

# PCS 2190 - IPR: Introdução ao Processing

Ricardo Nakamura e Romero Tori

2015

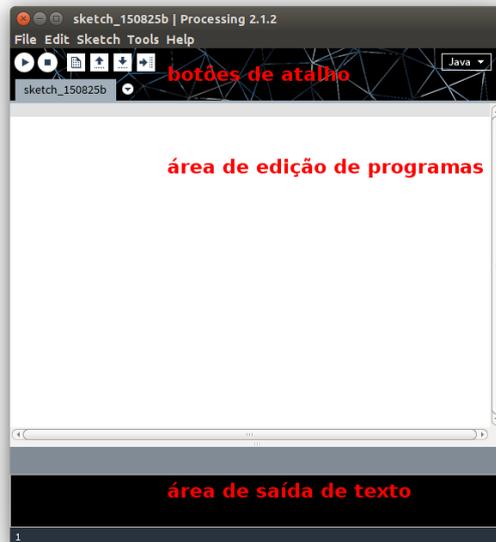
## 1 Introdução

O Processing é um software desenvolvido com o objetivo de servir como um ambiente de programação acessível e intuitivo. Por este motivo, ele é adotado nesta disciplina como ferramenta para o ensino de lógica de programação. O site oficial do Processing é <http://www.processing.org>

Por outro lado, o Processing é baseado na linguagem Java, fazendo com que a experiência adquirida no uso deste software possa ser aplicada por aqueles que tenham interesse em se aprofundar em técnicas de programação.

## 2 O ambiente de edição

Ao iniciar o Processing, uma janela com conteúdo semelhante ao mostrado na figura a seguir será apresentada. Na área superior estão os menus de comandos e uma série de botões de atalho para as operações mais comuns. A área central da janela é ocupada por um editor de textos, onde os comandos de programação podem ser inseridos. O nome do arquivo que está sendo editado é exibido no topo desta área de edição. Na parte inferior da janela, existe uma área com fundo preto que é usada para exibir mensagens de texto do programa.



Os programas criados em Processing são geralmente conhecidos como “sketches”, remetendo à ideia de esboço, experimentação. Sempre que iniciamos o Processing, ele sugere como nome do arquivo o termo “sketch\_” seguido de uma abreviatura do mês e dia atual.

### 3 Programas de computador

Como visto anteriormente, um programa de computador é uma sequência de instruções que descrevem as ações que desejamos que o computador realize. Embora computadores sejam capazes de realizar milhões de operações por segundo (especialmente operações aritméticas), estas máquinas possuem capacidade bastante limitada de interpretar informações baseadas em seu contexto.

Dessa forma, as instruções que compõem um programa de computador precisam ser completas e detalhadas. Além disso, as linguagens humanas dependem muito de contexto: uma mesma palavra pode ter vários significados, pode-se escrever uma mesma frase de diferentes maneiras etc. Assim, as linguagens de programação utilizam uma estrutura simplificada que seja mais facilmente interpretável pelo computador.

Por exemplo, ao invés de escrever num programa uma instrução como “desenhe uma linha diagonal do canto da folha de papel até o seu

centro”, no Processing escrevemos, simplesmente:

```
line(0, 0, 200, 200);
```

Como se pode ver, é uma forma bastante “minimalista” de comunicação. Adicionalmente, o computador só reconhece um vocabulário pequeno de comandos, como por exemplo o comando **line** no exemplo anterior. Os comandos disponíveis dependem da linguagem de programação utilizada.

O ponto-e-vírgula colocado ao final de cada linha serve para indicar onde termina um comando. Dessa forma, podemos dividir comandos longos em mais de uma linha, pois eles só serão considerados completos quando for encontrado o próximo ponto-e-vírgula.

## 4 Seu primeiro programa em Processing

Inicie o Processing e digite o seguinte programa na área de edição:

---

```
size(400, 400);  
line(0, 0, 200, 200);
```

---

Em seguida, clique no primeiro botão (da esquerda para a direita) da barra de atalhos, ou selecione o menu Sketch → Run ou, ainda, pressione a combinação de teclas **CTRL + R**. Qualquer uma destas ações irá fazer com que o Processing execute o seu programa. Quando o programa é executado, uma pequena janela quadrada com fundo cinza é exibida na tela com uma linha preta indo do canto superior esquerdo até o meio da janela. Os programas mais simples de Processing são feitos desta forma: digitando comandos na área de edição, na ordem em que eles devem ser executados.

