ponto com restrições e suas coordenadas? Essa função é muito utilizada quando desejamos, por exemplo, construir o lugar geométrico de um ponto. Essa construção supõe a existência de dois valores (veja “Ferramentas numéricas”, na seção 2.1.4, página 15), cujo ponto é construído após a seleção desses dois valores, que podem ser independentes ou interdependentes.

### 2.1.2 Ferramentas de linhas

 **Reta** – cria uma linha reta definida por seus dois pontos. É necessário selecionar dois pontos.

 **Semirreta** – cria uma semirreta (também chamada de “raio”) definida por dois pontos. É preciso selecionar dois pontos. O primeiro é a origem; o segundo pertence à semirreta.

 **Segmento** – cria um segmento definido por dois pontos.

 **Vetor** – cria um vetor definido por dois pontos. É preciso

selecionar dois pontos. O primeiro é a origem; o segundo é o extremo. Uma vez que o vetor é criado, este pode ser movido independentemente dos dois pontos que lhe deram origem. Isso funciona ou pode ser utilizado para vetores construídos por meio de uma transformação (ver a seção 2.1.3, sobre transformações, na página 14).

 **Círculo** – cria um círculo. É possível criar um círculo a partir de várias opções:

- o centro e um ponto do círculo;
- o centro e um número (o raio do círculo);
- o centro e um segmento cuja longitude é o raio do círculo.

 **Arco do círculo** – cria um arco do círculo definido por três pontos. O primeiro ponto selecionado é a origem do arco, o segundo é um ponto sobre o arco e o terceiro é seu extremo. Os três pontos selecionados farão parte do arco.

 **Lugar geométrico** – cria um lugar geométrico (*locus*, em latim) dados dois pontos. É preciso selecionar dois pontos. Um é um ponto livre sobre uma linha (linha ou círculo); o outro é um ponto que depende do primeiro (isto é, quando um se move, o segundo também tem que se mover).

 **Polígono** – cria um polígono definido por  $n$  pontos. Selecionamos  $n+1$  pontos delimitando o polígono. O primeiro e o último pontos selecionados devem ser os mesmos, indicando ao Dr. Geo que a seleção está terminada. O objeto polígono não é como um objeto unidimensional; não é possível colocar um ponto sobre ele ou calcular a intersecção entre um polígono e outro objeto unidimensional. Por outro lado, é possível aplicar a um polígono uma transformação geométrica (rotação, reflexão, homotetia, etc.).

### 2.1.3 Ferramentas de transformação

 **Linha paralela** – cria uma linha paralela a uma direção que passa através de um ponto. Selecionamos um ponto e uma direção (ou seja, uma linha reta, uma semirreta, um segmento ou um vetor).



**Linha perpendicular** – cria uma linha perpendicular a uma direção que passa através de um ponto. Seleccionamos um ponto e uma direção (uma linha reta, uma semirreta, um segmento ou um vetor).

**Nota:** “ortogonal” é um sinônimo de “perpendicular”.



**Simetria axial (reflexão)** – cria a imagem de um objeto através de uma simetria axial (reflexão em uma reta). Seleccionamos o objeto a ser transformado e o eixo de simetria (que deve ser uma reta). Quando quisermos construir a imagem de uma linha reta, a primeira linha reta selecionada pelo usuário será a reta a refletir.



**Simetria central** – cria a imagem de um objeto através de uma simetria central. Seleccionamos os objetos que serão transformados e o centro de simetria (um ponto). Quando quisermos construir a imagem de um ponto, o primeiro ponto selecionado será o ponto a ser transformado.

**Nota:** a simetria central é equivalente a uma rotação de 180 graus.



**Translação** – cria a imagem de um objeto através de uma translação. Seleccionamos o objeto que será transformado e o vetor de translação. Quando o usuário quiser construir a imagem de um vetor, o primeiro vetor selecionado será o vetor a ser trasladado.



**Rotação** – cria a imagem de um objeto através de uma rotação. Seleccionamos o ponto que será rotacionado, o centro e o ângulo da rotação. Quando quisermos criar a imagem de um ponto, o primeiro ponto selecionado será o ponto a ser transformado.

O ângulo pode ser selecionado a partir de vários tipos de valores:

- **valor numérico** – o ângulo é expresso em radianos. Exemplos de valores numérico: valor livre, um valor devolvido a partir de um *script* **Guile do Dr. Geo**, etc.;

- **medida de um ângulo geométrico formado por três pontos** – sua medida é expressa em graus. Atenção, pois, neste caso, a medida estará no intervalo  $[0; 180]$ ;
- **medida de um ângulo orientado entre dois vetores** – esta medida é expressa em graus e cobre o intervalo  $[-180; 180]$ .

 **Escala (homotetia)** – cria a imagem de um objeto através de uma transformação de escala (ou seja, homotetia). O usuário seleciona o objeto a ser transformado, o centro de homotetia e o fator (isto é, um número). Quando o usuário quiser criar a imagem de um ponto, o primeiro ponto selecionado será o ponto a ser transformado. Por exemplo, se deseja fazer um polígono com um terço de tamanho, seleciona um ponto – centro de homotetia –, o polígono e o valor 0.33333.

#### 2.1.4 Ferramentas numéricas

 **Distâncias, longitudes e números** – cria um valor numérico que pode ser computado ou editado, dependendo do que for selecionado:

- **dois pontos** – a distância entre dois pontos;
- **um segmento** – a longitude desse segmento;
- **um vetor** – a magnitude desse vetor (também chamada norma do vetor);
- **um círculo** – o perímetro do círculo;
- **um arco de círculo** – a longitude do arco;
- **uma linha reta** – a pendente da reta;
- **uma linha reta a um ponto** – a distância entre a linha e o ponto;
- **um clique diretamente sobre o fundo da figura** – permite entrar num valor novo (isto é, num valor numérico livre).

Esta possibilidade é muito interessante, pois permite, por exemplo, fixar uma longitude, o raio e um círculo, a medida de um ângulo (em radianos) ou as coordenadas de um ponto. Esses valores numéricos podem ser, em seguida, utiliza-

dos pelas ferramentas de construção de círculos, de rotação de objetos ou para a criação de pontos, dadas suas coordenadas.



**Ângulo** – calcula a magnitude de um ângulo definido por três pontos ou pelos vetores. No primeiro caso, temos um ângulo não orientado (isto é, um ângulo geométrico com valores no intervalo [0; 180°]. No segundo, um ângulo orientado, e com valores na faixa [-180; 180].



**Coordenadas** – ao selecionar um ponto ou um vetor, temos como resultado as coordenadas do ponto ou as coordenadas do vetor (coordenadas do ponto final menos coordenadas do ponto inicial). Essa ferramenta cria tanto a abscissa como a ordenada.



**Script Guile do Dr. Geo** – cria um *script Scheme* do Dr. Geo (isto é, um *script* em linguagem *Scheme*). O *script* recebe um número que será apresentado na figura. Um *script* pode ser utilizado para obter efeitos colaterais ou pelo valor que dá como resultado. Os *scripts Scheme* do Dr. Geo estão detalhados na seção 3.2, página 27.

### 2.1.5 Ferramentas para macros de construções



**Criar uma macro** – extrai uma sequência de construção dada em uma figura e a converte na macro de uma construção.



**Executar uma macro** – executa (isto é, “lança” ou “corre”) uma macro previamente construída. A macro pode ter sido recém-criada ou pode ser baixada de um arquivo.

**Nota:** as construções e macro estão expostas em “Macros de construções”, na seção 3.1, página 22).

## 2.2 OUTRAS FERRAMENTAS

### 2.2.1. Árvore lógica de construção

Cada figura está associada a uma árvore lógica de construção. Essa árvore é cronológica, ou seja, ela “recorda”, desde a parte alta até a parte baixa, a ordem de construção da figura. Certas entradas da árvore podem ser mobilizadas para que apareçam os antecedentes – isto é, os objetos originadores (objetos-pais) – utilizados na definição do objeto.

Inicialmente, a árvore está oculta (mascarada), recuada para a borda esquerda da janela. Para fazê-la aparecer, não é preciso implementá-la com a ajuda do *mouse*. Mova o ponteiro do *mouse* para a extremidade esquerda da janela do Dr. Geo. Quando este se transformar em < - >, pressione o botão do *mouse* e, mantendo-o pressionado, mova o *mouse* para a direita.

### 2.2.2 Mover a figura

A figura pode ser movida pressionando-se ao mesmo tempo a tecla **Ctrl** e o botão esquerdo do *mouse*.

### 2.2.3 Mover um objeto

Um objeto pode ser movido ao ser arrastado com o *mouse*. A figura é redesenhada, respeitando suas novas posições. Praticamente qualquer tipo de objeto geométrico pode ser movido. Quando necessário, o Dr. Geo move os pontos livres associados à figura. Por exemplo, quando o usuário move uma linha definida por dois pontos livres, o Dr. Geo moverá os dois pontos simultaneamente.

### 2.2.4 Apagar um objeto

Os objetos em uma figura podem ser apagados quando se ativa esse menu. Em qualquer momento, o usuário pode anular o objeto apagado utilizando a função **Desfazer**, da barra de ícones, ou do menu **Editar / Desfazer**. Inicialmente, o número de anulações possíveis é 10, porém o usuário pode ajustar este valor por meio do diálogo de preferências (menu **Editar / Preferências**).

