
	<p>FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS TECNICATURA UNIVERSITARIA EN DISEÑO INTEGRAL DE VIDEOJUEGOS FACULTAD DE INGENIERÍA Universidad Nacional de Jujuy PENSAMIENTO COMPUTACIONAL y PROGRAMACIÓN: Problema y Solución – PC y P – Algoritmos – Principio de la P</p>	
---	--	---

FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS

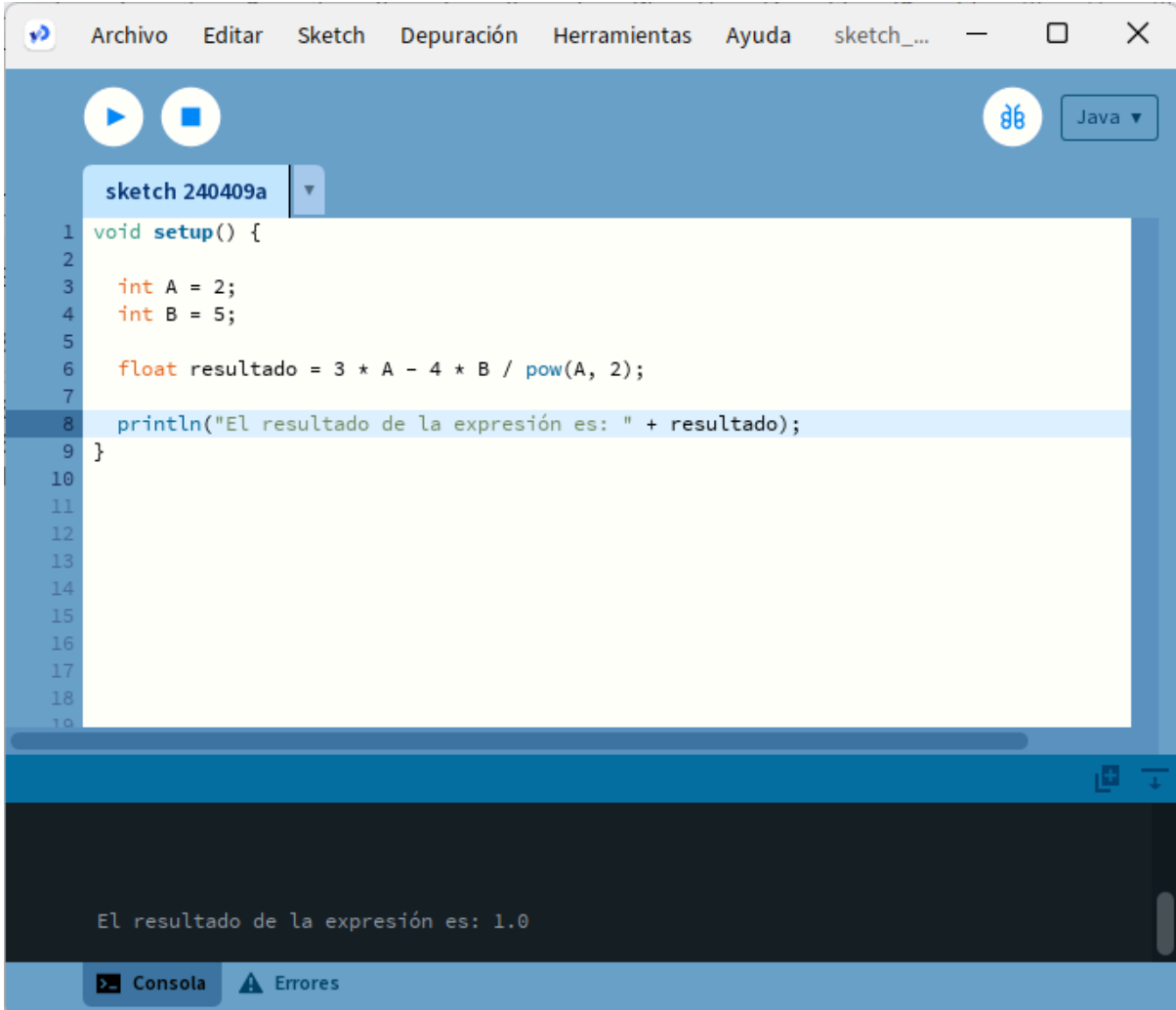
Trabajo Práctico N° 1

Jose Omar Murillo
LU: TUV000289

Profesores:
Mg. Ing. Ariel Alejandro Vega
Ing. Carolina Cecilia Apaza
Año

Ejercicio 1: Evaluar y obtener resultado de la siguiente expresión para $A = 2$ y $B = 5$