Vamos ver o que cada comando do programa faz:

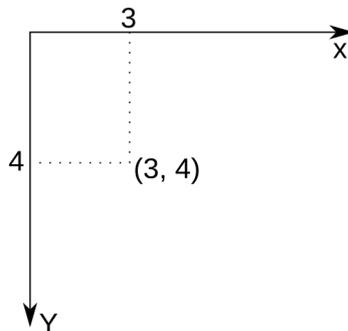
```
size(400, 400);
```

O comando na primeira linha configura o Processing para utilizar uma janela gráfica de 400 x 400 pixels. O pixel é a unidade mínima de desenho na tela.

```
line(0, 0, 200, 200);
```

O segundo comando é responsável por desenhar a linha diagonal. No Processing, as informações adicionais para um comando são passadas entre parênteses, separadas por vírgulas. O comando **line** requer as coordenadas de dois pontos, correspondentes ao início e término do segmento de reta a ser desenhado. Assim, neste caso a reta é desenhada entre os pontos (0, 0) e (200, 200).

As coordenadas de pontos no Processing são sempre indicadas na ordem (coordenada horizontal, coordenada vertical) ou (X, Y). Elas são medidas em pixels e variam de 0 até o tamanho da janela gráfica. Por isso é importante começar o programa com um comando **size**, que define esse tamanho.



Ao contrário do que você pode estar acostumado, a coordenada vertical começa no alto da janela e aumenta para baixo (diferentemente das coordenadas cartesianas tradicionais). Assim, a posição (0, 0) corresponde ao canto superior esquerdo da janela gráfica!

## Atividade 1

Modifique o programa para desenhar uma linha do canto inferior esquerdo até o canto superior direito da janela gráfica.

## 5 Trabalhando com tons de cinza

No nosso primeiro programa, desenhamos uma linha preta sobre um fundo acinzentado. Modifique o programa para que fique como na listagem a seguir, para desenhar uma linha branca sobre fundo preto:

```
size(400, 400);  
background(0);  
stroke(255);  
line(0, 0, 200, 200);
```

O comando **background** é usado para pintar a janela toda de uma determinada cor. Quando informamos somente um número para esse comando, ele indica um tom de cinza, sendo que zero corresponde a preto e 255 corresponde a branco.

O mesmo raciocínio vale para o comando **stroke**, que indica a cor das linhas e contornos que serão desenhados a partir daquele momento.

### Atividade 2

Faça um programa que desenha um “X” na janela gráfica.

## 6 Comentários

Podemos acrescentar anotações e comentários em nossos programas, começando uma linha com //. Isso faz com que o Processing ignore o restante da linha. Por exemplo, a listagem a seguir mostra o uso de comentários no programa que foi feito anteriormente.

```
// cria uma janela de 400 x 400 pixels  
size(400, 400);  
// pinta o fundo de preto  
background(0);  
// escolhe a cor branca para desenho  
stroke(255);  
// traça uma linha diagonal  
line(0, 0, 200, 200);
```

Na prática, não é necessário escrever comentários para **cada linha do programa** – isso pode inclusive atrapalhar a sua leitura. Por outro lado, comentários podem ser úteis para anotar as partes mais complexas do programa.

## 7 Alterando a ordem dos comandos

Em um programa de computador, os comandos são realizados exatamente na ordem em que são escritos. Por este motivo, essa ordem é importante e, muitas vezes, mudar a ordem dos comandos dá resultados diferentes. Por exemplo, na listagem a seguir simplesmente levamos o comando **background** para o final do programa.

---

```
size(400, 400);  
stroke(255);  
line(0, 0, 200, 200);  
background(0);
```

---

Note que nesta ordem, o computador irá abrir uma janela gráfica de tamanho 400x400 pixels. Em seguida selecionará a cor branca para desenho de linhas e desenhará uma linha diagonal. Por fim, irá pintar a janela de preto – apagando, com isso, a linha desenhada!