### 2.2.5 Editar a aparência de um objeto

Cada objeto geométrico possui atributos de aparência como som, cor, espessura, nome, tamanho ou formato. Além disso, é possível ocultar temporariamente um objeto sem apagá-lo. Por exemplo: pode ser útil ocultar construções intermediárias sem apagá-las. Todos esses atributos podem ser ajustados a partir de um diálogo ativado quando o usuário seleciona um objeto na figura.

O diálogo de aparência para pontos se relaciona com qualquer tipo de objeto de ponto. A partir dele, é possível ajustar a cor, a forma, o tamanho, o nome e a visibilidade.



Figura 5 – Diálogo de estilo para um objeto tipo ponto

O diálogo de aparência para linhas se refere a linhas retas, semirretas (raios), segmentos, vetores, círculos, arcos de círculos, lugares geométricos. A partir dele, é possível ajustar a cor, a espessura, o nome e sua visibilidade. Quando uma reta, uma semirreta, um vetor ou um segmento são definidos por dois pontos que têm nomes não vazios, então o nome da reta é automaticamente deduzido a partir dos nomes dos pontos que a originam. Neste caso, o usuário não pode renomear a reta.

O diálogo de estilo para os objetos tipo numéricos e tipo polígono se refere a todo tipo de valores (editados pelo usuário, calculados por um *script Scheme* do Dr. Geo ou valores que representam uma medida de um objeto geométrico) e de formas de polígono (note que dois polígonos podem ter cores distintas, pois é possível mudar sua cor).

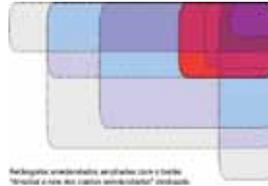


Figura 6 – Diálogo de aparência para objetos tipo linha

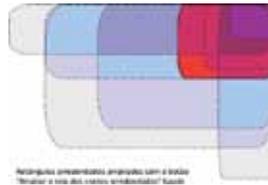


Figura 7 – Diálogo de estilo para objetos numéricos e tipo polígono

### 2.2.6 Editar propriedades de objetos

Certas propriedades dos objetos são ajustáveis pelo usuário. Quando o usuário clica em tais objetos, um diálogo apropriado aparece. Na verdade, os seguintes objetos possuem propriedades ajustáveis pelo usuário:

- **ponto livre** – a abscissa e a ordenada podem ser editadas;

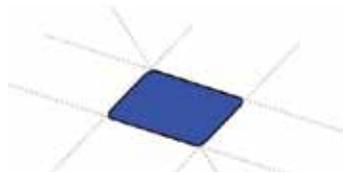


Figura 8 – Editar as coordenadas de um ponto livre

- **valor livre** – seu valor pode ser editado;



Figura 9 – Editar valor livre

- *script* – seu código pode ser editado (o código está escrito em linguagem *Scheme*, que é um dialeto da linguagem Lisp).



Figura 10 –  
Editando um *script*

### 2.2.7 Ativar uma grade

É possível mostrar uma grade unitária dentro de qualquer figura do Dr. Geo. O comando está acessível a partir do menu **Editar / Mostrar** ou **Ocultar grade**. Pode ser também ativado com a tecla **Ctrl + G**. Se o comando é reativado, a grade é ocultada. A grade é unitária: cada subdivisão representa uma unidade. Por último, se a grade está ativada ao salvar a imagem, esta também será salva junto com ela.

## 2.3 PREFERÊNCIAS DE USUÁRIO

### 2.3.1 Comportamento predeterminado

O comportamento predeterminado do Dr. Geo pode ser modificado de várias maneiras. Para ajustar as preferências, clique no menu **Editar / Preferências** para abrir o diálogo de preferências.

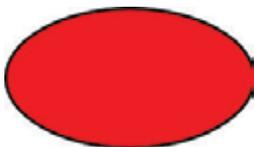


Figura 11 –  
Preferências das  
figuras geométricas

O diálogo se compõe de duas partes. A primeira trata sobre propriedades das figuras geométricas; as diferentes opções permitem regras predeterminadas para cada tipo de objeto (geométrico ou numérico). Essas regras referem-se à aparência dos objetos.

A segunda parte trata sobre preferências globais:

- o nível para desfazer/refazer;
- o nome da figura predeterminada para ser utilizada quando uma nova figura é criada. O %d é anulado por um valor inteiro que é substituído pelo Dr. Geo; este valor é aumentado a cada nova figura criada;
- o navegador da Internet predeterminado para visualizar a ajuda em linha;
- os nomes predeterminados para serem utilizados quando se salva uma figura ou uma seção;
- os nomes predeterminados para exportar nos formatos Latex e PostScript.

### 2.3.2 Outras preferências

Além do comportamento predeterminado do Dr. Geo, o usuário pode mudar o nome de uma figura a partir do menu **Editar / Renomear**.



Figura 12 –  
Renomear a visão  
de uma figura

## 3 CARACTERÍSTICAS AVANÇADAS

Neste capítulo, apresentaremos as características utilizadas para estender as funções do Dr. Geo ou adaptá-lo a uma situação pedagógica dada.

A primeira é a macro de uma construção geométrica. Ela permite a extração da lógica de uma construção em um registro. Feito isso, esse registro pode ser repetido ou salvo em um arquivo com extensão **.mgeo**, e pode ser aberto quando necessário.

Os *scripts Scheme* do Dr. Geo (Dr. Geo *scripts*, em inglês, com a abreviatura DGS) representam outra forma de estender o Dr. Geo. Esses *scripts* são, na realidade, elementos de figuras como qualquer outro elemento geométrico. Eles recebem como entrada as referências aos elementos da figura selecionados pelo usuário e dão como resultado um valor

numérico, que se insere na figura. São, de fato, funções<sup>2</sup> inseridas em uma figura e são avaliados em cada atualização da figura (isto é, cada vez que o programa voltar a desenhar a figura).

Os *scripts* Guile do Dr. Geo podem ser úteis pelo valor que dão como resultado ou por seus efeitos colaterais, dependendo do que o usuário deseja fazer.

Estendendo a funcionalidade dos *scripts* do Dr. Geo, ele pode ir ainda mais além com a figura *Scheme* do Dr. Geo. Esta consiste em um arquivo de texto que contém uma figura geométrica completamente escrita em linguagem de programação *Scheme*. A potência desse modo de criar uma figura consiste em permitir uma construção utilizando o tipo de programação chamado funcional,<sup>3</sup> e não somente declarativo, como é o caso da interface gráfica. Desse modo, o Dr. Geo se converte em um programa que pode ser utilizado como introdução à programação.

Por último, a adaptação de interface de usuário do Dr. Geo permite ao professor preparar uma seção de trabalho com vários documentos em que certas funções têm sido bloqueadas através de uma senha (*password*). O objetivo disso é permitir o uso do programa em certas situações pedagógicas dadas.

### 3.1 MACROS DE CONSTRUÇÕES

Uma macro para construção é uma espécie de procedimento que recebe como entrada elementos de figuras e devolve um ou mais elementos de figuras, construídas pela macro. Uma macro é construída a partir de um modelo definido pelo usuário, que tem que elaborar a série de passos para a construção uma única vez. A partir de então, ele pode ordenar ao Dr. Geo que grave essa série de passos em uma macro, que pode ser salva com extensão **.mgeo**.

Para gravar uma série de passos para a construção, o Dr. Geo necessita saber quais são os iniciais da série e os elementos que se darão como resultado.

---

2 Ou procedimentos para os que iniciarem na linguagem Pascal

3 Permite a recursão (ou recorrência) – forma em que especifica um processo baseado em sua própria definição, por exemplo.

Certamente que os elementos resultantes devem depender apenas dos elementos iniciais;<sup>4</sup> caso contrário, o Dr. Geo não poderia deduzir os elementos produzidos a partir dos dados. Assim, o Dr. Geo segue a lógica da série de passos da construção e armazena os elementos em uma macro. O usuário pode simplesmente executar essa macro para especificar a entrada (para corresponder aos parâmetros exigidos pelo macro) na figura. Em seguida, a macro gera os elementos especificados como um resultado.

**Nota:** os elementos invisíveis das figuras intermediárias também são construídos pela macro. Esses elementos são necessários para criar os elementos resultantes.

Para ilustrar a característica de macros para construções, suponhamos que o usuário deseja gravar a construção de um círculo que passa por três pontos e obter o centro do círculo.

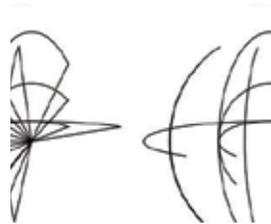


Figura 13 – Figura inicial

Antes da criação da macro, o usuário necessita construir a figura final, que é utilizada como um molde, com a qual se cria a macro.

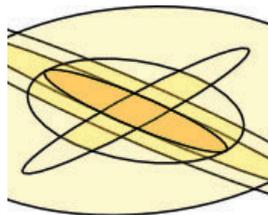


Figura 14 – Figura com a construção resultante

<sup>4</sup> Essa restrição foi posteriormente flexibilizada, facilitando a utilização das macros (ver “Criar um polígono regular”, seção 4.1, página 26 do volume 2 deste Tutorial).

