

# Processing.org – como usar (1)



deti

universidade de aveiro  
departamento de electrónica,  
telecomunicações e informática

```
import processing.core.*;  
import java.util.*;
```

Biblioteca do Processing

```
public class PBatalhaNaval1 extends PApplet {  
    int LADO = 100;  
    int LINHAS = 5, COLUNAS = 5;  
    int tiro = 11;  
    int[][] tabuleiro = new int[LINHAS][COLUNAS];  
    Random generator = new Random();
```

A nossa Classe herda o ambiente do Processing

```
    public static void main(String args[]) {  
        PApplet.main("PBatalhaNaval1");  
    }
```

A função main lança aplicação como um applet (aplicações java para web) do Processing

## Processing – como usar (2)



deti

universidade de aveiro  
departamento de electrónica,  
telecomunicações e informática

```
public void settings() {  
    size(500, 500);  
}
```

Esta função do Processing é usada para definir o tamanho da tela gráfica

```
public void setup() {  
    noStroke();  
    // coloca barcos aleatoriamente  
    for (int b = 0; b < 3; b++) {  
        int nextl = generator.nextInt(5);  
        int nextc = generator.nextInt(5);  
        tabuleiro[nextl][nextc] = 1;  
    }  
}
```

Esta função do Processing é corrida uma vez, no início do programa

			1	1
	1			

# Processing – como usar (3)



deti

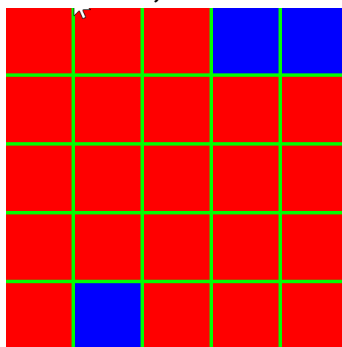
universidade de aveiro  
departamento de electrónica,  
telecomunicações e informática

```
public void draw() {  
  background(0, 255, 0);  
  // desenha tabuleiro  
  for (int l = 0; l < LINHAS; l++) {  
    for (int c = 0; c < COLUNAS; c++) {  
      fill(255, 0, 0);  
      if (tabuleiro[l][c] < 0) {  
        fill(0, 0, 255);  
      }  
      rect(c * LADO, l * LADO, LADO - 5, LADO - 5);  
    }  
  }  
}
```

Esta função do Processing fica em ciclo a correr cerca de 60 vezes por segundo

Esta função do Processing é corrida quando se carrega num botão do rato

```
public void mousePressed() {  
  int l, c;  
  c = mouseX / LADO;  
  l = mouseY / LADO;  
  if (tabuleiro[l][c] > 0) {  
    tabuleiro[l][c] = -tabuleiro[l][c];  
  }  
}
```



# Processing – como usar (4)

The screenshot shows the Geany IDE interface. The main editor displays a Java file named `Turma.java` with the following code:

```
1 import java.util.Scanner;
2
3 /*
4  * Created by jam on 26-out-2016.
5  * Ler notas de 0 a 20 (termi
6  * arrays 2 dimensões: id alu
7  */
8
9 public class Turma {
10
11     public static void main(
12         // Scanner para leitura de dados do teclado
13         Scanner ler = new Scanner(System.in);
14         final int NALUNOS = 10;
15         int[][] turma= new int[NALUNOS][5];
16         int id, soma=0, num=0;
17         float media;
18         System.out.println("Introduza as notas dos al
19         System.out.printf("ID do aluno: ");
20         id = ler.nextInt();
```

The **Build** menu is open, showing options: **Compile** (F8), **Make** (Shift+F9), **Make Custom Target...** (Shift+Ctrl+F9), **Make Object** (Shift+F8), **Next Error**, **Previous Error**, and **Execute** (F5). The **Set Build Commands** dialog box is open, showing the following configuration:

#	Label	Command	Working directory	Reset
<b>Java commands</b>				
1.	Compile	javac "%f"		
2.				
3.				
Error regular expression:				
<b>Independent commands</b>				
	Make	make		
	Custom Target...	make		
	Make Object	make %e.o		
expression:				
opens a dialog and appends the response to the command.				
<b>Commands</b>				
	Execute	java -cp "D:\processing-3.0\		

The status bar at the bottom left indicates "This is Geany 1.25."

Para usar o Processing no Geany ou outros IDEs  
Tem de se incluir a biblioteca `core.jar` do Processing na classpath.

No Geany incluir caminho para o diretório :  
`java -cp "D:\processing-3.0\core\library\core.jar;." "%e"`

Ou copiar o `core.jar` para o diretório equivalente do vosso jdk:  
`C:\Program Files (x86)\Java\jdk1.8.0_92\jre\lib\ext\core.jar`