$3 * A - 4 * B / A ^ 2$
 $(3 * 2) - ((4 * 5) / (2 ^ 2))$
 $6 - (20 / 4)$
 $6 - 5$
 1



```

1 void setup() {
2
3   int A = 2;
4   int B = 5;
5
6   float resultado = 3 * A - 4 * B / pow(A, 2);
7
8   println("El resultado de la expresión es: " + resultado);
9 }
10
11
12
13
14
15
16
17
18
19

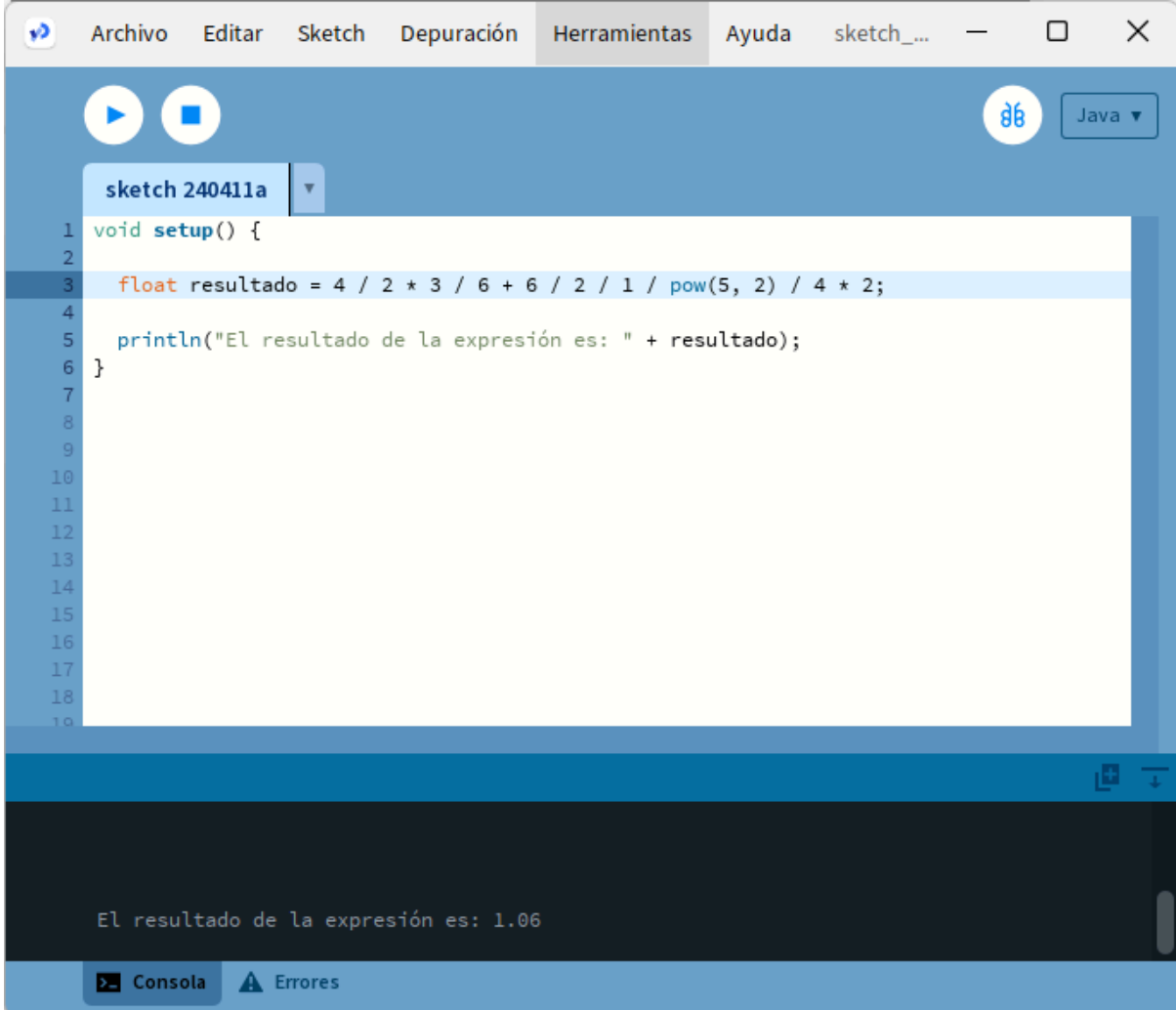
```