### 3.1.1 Criar uma macro

Nesta etapa, é necessário ordenar ao Dr. Geo a criação de uma macro a partir da série de passos já realizados. Para isso, clique sobre o botão **Construir uma macro**, da barra de ícones ou do menu contextual, que aparece ao se clicar com o botão direito do *mouse* sobre o fundo da figura.

A partir do menu de criação da macro, selecione os parâmetros de entrada e de saída, o nome e a descrição da macro.

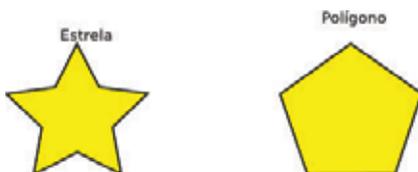


Figura 15 – Primeira janela do menu de criação da macro

A segunda janela do diálogo serve para selecionar os parâmetros de entrada. Selecione os três pontos na figura, que piscarão ao serem selecionados.

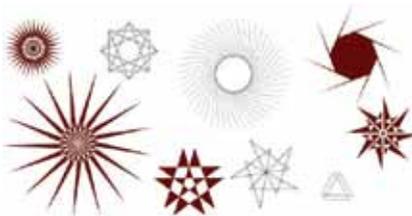


Figura 16 – Segunda janela do menu de criação da macro com os três pontos já selecionados

Na terceira janela do diálogo, selecione os parâmetros de saída. Em nosso exemplo, desejamos que o círculo e o centro sejam os parâmetros de saída para a macro. Para isso, proceda como no caso dos parâmetros de entrada.

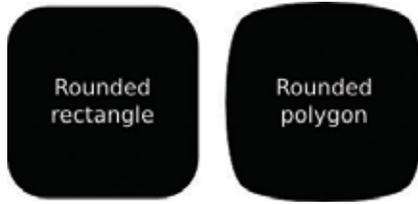


Figura 17 – Terceira janela do menu de criação da macro com o círculo e seu centro já selecionados

Na quarta janela do diálogo, escreva o nome e a descrição da macro. Esses dados são aplicados ao executar uma macro, para distingui-las.

Na última janela do diálogo (a quinta), crie a macro clicando sobre o botão **Aplicar** (finalizar a construção). Alternativamente, é possível regressar para as partes prévias a fim de ajustar os parâmetros da macro.

**Nota:** Se os parâmetros de entrada e de saída não se correspondem (o Dr. Geo não pode extrair a lógica da construção), a macro não pode ser construída. Nesse caso, deve-se reconsiderar a seleção de parâmetros de entrada e de saída, podendo retornar para a segunda ou terceira parte do menu de criação para ajustar suas escolhas. Nesse momento, a macro está construída e salva no Dr. Geo. Na seção seguinte, veremos como utilizá-la.

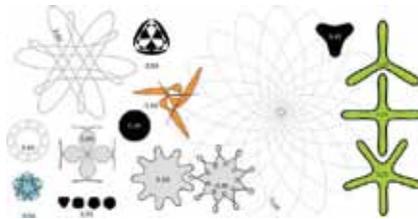


Figura 18 – Quarta janela do menu de criação da macro com nome e descrição

### 3.1.2 Executar uma macro

#### 3.1.2.1 Executar macro através da janela de diálogo

Para executar a macro de uma construção, clique em **Executar macro**, localizado na barra de ícones, ou no menu contextual, que aparece ao se clicar com o botão direito do *mouse* sobre a figura. Uma janela de diálogo que descreve o procedimento se abre.

A partir dessa janela, selecione a macro. Uma nova janela irá abrir. Selecione a macro desejada na primeira caixa. Seleccionada a macro, clique diretamente nos parâmetros de entrada na figura. Após todos os parâmetros de entrada serem selecionados, a macro é executada e os elementos de saída (resultados) serão exibidos.



Figura 19 – Seleção de parâmetros de entrada diretamente na figura

No exemplo, a macro necessita de três parâmetros de entrada (três pontos) e cria a partir deles um círculo e um ponto. Para executar a macro, é necessário uma figura com, ao menos, três pontos. Uma vez que a macro é executada utilizando esses três pontos, obtemos o círculo desenhado e seu centro.



Figura 20 – Figura com três pontos



Figura 21 – Figura resultante com o círculo e seu centro

### 3.1.2.2 Executar macro com a ajuda do menu Macros

Existe outro procedimento, mais rápido, para executar uma macro. A barra de menu principal do Dr. Geo contém um menu **Macros**. Esse menu está ocupado com os nomes das macros carregadas na memória do programa. Para executar uma delas, selecione diretamente a macro de sua preferência.

Além disso, movendo o *mouse* sobre cada item de menu, aparece uma pequena nota informativa para a descrição da macro. O usuário pode, assim, ter uma breve explicação de cada uma das macros.

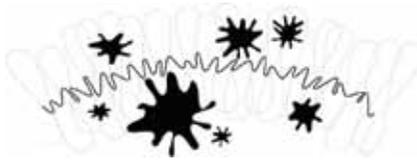


Figura 22 – Executar uma macro diretamente do menu **Macro-construções**

## 3.2 SCRIPT SCHEME DO DR. GEO

O **Dr. Geo** é compatível com o Guile. Isso quer dizer que é possível executar um *script* da linguagem *Scheme* no Dr. Geo. Mas o que é Guile? Lemos no manual do Guile: “Guile é um intérprete da linguagem de programação *Scheme*, concebida para seu uso em uma grande variedade de ambientes”.

A seguinte definição descreve como o Guile é utilizado dentro do Dr. Geo:

Como um *shell*, o Guile pode correr interativamente, lendo expressões do usuário, avaliando-as e mostrando os resultados, ou como um intérprete para um *script*, lendo e executando o código *Scheme* a partir de um arquivo. Por outro lado, o Guile também está disponível

como uma biblioteca (*library*) de objetos, o que permite a outras aplicações facilmente incorporar um intérprete do *Scheme* completo. Um programa pode usar o Guile como uma linguagem de extensão, uma linguagem de configuração simples e poderosa, ou como uma “cola” multiproposital que conecta primitivas elaboradas dentro do programa.

No Dr. Geo, uma **API** (interface de programa para uma aplicação, ou *Application Program Interface*, em inglês) está disponível a partir do intérprete Guile. Esta API é um conjunto de *links* na maquinaria geométrica. Por isso, é possível escrever um *script* (roteiro) para manipular elementos de figuras (geométricas e numéricas). Além disso, por serem os *scripts* figuras como quaisquer outras, podem ser salvos no arquivo da figura. Na sequência, utilizaremos a sigla DGS para nos referirmos a um *script Scheme* do Dr. Geo (ou Dr. Geo *script*).

### 3.2.1 **DGS** por meio de exemplos

A ferramenta utilizada para criar um **DGS** está disponível ao selecionar a seção **Valor numérico**, dentro do menu contextual que se obtém ao clicar-se, com o botão direito do *mouse*, no fundo da figura. Um **DGS** pode receber de **0** a **n** parâmetros de entrada.

Depois de haver escolhido o **DGS**, basta clicar nos objetos geométricos de entrada. Ao terminar de escolher os parâmetros de entrada, deve-se clicar em uma área livre do fundo da figura; assim, o *script* será visível.

A seguir, temos alguns exemplos de **DGS** que permitirão compreender facilmente seu uso e sua potencialidade. Os DGS, assim como as macros, dão uma dimensão particular ao Dr. Geo. Eles permitem, junto com as macros,<sup>5</sup> realizar procedimentos que os autores do programa não incorporaram ou não quiseram incorporar (“**Ir para onde eles não foram ou não quiseram ir**”).

É também importante compreender que a maior parte das funções do intérprete **GNU Guile** estão disponíveis por

---

5 As macros constituem o aspecto geométrico, enquanto que a DGS também inclui o aspecto numérico.

meio do **DGS**. Isso é particularmente certo para as bibliotecas de funções<sup>6</sup> que utilizaremos amplamente.

### 3.2.1.1 **DGS** sem parâmetros de entrada

O procedimento para criar um *script* sem parâmetros de entrada é o seguinte:

- depois de haver escolhido o *script* Guile do Dr. Geo (seção 2.1.4, página 16), clique diretamente no lugar da figura onde se deseja colocar o *script*. Como desejamos que o *script* não tenha parâmetros de entrada, devemos prestar atenção e não clicar erroneamente em algum objeto geométrico, a não ser que o Dr. Geo considere esse objeto como um parâmetro de entrada;<sup>7</sup>
- uma vez que o **DGS** foi criado, aparece na figura a cadeia de caracteres **Dr. Genius**. Todo *script* recém-criado contém um comando predeterminado que mostra essa mensagem. Você pode editá-lo selecionando **Editar propriedades do objeto** (seção 2.2.6, página 19);
- uma vez que esta ferramenta foi selecionada, clique sobre o *script*, ou para ser mais preciso, sobre seu valor, e faça sua escolha. Uma janela se abrirá com o conteúdo do *script* e será permitido editá-lo. Na sequência, utilizaremos esse diálogo para escrever exemplos, como um gerador de números aleatórios. Para isso, seu *script* deve conter unicamente a seguinte linha (lembre-se de clicar em **Aplicar** e, depois, **Fechar**):

(random 10)

Assim, cada vez que a figura é desenhada, este *script* devolve um número inteiro aleatório no intervalo  $(0; 10[$  (ou seja,  $[0; 10)$ ).