Considere agora a próxima listagem, que desenha três linhas na tela. O que acontece se você alternar a posição dos comandos **line** entre si?

---

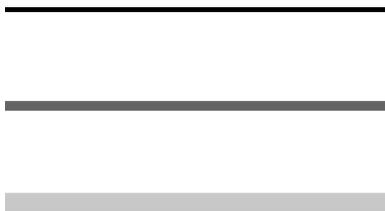
```
size(400, 400);  
background(255);  
strokeWeight(10);  
stroke(0);  
line(0, 100, 400, 100);  
line(0, 200, 400, 200);  
line(0, 300, 400, 300);
```

---

Esse programa também mostra um novo comando: **strokeWeight**, usado para modificar a espessura das linhas e contornos. O valor passado para o comando corresponde à espessura das linhas.

## Atividade 3

Modifique o programa anterior para criar uma imagem como a mostrada a seguir.



## 8 Outros comandos de desenho

O Processing conta com vários outros comandos de desenho, além do **line**. Alguns deles são listados a seguir:

- **rect(x, y, l, a)**: retângulo com canto superior esquerdo (x, y), largura l e altura a;
- **ellipse(x, y, a, b)**: elipse com centro em (x,y), diâmetro horizontal a e diâmetro vertical b;
- **triangle(x1, y1, x2, y2, x3, y3)**: triângulo ligando os pontos (x1, y1), (x2, y2) e (x3, y3);
- **quad(x1, y1, x2, y2, x3, y3, x4, y4)**: quadrilátero ligando os pontos (x1, y1), (x2, y2), (x3, y3) e (x4, y4);

O programa a seguir ilustra o uso de alguns desses comandos. Note também a presença do comando **fill**, que é usado para definir a cor de preenchimento das formas geométricas.

---

```
size(400, 400);
background(0);
stroke(255);
fill(128);
rect(100, 100, 200, 200);
ellipse(250, 250, 100, 100);
triangle(100, 200, 200, 200, 200, 100);
```

---

Se quiser que uma forma geométrica seja desenhada sem preenchimento, use o comando **noFill**. Se quiser que uma forma geométrica seja desenhada sem contorno, use o comando **noStroke**. Note que **noStroke** faz com que linhas desenhadas com o comando **line** não sejam visíveis!

## 9 Trabalhando com cores

Os comandos **background**, **stroke** e **fill** também podem receber três valores, ao invés de um só. Neste caso, eles definem uma cor no padrão RGB (vermelho, verde, azul). Alguns exemplos de cores e seu uso no comando **fill** são mostrados a seguir:

---

```
fill(255, 0, 0);    // vermelho
fill(0, 255, 255); // ciano
fill(100, 100, 100); // cinza médio, o mesmo que fill(100)
```

---

O programa a seguir cria um gradiente de tons de verde, usando os comandos de cores e desenho de retângulos.

---

```
size(400, 400);
// não queremos bordas nos retângulos
noStroke();
fill(255, 0, 0);
rect(0, 0, 100, 400);
fill(200, 0, 0);
rect(100, 0, 100, 400);
fill(150, 0, 0);
rect(200, 0, 100, 400);
fill(100, 0, 0);
rect(300, 0, 100, 400);
```

---

## Atividade 4

Modifique o programa anterior para desenhar cinco retângulos de cores diferentes. Os retângulos não precisam ter o mesmo tamanho. Procure fazer um arranjo visualmente interessante de cores ou proporções.

## 10 Resumo

Nesta apostila, nós tivemos o primeiro contato com o ambiente do Processing e começamos a entender como utilizar sua interface. Fizemos nossos primeiros programas neste ambiente e com isso, pudemos formular uma primeira definição do que é um programa de computador: uma lista de comandos que são executados pelo computador sequencialmente, do primeiro ao último.

## Atividade 5

Faça um programa para reproduzir o desenho abaixo. Não se esqueça de acrescentar comentários onde for necessário.