El resultado de la expresión es: 1.0

Ejercicio 2: Evaluar la siguiente expresión $4 / 2 * 3 / 6 + 6 / 2 / 1 / 5 ^ 2 / 4 * 2$

$(4 / 2 * 3 / 6) + (6 / 2 / 1 / (5 ^ 2) / 4 * 2)$
 $1 + (6 / 2 / 1 / 25 / 4 * 2)$
 $1 + (0.03 * 2)$
 $1 + 0.06$

1.06



```
void setup() {  
  float resultado = 4 / 2 * 3 / 6 + 6 / 2 / 1 / pow(5, 2) / 4 * 2;  
  println("El resultado de la expresión es: " + resultado);  
}
```

El resultado de la expresión es: 1.06

Ejercicio 4: Evaluar las siguientes expresiones aritméticas, para lo cual indicar en el caso de las variables, el valor indicado. Luego escribirlas como expresiones algebraicas.

a) $b^2 - 4 * a * c$

$b = 4$

$a = 2$

$c = 1$

$4^2 - 4 * 2 * 1$

$16 - 8$

8



$$4^2 - 4 * 2 * 1$$

```
1 int b = 4;
2 int a = 2;
3 int c = 1;
4
5 void setup() {
6
7     float resultado = pow(b, 2) - 4 * a * c;
8
9     println("El resultado de la expresión es: " + resultado);
10 }
11
12
13
14
15
16
17
18
19
```

El resultado de la expresión es: 8.0

b) $3 * x^4 - 5 * x^3 + x^2 - 17$

$x = 2$

$$(3 * (2^4)) - (5 * (2^3)) + 2 * 12 - 17$$

$$(3 * 16) - (5 * 8) + 2 * 12 - 17$$

$$48 - 40 + 24 - 17$$

$$15$$

$$3 * 2^4 - 5 * 2^3 + 2 * 12 - 17$$



```
1 int x = 2;
2
3 void setup() {
4
5     float resultado = 3 * pow(x, 4) - 5 * pow(x, 3) + x * 12 - 17;
6
7     println("El resultado de la expresión es: " + resultado);
8 }
9
10
11
12
13
14
15
16
17
18
19
```

El resultado de la expresión es: 15.0

c) $(b + d) / (c + 4)$

b = 3

d = 5

c = 4

$(3 + 5) / (4 + 4)$

8 / 8

1

$(3 + 5) / (4 + 4)$



```
1 int b = 3;
2 int c = 4;
3 int d = 5;
4
5 void setup() {
6
7   float resultado = (b + d) / (c + 4);
8
9   println("El resultado de la expresión es: " + resultado);
10 }
11
12
13
14
15
16
17
18
19
```

El resultado de la expresión es: 1.0

d) $(x^2 + y^2)^{1/2}$

$x = 2$

$y = 4$

$(2^2 + 4^2)^{1/2}$


$(4 + 16)^{1/2}$

$20^{1/2}$

$\sqrt{20}$

4.47

$(2^2 + 4^2)^{1/2}$



```

1  int x = 2;
2  int y = 4;
3
4  void setup() {
5
6      float resultado = pow(pow(x, 2) + pow(y, 2), 0.5);
7
8      println("El resultado de la expresión es: " + resultado);
9  }
10
11
12
13
14
15
16
17
18
19

```

El resultado de la expresión es: 4.472136

Ejercicio 5: Si el valor de A es 4, el valor de B es 5 y el valor de C es 1, evaluar las siguientes expresiones

a) $B * A - B^2 / 4 * C$

$A = 4$

$B = 5$

$C = 1$

$(5 * 4) - (5^2 / 4 * 1)$

$20 - (25 / 4)$

$20 - 6.25$

13.75



```
1 int A = 4;
2 int B = 5;
3 int C = 1;
4
5 void setup() {
6
7   float resultado = B * A - (pow(B, 2) / 4 * C);
8
9   println("El resultado de la expresión es: " + resultado);
10 }
11
12
13
14
15
16
17
18
19
```