Se preferir um número real no intervalo  $[0, 1)$ , utilize o seguinte *script*:

(random:uniform)

Alguns detalhes:

- O valor devolvido pelo *script* é o valor calculado pela última linha do *script*. Em nossos exemplos, trata-se do valor devolvido pela função que utilizamos.

<sup>6</sup> Em particular, as funções matemáticas.

<sup>7</sup> Se por acidente clicarmos sobre um objeto, bastará selecionar novamente a ferramenta Script, no menu, para anular o erro.

- A última linha deve devolver um número real; de outro modo, o Dr. Geo imprimirá “Resultado não imprimível”.
- Se o que se deseja é mostrar o valor de uma variável, basta adicionar essa variável na última linha.

### 3.2.1.2 Cálculo de algumas constantes comuns

Para calcular o valor aproximado de  $\pi$ :

(acos -1)

Para calcular o valor de **e**:

(exp 1)

Os valores devolvidos por estes **DGS** podem ser imediatamente utilizados como todos os demais valores numéricos que o Dr. Geo gera. Pelo descrito anteriormente, os **DGS** são verdadeiramente nossos aliados. No entanto, isso não é tudo: os **DGS** podem fazer muitas coisas mais interessantes quando recebem parâmetros de entrada. Em seguida, veremos como.

### 3.2.1.3 **DGS** com ao menos um parâmetro de entrada

O procedimento para criar um **DGS** com um parâmetro de entrada é essencialmente o mesmo. Imediatamente após selecionar a ferramenta *Script*, basta clicar sobre o objeto que será o parâmetro de entrada e, em seguida, clicar sobre o fundo da figura, no lugar onde se deseja ter o novo *script*.

Em seguida, dentro do *script*, o parâmetro de entrada será referido pela variável **a1**. No caso de termos dois ou mais parâmetros de entrada, utilizaremos, respeitando a ordem em que foram selecionadas, as variáveis **a1**, **a2**, **a3**, **a4**, etc., de acordo com o número de objetos selecionados.

Dependendo do objeto selecionado, diversos métodos estarão disponíveis para obtenção de valores, como coordenadas, longitude, etc. Esses métodos são mostrados na seção 3.2.2 (“Métodos de referência para os *scripts* do Dr. Geo”, na página 34).

Na sequência, exporemos gradualmente um exemplo, em que construiremos a curva de uma função, assim como a tangente em um ponto de curva que poderemos mover. A figura final, mostrada pelo Dr. Geo, é **slope.fgeo** (`/usr/sharedrgeo/examples/figures/slope.fgeo`).



Figura 23 – Figura obtida

**Definir um valor em um intervalo dado:** em uma nova figura, iniciamos colocando dois pontos e o segmento que eles determinam. Sobre esses segmentos, colocamos um ponto livre que chamaremos **Mova-me!** (*Move me*, em inglês). Esse ponto serve para determinar um valor numérico que identifica a posição do ponto dentro do segmento; fazemos isso dentro do *script*. Em seguida, criaremos um *script* que tem com único parâmetro de entrada o ponto **Mova-me!**.

Sendo **a1** a variável no *script* do ponto **Mova-me!**, o *script* seguinte (que chamaremos **Xo**) nos dará um valor decimal compreendido entre [-10; 10].

```
define x (getAbcissa a1))
(*20 (- x 0.5))
```

Cabe esclarecer algumas coisas. Na primeira linha (*getAbcissa a1*), obtemos a **abcissa curvilínea** do objeto *a* que se refere a1.<sup>8</sup> Na segunda, calibramos o valor.

Nesse caso, a expressão significa  $20x(x-0,5)$ . Dado que essa é a última linha do *script*, o resultado desta última operação será o valor que o *script* devolve, e será um número decimal no intervalo [-10; +10], como se desejava.<sup>9</sup>

Esse *script* será denominado **Xo**. Isso pode ser feito no menu contextual: selecione **Outros, Aparência** e, em seguida, clique para colocar o nome do *script*.

**Desenhar o gráfico de uma função:** o valor obtido pelo *script* precedente nos servirá, através de outro *script*, para calcular a imagem do ponto baixo da função:

$x \longrightarrow \cos(x)$

8 Neste ponto, **x** terá um valor entre 0 e 1 inclusive. A “abcissa curvilínea” do objeto é 0 em um extremo, e vai crescendo até tornar-se 1 no outro extremo; é 0,5 na metade, 0,25 a um quarto, etc.

9 Como **x** estava entre 0 e 1,  $20x(x-0,5)$  estará entre -10 e 10.

Este segundo *script* tem como parâmetro de entrada o *script Xo*.

```
(define x (getValue a1))  
(cos x)
```

Em (getValue a1), o objeto a que se refere a1 não é o ponto **Mova-me!**, mas o *script Xo*, e não estamos obtendo a **abscissa curvilínea** de um ponto, mas o valor de um número (neste caso, do *script Xo*, que, dentro do segundo *script*, é chamado a1). Chamamos a este *script Yo*.

Por exemplo, se o ponto **Mova-me!** estava na metade, então sua **abscissa curvilínea** é 0,5, de modo que **Xo** toma o valor  $20 \times (0,5 - 0,5) = 0$ , e **Yo** toma o valor  $\cos(0) = 1$ .

Em seguida, criaremos o ponto **Mo** de coordenadas (**Xo; Yo**), que é um ponto de curva da função  $x \cos(x)$ . Para desenhar a porção do gráfico com domínio  $[-10; 10]$ , criaremos o lugar geométrico do ponto **Mo** quando o ponto **Mova-me!** descreve o segmento.

Com ele, obteremos o gráfico (lembre-se de que, para criar **Mo**, foram selecionados, no menu contextual, ponto e coordenadas e, depois, clicou-se nos *scripts* que dão as coordenadas. Para conseguir o lugar geométrico, é preciso selecionar, no menu contextual **Curva**, lugar geométrico e, em seguida, selecionar os pontos **Mova-me!** e **Mo**).

**Calcular e desenhar a tangente ao gráfico num ponto:** para desenhar a tangente em  $Mo = (Xo, Yo)$ , há a necessidade de ter a inclinação nesse ponto. Utilizando cálculo diferencial, sabemos que a inclinação está dada pela derivada, que, neste caso, é  $x \rightarrow -\sin(x)$ , calculada no ponto **Xo**, de modo que criaremos um *script* que tem como parâmetro de entrada o *script Xo*.

```
(-0 (sin (getValue a1)))
```

A notação prefixa (isto é, **\*23** em vez de **2\*3**), utilizada pelo *Scheme*/Guile, pode parecer pouco intuitiva, porém é questão de acostumar-se a ela e simplesmente dizer "O produto de dois e três" ao invés de "dois por três". Chamaremos o *script* recém-criado de "Inclinação no ponto Mo".

Resta, pois, desenhar a tangente. Para isso, calcularemos, primeiro, as coordenadas de um segundo ponto – **M1** – dessa reta.

Começemos por sua abcissa (não confundir com a **abcissa geométrica**), por exemplo,  $X1=X0+2$ . Para fazer isso, criaremos um *script* que terá como parâmetro de entrada o *script* **Xo**, que chamaremos **X1**:

```
(define x1 (getValue a1))
(+ x1 2)
```

Encontraremos, nesse momento, a ordenada do ponto **M1**. Para isso, necessitamos de:

- **Mo** (referência **a1**);
- a inclinação **Mo** (**a2**);
- a abcissa **X1** (**a3**). (Não confundir com a **abcissa geométrica**.)

Depois de selecionar como parâmetros de entrada o **ponto Mo**, a inclinação em **Mo** (que chamamos “Inclinação no ponto Mo”, ou *slope at Mo*, em inglês) e a **X1**, determinamos a ordenada de **M1**, que chamaremos **Y1**, por meio do seguinte cálculo:

```
Y o + m x (X1 - X0)
(define x0 (car (get Coordinates a1)))
(define y0 (cadr (get Coordinates a1)))
(define m (getValue a2))
(define x1 (get Value a3))
(+ (* m (- x1 x0)) y0)
```

Na relação com a função (`getCoordinates a1`, na qual **a1** deve ser uma referência a um objeto do tipo ponto, o método devolve uma lista que contém as coordenadas do ponto – nesse caso, **Mo**). A instrução `car` permite extrair o primeiro elemento desta lista; a instrução `cadr`, o segundo. O resto do *script* deve ser claro.

Após nomear este último *script* (**Y1**), construímos o ponto  $M1=(X1; Y1)$  e, para terminar, construímos a tangente (**MoM1**). Para isso, já não necessitamos de *scripts*; podemos utilizar o menu, escolhendo **Ponto dadas suas coordenadas e a reta dados dois pontos**.

É claro que teria sido possível utilizar dois ou três *scripts* no lugar da grande quantidade que utilizamos; no entanto, esperamos que esses pequenos exemplos despertem em você o desejo de experimentar por si mesmo os **DGS**.

### 3.2.2 Métodos de referência para os *scripts* do Dr. Geo

As seções seguintes contêm a descrição dos métodos disponíveis para os **DGE**. Esses métodos estão classificados de acordo com o tipo de objeto geométrico ou numérico.

#### **Ponto**

valor (getAbcissa ponto)

**Ponto:** referência a um ponto sobre uma curva.

**Retorna:** **abcissa geométrica** do ponto sobre a curva.

O valor pertence ao intervalo [0; 1].

Exemplo:

```
(define x (getAbcissa a1))  
(* x 1 w0)
```

```
(setAbcissa ponto x)
```

**Ponto:** referência a um ponto livre sobre uma linha.

**X:** valor decimal no intervalo [0; 1], que representa a nova **abcissa geométrica**.

Exemplo:

```
(setAbcissa a1 0.5)
```

```
lista (getCoordinates ponto|vetor)
```

Retornar às coordenadas de um ponto ou de um vetor.

**Ponto vetor:** referência a um ponto ou a um vetor.

**Retorna:** lista que contém as coordenadas do ponto ou do vetor.

Exemplo:

```
(define c (getCoordinates a1))  
(define x (car c))  
(define y (cadr c))  
(+ (* x x) (* y y))
```

```
(setCoordinates ponto coord)
```

Fixa coordenadas de um ponto.

**Ponto:** referência a um ponto livre no plano.

**Coord:** lista de dois números decimais.

Exemplo:

```
(define l (list 1.4 (random 5)))
(setCoordinate a1 l)
```

**Reta, Semirreta, Segmento, Vetor**

valor (getSlope direção)

**Direção:** referência a um objeto do tipo reta, semirreta, segmento ou vetor.

**Retorna:** inclinação nessa direção.

Exemplo:

```
(define p (getSlope a1))
```

valor (getUnit direção)

**Direção:** referência a um objeto do tipo reta, semirreta, segmento ou vetor.

**Retorna:** lista que contém as coordenadas de um vetor unitário na direção dada.

Exemplo:

```
(define v (getUnit a1))
```

valor (getNormal direção)

**Direção:** referência a um objeto do tipo reta, semirreta, segmento ou vetor.

**Retorna:** lista que contém as coordenadas de um vetor normal para a direção dada.

Exemplo:

```
(define n (getNormal a1))
```

valor (getNorm vetor)

**Vetor:** referência a um vetor.

**Retorna:** norma (magnitude, longitude) desse vetor.

Exemplo:

```
(define n (getNorm a1))
```

valor (getLenght segmento)

**Segmento:** referência a um segmento.

**Retorna:** longitude desse segmento.

Exemplo:

```
(define l (getLenght a1))
```

### **Círculo, Arco do círculo**

```
lista (getCenter circulo|arco)
```

**Círculo arco:** referência a um círculo ou a um arco de círculo.

**Retorna:** lista que contém as coordenadas do centro de um círculo ou de um arco de círculo.

Exemplo:

```
(define c (getCenter a1))
```

```
(car c)
```

```
valor (getRadius circulo|arco)
```

**Círculo arco:** referência a um círculo ou a um arco de círculo.

**Retorna:** raio do círculo ou do arco de círculo.

Exemplo:

```
(define r (getRadius a1))
```

```
valor (getLeght circulo|arco)
```

**Círculo arco:** referência a um círculo ou a um arco de círculo.

**Retorna:** perímetro do círculo ou longitude do arco do círculo.

Exemplo:

```
(define | (getLenght a1))
```

### **Funções numéricas**

```
valor (getValue número)
```

**Número:** referência a um número.

**Retorna:** valor desse número.

Exemplo:

```
(define a (getValue a1))
```

```
(define b (getValue a2))
```

```
(+ a b)
```

```
(set Value número v)
```

**Número:** referência a um número.

**v:** valor decimal.

**Exemplo:**

```
(define v (getValue a1))
(setValue a2 v)
```

**Ângulo**

valor (getAngle ângulo)

**Ângulo:** referência a um ângulo orientado ou geométrico.

**Retorna:** medida desse ângulo em graus. Para obter a medida em radianos, utilize a função **getValue**.

**Exemplo:**

```
(define angle1 (getAngle a1))
(define angle2 (getAngle2))
(define angle3 (getAngle a3))
(+ angle1 angle2 angle3)
```

**Outros**

(mova item t)

**Item:** referência a um objeto da figura.

**t:** vetor de duas dimensões.

**Exemplo:**

```
(define v (vector .1 0))
(mova a1 v)
```

**3.3 FIGURAS SCHEME DO DR. GEO**

As figuras *Scheme* do Dr. Geo (**FSD**) são figuras escritas em uma linguagem relativamente natural. Não se trata então de construir uma figura com a ajuda da interface gráfica do Dr. Geo, mas, sim, de descobrir uma figura dentro da linguagem *Scheme* (uma variante de Lisp). Temos muito cuidado para que a sintaxe utilizada seja fácil e concisa. Ademais, o conjunto de palavras-chave utilizadas no *Scheme* para descrever uma figura simples são adaptáveis de modo que apareçam em espanhol, francês, inglês, etc. Uma combinação de vários idiomas é também possível, porém não é recomendado (é por isso que muitas palavras-chave, neste manual, aparecem em português). Basta traduzir um arquivo **SCM** no diretório onde está instalado o Dr. Geo; note que na tradução feita do arquivo, foi incluída uma versão dos comandos com acentos,

assim como outra sem acentos. As palavras-chave não traduzidas serão escritas em inglês.

### 3.3.1 Alguns exemplos

Na sequência, serão utilizadas as palavras **referência** e **nome**. **Referência** é uma variável que aponta para um objeto. **Nome** é o rótulo que aparece na figura. Às vezes, se utilizará o termo variável, porém, **referência** será mais necessário em outros casos.

O *Scheme* em si é uma linguagem de alto nível. Quando uma figura é definida nesta linguagem, é possível, por exemplo, definir recursivamente alguma parte dessa figura ou colocar aleatoriamente certos objetos de maneira que, cada vez que a figura seja carregada, os objetos apareçam ligeiramente distintos. Em poucas palavras, os **FSD** não estão atados às limitantes da interface gráfica e, ademais, possuem linguagem *Scheme*. Um **FSD** é então um arquivo com extensão **.scm** criado com a ajuda de um editor de texto, o qual é aberto no Dr. Geo com a ajuda do comando **Arquivo / Avaliar**.

(nova-figura "Minha figura")

Esse é o menor *script* que podemos definir. Depois de ser carregado no Dr. Geo, ele simplesmente cria uma nova figura vazia com o nome **Minha figura**. Podemos escrever comandos (nova-figura **Minha figura**) quantas vezes desejarmos.

Vejamos, agora, um segundo exemplo:

(nova-figura "Minha figura")

(seja Ponto " A " livre 1.2 -2)

Esse **FSD** define uma figura como um ponto livre **A** de coordenadas iniciais (1, 2; -2). Como podemos ver, a sintaxe da definição de um objeto geométrico é relativamente cômoda, a tal ponto que é expressa na nossa língua materna. Isso é de interesse especial para o ensino fundamental. Na realidade, todos os comandos para a definição de um objeto têm uma sintaxe comum. Tal sintaxe é como segue:

- começa sempre com a palavra-chave **seja**, a qual indica que desejamos definir um novo objeto;

- imediatamente segue a classe do objeto – nesse caso, **Ponto**;
- o nome do objeto vai em seguida; **A** deve sempre estar entre aspas (""). Se não desejamos nomear o objeto, é necessário dar um nome vazio, como segue: "";
- para terminar, necessitamos especificar, dentro da classe, qual é o tipo de objeto; em nosso exemplo, classe é **ponto**, e o tipo é **livre**. Isso significa que o ponto **A** é livre;
- o tipo de objeto é seguido por uma lista de argumentos que depende da classe e do tipo do objeto. Em nosso exemplo, essa lista se compõe de dois números: as coordenadas do ponto livre **A**.

Prossigamos com um terceiro exemplo:

```
(define (triângulo p1 p2 p3)
  (Segmento "" extremos p1 p2)
  (Segmento "" extremos p2 p3)
  (Segmento "" extremos p1 p3))

(define (random)
  (-8 (* 16 (random:uniform))))

(nova-figura "Minha figura")
(seja Ponto " A " livre (random) 0)
(seja Ponto " B " livre 5 0)
(seja Ponto " C " livre (random) 5)

(triângulo A B C)
```

Nesse exemplo, duas palavras estão em inglês: *random*, que significa "azar", e *uniform*, que significa "uniforme". Como se comentou antes, é possível traduzi-las modificando o arquivo dentro do diretório **scm** onde o Dr. Geo está instalado. O arquivo para espanhol é:

`/usr/share/drgeo/scm/drgeo_scm_interface_constant_es.scm.`

Esse exemplo é particularmente interessante nos mostra três aspectos importantes:

- a definição de uma construção de alto nível não prevista pelos autores do Dr. Geo. No programa que acabamos de

ver, temos definido a função triângulo, que, a partir de três pontos, constrói o triângulo que passa por eles. Podemos comparar isso com as macros, porém, com um grau de liberdade muito mais amplo e importante;

- a definição das funções associadas. Neste caso, definimos a função **Random**, que retorna um número decimal compreendido entre -8 e 8. Utilizamos essa função para adicionar um elemento aleatório em certos pontos em nossa figura. Desse modo, cada vez que se carrega a figura, esta será ligeiramente distinta;
- a utilização da palavra-chave como não obrigatória (utilizada para guardar uma referência ao objeto criado). Por exemplo, dentro da função triângulo não guardamos referências para os segmentos criados; por outro lado, quando definimos os pontos **A**, **B** e **C**, necessitamos guardar uma referência para eles, as quais serão nomeadas da mesma forma,<sup>10</sup> porém sem as aspas; assim, nossas referências são: **A**, **B** e **C**. Na sequência, chamaremos essas variáveis de símbolos (essa é a terminologia precisa da linguagem *Scheme*). Desse modo, ao chamar a função triângulo, passamos como parâmetros aos símbolos **A**, **B** e **C**, que são utilizados para definir nossos três segmentos.