El resultado de la expresión es: 13.75

b) $(A * B) / 3^2$

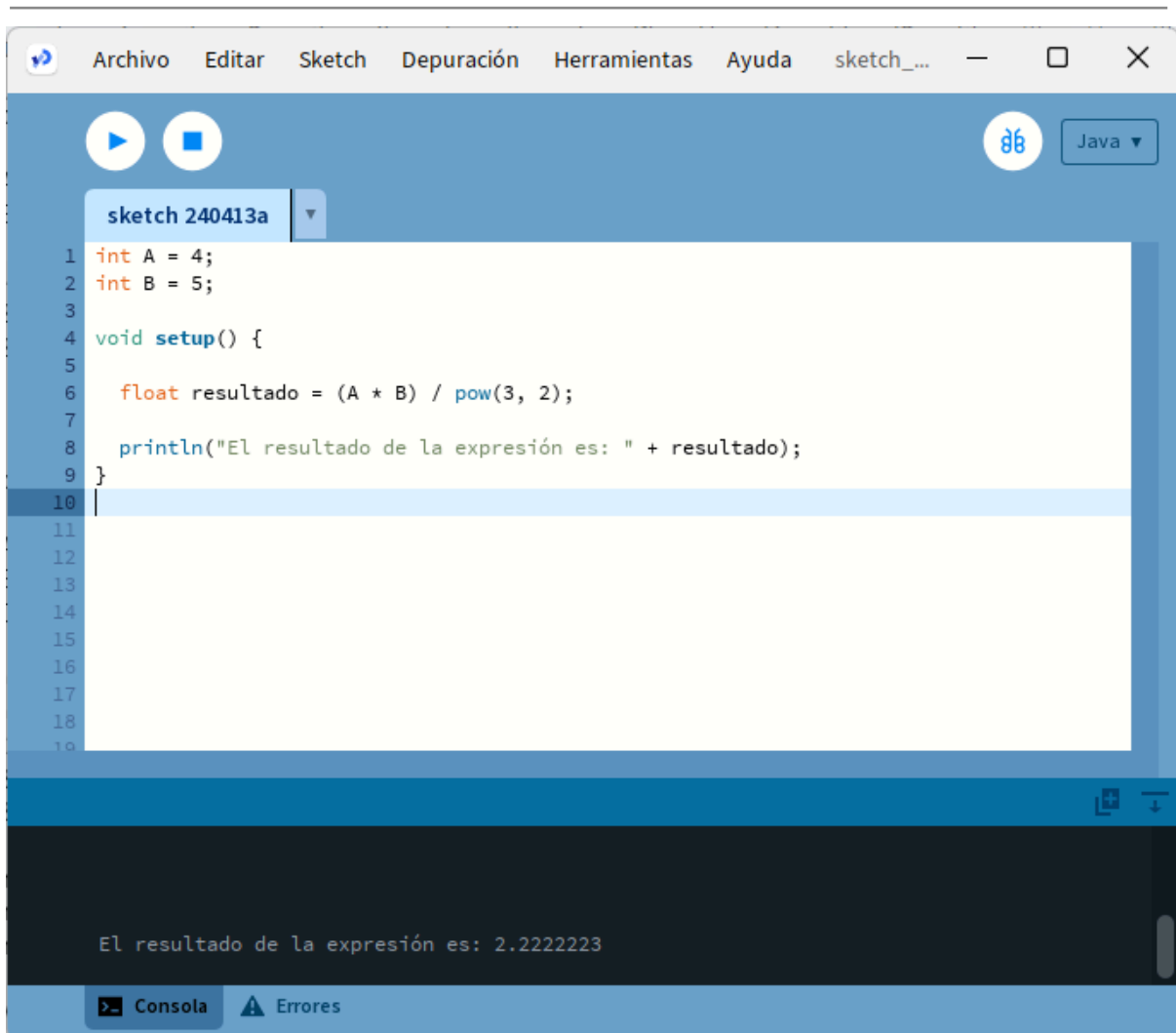
A = 4

B = 5

$(4 * 5) / 3^2$

20 / 9

2.22



```
1 int A = 4;
2 int B = 5;
3
4 void setup() {
5
6   float resultado = (A * B) / pow(3, 2);
7
8   println("El resultado de la expresión es: " + resultado);
9 }
10
11
12
13
14
15
16
17
18
19
```

El resultado de la expresión es: 2.2222223

c) $((B + C) / 2 * A + 10) * 3 * B) - 6$

A = 4

B = 5

C = 1

$((5 + 1) / 2 * 4 + 10) * 3 * 5) - 6$

$(6 / 2 * 4 + 10) * 3 * 5) - 6$

$(22 * 3 * 5) - 6$

$330 + 6$

324



```
1 int A = 4;
2 int B = 5;
3 int C = 1;
4
5 void setup() {
6
7   float resultado = (((B + C) / 2 * A + 10) * 3 * B) - 6;
8
9   println("El resultado de la expresión c) es: " + resultado);
10 }
11
12
13
14
15
16
17
18
19
```