Note que, quando definimos os segmentos, não lhes damos nenhum nome; nesse caso, o Dr. Geo lhes dará um nome que está determinado a partir do nome de seus extremos. Nossos três segmentos terão, então, os nomes **[AB]**, **[BC]** e **[AC]**.

Para concluir esta seção, vejamos um último exemplo:

```
(seja Ponto " A " livre 1 0)
(seja Ponto " B " livre 5 0)
(seja Reta " d1 " 2pontos A B)
(envia A cor amarela)
(envia A forma redonda)
(envia A tamanho grande)
(envia B oculto)
(envia d1 espessura pontilhada)
```

<sup>10</sup> Do ponto de vista da linguagem *Scheme*, essas referências são símbolos que apontam para uma estrutura interna do objeto "um protótipo", enquanto os nomes são simplesmente cadeias de caracteres.

Os três primeiros comandos criam dois pontos e uma reta. A parte que nos interessa é a série de comandos **envia**. Esses comandos permitem a comunicação com um objeto cujo símbolo temos guardado; nesse caso, temos os símbolos **A**, **B** e **d1**. O comando consiste em enviar uma mensagem a um objeto. Seu primeiro parâmetro de entrada é o objeto com o qual desejamos nos comunicar; o segundo argumento é a mensagem e o restante dos argumentos estão determinados pelo tipo de mensagem, que é enviada. Por exemplo "(envia A cor amarelo)" envia a mensagem cor com parâmetro de entrada amarelo. O ponto **A** toma então a cor amarela. É fácil compreender o sentido dos outros comandos "envia". Estes serão explicados na seção seguinte.

Terminamos nossa pequena visita guiada para as **Figuras Scheme** do Dr. Geo, **FSD**. Nas seções seguintes, exploraremos o conjunto de comandos disponíveis para definir as **FSD**.

### 3.3.2 Métodos de referência para as figuras *Scheme* do Dr. Geo

A definição dos objetos em um documento **FSD** se faz por meio de protótipos. Os protótipos são objetos que podemos interrogar e modificar, como veremos na sequência. Antes, porém, da definição de algum objeto no **FSD**, este deve ser criado com o comando "nova-figura" no início do arquivo.

#### 3.3.2.1 Comandos gerais

(nova-figura nome)

**Nome:** cadeia de caracteres

**Retorna:** não retorna nenhum valor. Tem este nome pelo efeito que produz, não pelo valor que retorna. Esse comando cria uma figura nova. Os objetos produzidos posteriormente se criam dentro desta figura até que este comando seja solicitado novamente.

Exemplo:

(nova-figura "Mim 1er figura")

#### 3.3.2.2 Definição de objetos em uma figura

Um objeto pode ser definido por meio de várias sintaxes:

- **(seja Ponto "p1" tipo args)** – o ponto é criado e sua

referência salva na variável **p1**. Esta sintaxe utiliza uma macro *Scheme*;

- **(Ponto "Nome" tipo args)** – o ponto é criado, porém nenhuma referência ao ponto é conservada;
- **(define p1 (Ponto "Nome" tipo args))** – o ponto é criado e sua referência é salva na variável **p1**;
- **(set! p1 (Ponto "Nome" tipo args))** – o ponto é criado e sua referência é copiada na variável já existente, **p1**.

Se alguns objetos são criados depois do corpo de uma função, utilize **já seja a forma set!** ou a forma especial do *Scheme* **let**. É importante destacar que a chamada se faz a uma função que devolve uma referência ao objeto criado.

Para saber mais sobre a correspondência entre os nomes dos comandos *Scheme* em espanhol e em inglês, veja o arquivo `/ur/share/drgeo/scm/drgeo/interface_constant_es.scm`.

## Ponto

protótipo (Ponto nome livre x y)

**Nome:** cadeia de caracteres que designa o nome do objeto.

**x:** abcissa do ponto.

**y:** ordenada do ponto.

**Retorna:** referência a um ponto livre do plano com coordenadas iniciais **x** e **y**.

Exemplo:

```
(define p1 (Ponto " A " livre 1.2 (acos -1)))
```

protótipo (Ponto nome en-linha linha x)

**Nome:** cadeia de caracteres que designa o nome do objeto.

**linha:** referência a uma linha (reta, semirreta, segmento, arco, círculo, etc.).

**x:** abcissa curvilínea (a **abcissa geométrica**) de ponto livre; o valor pertencente ao intervalo [0 ; 1).

**Retorna:** referência a um ponto livre sobre a curva.

Exemplo:

```
(Ponto "M" en-linha s1 0.5)
```

protótipo (Ponto nome metade-2pts p1 p2)

**Nome:** cadeia de caracteres que designa o nome do objeto.

**p1:** referência a um ponto.

**p2:** referência a um ponto.

**Retorna:** referência ao ponto médio dos dois pontos.

Exemplo:

(seja Ponto " A " livre 1 1)

(seja Ponto " B " livre 4 4)

(Ponto " l " metade-2pts A B)

protótipo (Ponto nome metade-segmento s)

**Nome:** cadeia de caracteres que designa o nome do objeto.

**s:** referência a um segmento.

**Retorna:** referência à metade de um segmento.

Exemplo:

(Ponto " L " metade-segmento s)

protótipo (Ponto nome intersecção I1 I2)

**Nome:** cadeia de caracteres que designa o nome do objeto.

**I1:** referência a uma linha (reta ou curva).

**I2:** referência a uma linha (reta ou curva).

**Retorna:** referência ao ponto de interseção das duas linhas.

Exemplo:

(Ponto "l" intersecção reta segmento)

protótipo (Ponto nome intersecção2 I1 I2)

**Nome:** cadeia de caracteres que designa o nome do objeto.

**I1:** referência a uma linha (reta ou curva).

**I2:** referência a uma linha (reta ou curva).

**Retorna:** referência ao segundo ponto de interseção das duas linhas quando uma delas é do tipo arco de círculo ou do tipo círculo.

Exemplo:

(Ponto "l" intersecção2 reta círculo)

## Reta

protótipo (Reta nome 2pontos p1 p2)

**Nome:** cadeia de caracteres que designa o nome do objeto.

**I1:** referência a um ponto.

**I2:** referência a um ponto.

**Retorna:** referência a uma reta que passa pelos dois pontos.

## Exemplo:

(seja Ponto "A" livre 0 0)

(seja Ponto "M" livre 1 2)

(Reta " " 2pontos A M)

protótipo (Reta nome paralela p d)

**Nome:** cadeia de caracteres que designa o nome do objeto.

**p:** referência a um ponto.

**d:** referência a uma direção (reta, segmento, vetor...).

**Retorna:** referência a uma reta paralela na direção **d** e que passa pelo ponto **p**.

Exemplo:

(seja Ponto "A" livre 1 5)

(seja Reta "d1" paralela A d)

protótipo (Reta nome perpendicular p d)

**Nome:** cadeia de caracteres que designa o nome do objeto.

**p:** referência a um ponto.

**d:** referência a uma direção (reta, segmento, vetor...).

**Retorna:** referência a uma reta perpendicular para a direção de **d** e que passar por **p**.

Exemplo:

(seja Ponto "A" livre 1 5)

(seja Reta "d1" perpendicular A d)

## Semirreta

protótipo (Semirreta nome 2pontos o p)

**Nome:** cadeia de caracteres que designa o nome do objeto.

**o:** referência a um ponto, origem da semirreta.

**p:** referência a um ponto, ponto da semirreta.

**Retorna:** referência a uma semirreta definida por sua origem e por um ponto.

Exemplo:

(seja Ponto "A" livre 1 5)

(seja Ponto "O" livre 0 0)

(seja Semirreta "dd1" 2pontos A O)

## Segmento

protótipo (Segmento nome extremos p1 p2)

**Nome:** cadeia de caracteres que designa o nome do objeto.

**p1:** referência a um ponto.

**p2:** referência a um ponto.

**Retorna:** referência a um segmento definido por seus extremos.

**Exemplo:**

(seja Ponto "A" livre 1 5)

(seja Ponto "B" livre 10 4)

(seja Segmento " " extremos A B)

## Círculo

protótipo (Círculo nome 2pontos c p)

**Nome:** cadeia de caracteres que designa o nome do objeto.

**c:** referência a um ponto, centro do círculo.

**p:** referência a um ponto sobre o círculo.

**Retorna:** referência a um círculo definido por seu centro e por um ponto.

**Exemplo:**

(seja Ponto "A" livre 1 5)

(seja Ponto "B" livre 10 4)

(seja Círculo "C1" 2pontos A B)

protótipo (Círculo nome centro-raio c r)

**Nome:** cadeia de caracteres que designa o nome do objeto.

**c:** referência a um ponto, centro do círculo.

**r:** referência a um valor numérico, raio do círculo.

**Retorna:** referência a um círculo definido por seu centro e por seu raio.

**Exemplo:**

(seja Ponto "A" livre 1 5)

(seja Número "r" livre 10)

(seja Círculo "C1" centro-raio A r)

protótipo (Círculo nome centro-segmento c s)

**Nome:** cadeia de caracteres que designa o nome do objeto.

**c:** referência a um ponto, centro do círculo;

**s:** referência a um segmento dado em que a longitude de segmento será o raio do círculo.

**Retorna:** referência a um círculo definido por seu centro e por um segmento que tem como raio sua longitude.

Exemplo:

(seja Ponto "A" livre 1 5)

(seja Círculo "C1" centro-segmento A s)

## Arco de círculo

protótipo (Arco nome 3pontos p1 p2 p3)

**Nome:** cadeia de caracteres que designa o nome do objeto.

**p1:** referência a um ponto, 1º extremo do arco do círculo.

**p2:** referência a um ponto do arco.

**p3:** referência a um ponto, 2º extremo do arco do círculo.

**Retorna:** referência a um arco de círculo definido por seus extremos e por um ponto interno.

Exemplo:

(seja Ponto "A" livre 1 5)

(seja Ponto "B" livre, 0 5)

(seja Ponto "C" livre -1 -2)

(seja Arco "arc" 3pontos A B C)

## Polígono

protótipo (Polígono nome npontos args)

**Nome:** cadeia de caracteres que designa o nome do objeto.

**Args:** lista de referências de pontos; vértices do polígono.

**Retorna:** referência a um polígono definido por seus vértices.

Exemplo:

(seja Polígono "quad" npontos A B C D)

### 3.3.2.3 As transformações geométricas

Os protótipos das transformações geométricas permitem a obtenção de objetos rotacionados, refletidos, trasladados, a escala, etc. Esses protótipos utilizam como referência os tipos ponto, reta, semirreta, vetor, círculo, arco do círculo e polígono.

protótipo (TipoDeObjeto nome rotação objeto centro ângulo)

**TipoDeObjeto:** ponto, segmento, reta, semirreta, vetor, círculo, arco, polígono.

**Nome:** cadeia de caracteres que designa o nome do objeto.

**objeto:** referência do objeto a ser transformado.

**centro:** referência a um ponto, centro da rotação.

**ângulo:** referência a um valor numérico, ângulo da rotação.

**Retorna:** referência do objeto já transformado.

Exemplo:

(seja Ponto "I1" rotação I C a)

protótipo (TipoDeObjeto nome escala objeto centro k)

**TipoDeObjeto:** ponto, segmento, reta, semirreta, vetor, círculo, arco, polígono.

**Nome:** cadeia de caracteres que designa o nome do objeto.

**objeto:** referência do objeto a ser transformado.

**centro:** referência a um ponto, centro da escala (centro de homotetia).

**k:** referência a um valor numérico, fator da escala (homotetia).

**Retorna:** referência do objeto transformado.

Exemplo:

(seja Polígono "P1" escala P C k1)

protótipo (TipoDeObjeto nome simetria objeto centro)

**TipoDeObjeto:** ponto, segmento, reta, semirreta, vetor, círculo, arco, polígono.

**Nome:** cadeia de caracteres que designa o nome do objeto.

**objeto:** referência do objeto a ser transformado.

**centro:** referência a um ponto, centro da simetria (isto é, uma rotação de 180 graus).

**Retorna:** referência do objeto já transformado.

Exemplo:

(seja Segmento "S1" simetria S C)

protótipo (TipoDeObjeto nome reflexão objeto eixo)

**TipoDeObjeto:** ponto, segmento, reta, semirreta, vetor, círculo, arco, polígono.

**Nome:** cadeia de caracteres que designa o nome do objeto.

**objeto:** referência do objeto a transformar.

**eixo:** referência a uma reta, eixo da reflexão.

**Retorna:** referência do objeto já transformado.

Exemplo:

(seja Polígono "P1" reflexão P d1)

protótipo (TipoDeObjeto nome translação objeto vetor)

**TipoDeObjeto:** ponto, segmento, reta, semirreta, vetor, círculo, arco, polígono.

**Nome:** cadeia de caracteres que designa o nome do objeto.

**objeto:** referência do objeto a ser transformado.

**vetor:** referência a um vetor.

**Retorna:** referência do objeto já transformado.

Exemplo:

(seja Círculo "C1" traslação C v)

## Lugar geométrico

protótipo (Lugar-geométrico nome2 pontos m c)

**Nome:** cadeia de caracteres que designa o nome do objeto.

**m:** referência a um ponto móvel sobre uma linha (reta, círculo, etc.).

**c:** referência a um ponto fixo que depende do ponto **m**.

**Retorna:** referência ao lugar geométrico de **c** quando **m** se move sobre a linha (reta, círculo, etc.).

Exemplo:

(Lugar-geométrico "locus1" 2pontos M l)

## Vetor

protótipo (Vetor nome 2pontos o e)

**Nome:** cadeia de caracteres que designa o nome do objeto.

**o:** referência a um ponto, origem do vetor.

**e:** referência a um ponto, extremo do vetor.

**Retorna:** referência a um vetor.

Exemplo:

(seja Ponto "B" livre 0 5)

(seja Ponto "C" livre -1 -2)  
(Vetor " " 2pontos C B)

## Número

protótipo (Número nome livre x y v)

**Nome:** cadeia de caracteres que designa o nome do objeto.

**x,y:** coordenadas do número.

**v:** valor inicial do número.

**Retorna:** referência a um número livre.

Exemplo:

(seja Número "pi" livre 5 5 (acos -1))

protótipo (Número nome longitude-segmento x y s)

**Nome:** cadeia de caracteres que designa o nome do objeto.

**x,y:** coordenadas do número.

**s:** referência a um segmento.

**Retorna:** referência a um número, longitude de um segmento.

Exemplo:

(seja Número "l" longitude-segmento 5 5 S)

protótipo (Número nome norma-vetor x y v)

**Nome:** cadeia de caracteres que designa o nome do objeto.

**x,y:** coordenadas do número.

**s:** referência a um vetor.

**Retorna:** referência a um número, norma (longitude) de um vetor.

Exemplo:

(seja Número "l" norma-vetor 5 5 V)

protótipo (Número nome ponto-círculo x y p c)

**Nome:** cadeia de caracteres que designa o nome do objeto.

**x,y:** coordenadas do número.

**p:** referência a um ponto.

**c:** referência a um círculo.

**Retorna:** referência a um número, distância entre o ponto e o círculo.

Exemplo:

(seja Número "l" ponto-círculo 5 5 P C)

protótipo (Número nome ponto-reta x y p d)

**Nome:** cadeia de caracteres que designa o nome do objeto.

**x,y:** coordenadas do número.

**p:** referência a um ponto.

**c:** referência a uma reta.

**Retorna:** referência a um número, distância entre o ponto e a reta.

Exemplo:

(seja Número "d" ponto-reta 5 5 M D1)

protótipo (Número nome ponto-ponto x y p1 p2)

**Nome:** cadeia de caracteres que designa o nome do objeto.

**x,y:** coordenadas do número.

**p1:** referência a um ponto.

**p2:** referência a um ponto.

**Retorna:** referência a um número, distância entre os dois pontos.

Exemplo:

(seja Número "d" ponto-ponto 5 5 A B)

protótipo (Número nome longitude-círculo x y c)

**Nome:** cadeia de caracteres que designa o nome do objeto.

**x,y:** coordenadas do número.

**c:** referência a um círculo.

**Retorna:** referência a um número, perímetro do círculo.

Exemplo:

(seja Número "p" longitude-círculo 5 5 C)

protótipo (Número nome inclinação-linha x y d)

**Nome:** cadeia de caracteres que designa o nome do objeto.

**x,y:** coordenadas do número.

**d:** referência a uma reta.

**Retorna:** referência a um número, declive (inclinação) da reta dada.

Exemplo:

(seja Número "p" declive-linha 5 5 d1)

protótipo (Número nome longitude-arco x y arc)

**Nome:** cadeia de caracteres que designa o nome do objeto.

**x,y:** coordenadas do número.

**arc:** referência a um arco de círculo.

**Retorna:** referência a um número, longitude do arco de círculo dado.

Exemplo:

(seja Número "1" longitude-arco 5 5 ABC)

## Ângulo

protótipo (Ângulo nome geométrico A B C)

**Nome:** cadeia de caracteres que designa o nome do objeto.

**A:** referência a um ponto.

**B:** referência a um ponto, vértice do ângulo.

**C:** referência a um ponto.

**Retorna:** referência a um ângulo geométrico.

Exemplo:

(seja ângulo "a" geométrico A B C)

### 3.3.2.4 Modificação dos atributos (aparências) dos objetos

Para modificar os atributos de um objeto já criado, utilizamos um sistema de envio de mensagens diretamente ao protótipo que representa o objeto em questão. A modificação, e seus atributos, é resultado desse processo e se faz assim que os objetos foram criados.

(envia objeto cor valor)

**objeto:** referência a um objeto.

**valor da cor:** os valores possíveis são: negro, cinza-escuro, cinza, branco, verde-escuro, verde, azul-escuro, azul, vermelho, vermelho-escuro, amarelo e laranja.

Exemplo:

(seja Ponto "A" livre 1 2)

(envia A cor verde)

(envia linha espessura valor)

**linha:** referência a uma linha (reta, semirreta, círculo, lu-

gar geométrico, etc.).

**valor da espessura:** os valores possíveis são: pontilhado, pequeno e grande.

**Exemplo:**

(seja Ponto "A" livre 1 2)

(seja Ponto "O" livre 0 0)

(seja Reta "d" 2ponto,s A B)

(envia "d" espessura pontilhada)

(envia ponto tamanho valor)

**ponto:** referência a um ponto.

**valor do tamanho do ponto:** os valores possíveis são: pequeno, normal e grande.

**Exemplo:**

(seja Ponto "A" livre 1 2)

(envia A tamanho pequeno)

(envia ponto forma valor)

**ponto:** referência a um ponto.

**valor:** a forma do ponto – os valores possíveis são: redondo, cruz, redondo-vazio, quadrado e quadrado-vazio.

**Exemplo:**

(seja Ponto "A" livre 1 2)

(envia A forma redondo)

(envia objeto oculto)

**objeto:** referência a um objeto que será ocultado.

**Exemplo:**

(seja Ponto "A" livre 1 2)

(envia A oculto)

### 3.3.3 Sinônimos dos comandos das figuras *Scheme* do Dr. Geo

Esta seção consta de uma transcrição ligeiramente modificada do arquivo (/usr/shar/drgeo/scm/drgeo\_scm\_interface\_contant\_es.scm).

Ele define os sinônimos dos comandos utilizados para escrever uma figura *Scheme* em português. Esses sinônimos

são sempre definidos a partir da versão de referência dos comandos, que está em inglês.

Na verdade, é possível combinar o código em inglês com o código com sinônimos em espanhol, francês, português, etc. Contudo, se você deseja que sua figura seja compreendida por qualquer grupo linguístico, sugerimos utilizar o inglês.

Por exemplo, os seguintes comandos são sinônimos:

(lets Point "P" free 3 3)(send A 'color bordeaux)

(seja Ponto "P" livre 3 3)(envia A cor vermelho-escuro)

Note que algumas poucas terminações são iguais em português e em inglês. Além disso, estão traduzidas várias versões, de modo que é possível evitar escrever acentos.

#### Cores

negro	black
cinza-escuro	dark-grey
cinza	grey
branco	white
verde-escuro	dark-green
verde	green
azul-escuro	dark-blue
azul	blue
vermelho	red
vermelho-escuro	bordeaux
amarelo	yellow
alaranjado	orange
laranja	orange

#### Espessura

pontilhado	dashed
pequeno	small
grande	large
normal	normal

### Formas de Ponto

redondo	round
redonda	round
cruz	cross
quadrado	rec
quadrada	rec
redondo-vazio	round-empty
redonda-vazia	round-empty
quadrado-vazio	rec-empty
quadrado-vazio	rec-empty
quadrada-vazia	rec-empty
quadrada-vazia	rec-empty
rec-vazia	rec-empty
rec-vazia	rec-empty

### Estilos

cor	color
espessura	thickness
forma	shape
tamanho	size
tamanho	size
mascarado	masked
oculto	masked
ocultar	masked

### Pontos

livre	free
sobre-a-curva	on-curve
sobre-curva	on-curve
em-curva	on-curve
em-linha	on-line
meio-2pts	middle-2pts
meio-segmento	middle-segment
intersecção	intersection

intersecção	intersection
intersecção2	intersection2

### Segmentos

extremos	extremities
----------	-------------

### Comuns aos pontos, linhas, etc.

2pontos	2points
2pts	2points
3pontos	3points
3pts	3points

### Linhas

paralela	parallel
ortogonal	orthogonal
perpendicular	orthogonal
a circunferência	to circle
centro-raio	center-radius
centro-segmento	center-segment

### Polígonos

npontos	npoints
---------	---------

### Números

longitude-segmento	segment-length
norma-vetor	vector-norm
norma-vetor	vector-norm
ponto-reta	point-line
ponto-círculo	point-circle
ponto-circulo	point-circle
ponto-ponto	point-point
perímetro-círculo	circle-length
perímetro-circulo	circle-length
longitude-círculo	circle-length
longitude-circulo	circle-length
inclinação-linha	line-slope
inclinacao-linha	line-slope

inclinacao-reta	line-slope
longitude-arco	arc-length
<b>Ângulos</b>	
geométrico	geometric
geometrico	geometric
orientado	oriented
<b>Transformações</b>	
rotação	rotation
rotacão	rotationW
escala	scale
escala	scale
simetria	symmetry
reflexão	reflexion
reflexao	reflexion
translação	translation
translacao	translation
traslação	translation
<b>Geral</b>	
seja	lets
nova-figura	new-figure
figura-nova	new-figure
enviar	send
envia	send
Ponto	Point
Reta	Line
Linha	Line
Segmento	Segment
Raio	Ray
Raio	Ray
Círculo	Circle
Circulo	Circle
Arcos	Arc

Lugar-geométrico	Locus
Lugar-geometrico	Locus
Vetor	Vector
Numérico	Numeric
Numerico	Numeric
Númerico	Numeric
Numérico	Numeric
Ângulo	Angle
Angulo	Angle
Polígono	Polygon
Poligono	Polygon

### 3.3.4 Galeria de exemplos

Para ilustrar a utilização das figuras *Scheme* do Dr. Geo, proporemos uma breve série de exemplos. Neles mostraremos o potencial das **FSD** e esperamos igualmente que sejam uma fonte de inspiração. Para cada um desses exemplos, damos o código-fonte no *Scheme* da figura e, em seguida, o resultado. O código-fonte pode ser copiado dentro de um arquivo de texto com um editor, e salvo com a extensão **.scm**, podendo ser avaliado depois pelo Dr. Geo.

O código seguinte se encontra em inglês. Veja a seção 3.3.3, "Sinônimos dos comenados das figuras *Scheme* do Dr. Geo", página 52, para escrever a maior parte do código em português.

#### **Polígono regular**

Para construir um polígono regular, com um número arbitrário de lados, podemos fazer uso de uma função recursiva no *Scheme*.

```
(define pi (acos -1))
(define n 15)
(define x0 0)
(define y0 0)
```

```

(define p1 0)
(define (polygon center p a n)
  (if (> n 0)
      (begin
        (set! p1 (Point " " rotation p center a))
        (send p1 masked)
        (Segment " " extremities p p1)
        (polygon center p1 a (- n 1))))
      (new-figure "Regular Polygon!")
      (lets Point "C" free x0 y0)
      (lets Numeric "a" free 0 0 (*2 (/ pi n)))
      (send a masked)
      (set! p1 (Point "I" free 5 0))
      (lets Segment "S" extremities C p1)
      (Segment " " rotation S C a)
      (polygon C p1 a n)

```



Figura 24 –  
Polígono regular  
de 15 lados

## Fractal

A construção de uma curva fractal com a forma de uma árvore se consegue facilmente com uma figura *Scheme*. O código-fonte da figura é surpreendentemente breve, especialmente quando comparado com a construção a mão livre, utilizando a interface gráfica.

```

(new-figure "Baum")
(lets Numeric "A1" free 2 2 +3.4)
(lets Numeric "A2" free 2 3 -3.7)
(lets Numeric "S1" free 2 4 +0.5)
(lets Numeric "S2" free 2 5 +0.9)
(define(dec n)
  (- n 1))
(define (inc n)
  (+ n 1))
(define (invisible p)
  (send p masked)p)

```

```

(define (scalerot oP C a s)
  (let* ((sP (invisible (Point " " scale oP C s)))
        (rP (invisible (Point " " rotation sP C a))) )
    rP))
(define (Zweig p0 p1 n)
  (Segment " " extremities p0 p1)
  (left* ((left-scale (if (odd? n) S1 S2))
         (left-angle A1)
         (right-scale (if (odd? n) S2 S1))
         (right-angle A2) )
        (if (>n 0)
            (begin
              (Zweig p1 (scalerot p0 p1 left-angle left-scale) (dec n))
              (Zweig p1 (scalerot p0 p1 right-angle right-scale) (dec n))))))
  (lets Point "A" free -3 0)
  lets Point "B" free -3 2)
  (Zweig A B 6)

```



Figura 25 – Curva fractal que simula a representação de uma árvore

## REFERÊNCIAS

FERNANDES, Hilaire; CENTOMO, Andrea; SOTO, Adrian. **Manual de usuário de Dr. Geo**. Disponível em: <<http://documentation.ofset.org/drgeo/es/drgeenius.html>>. Acesso em: 05 fev. 2009.

SECRETARIA DE ESTADO DA EDUCAÇÃO  
Diretoria de Tecnologia Educacional  
Rua Salvador Ferrante, 1651 - Boqueirão  
CEP 81670-390 - Curitiba - PR  
[www.diaadiaeducacao.pr.gov.br](http://www.diaadiaeducacao.pr.gov.br)