El resultado de la expresión c) es: 324.0

Ejercicio 6: Para $x=3$, $y=4$; $z=1$, evaluar el resultado de

$$R1 = y + z$$

$$R2 = x \geq R1$$

$$x = 3$$

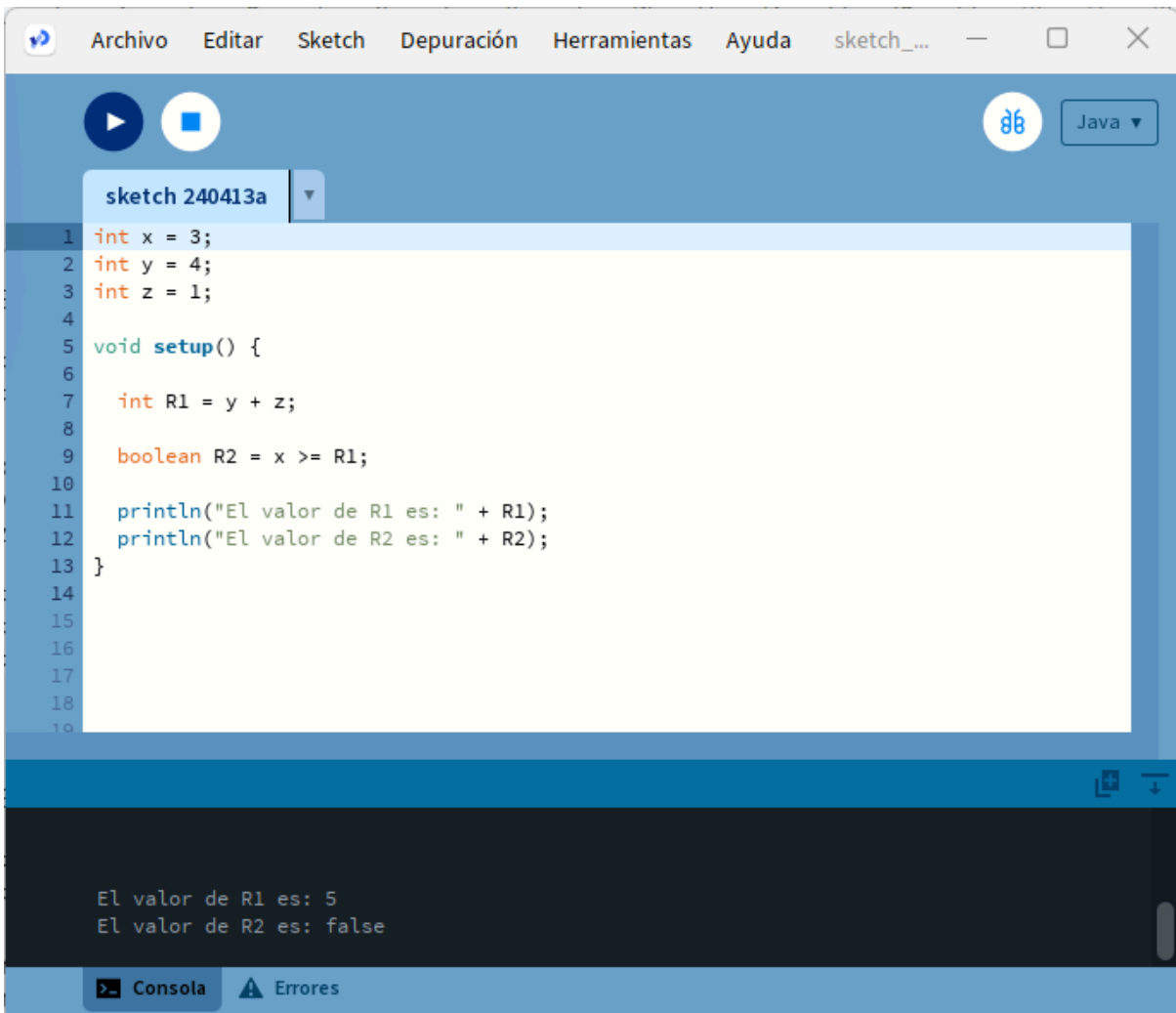
$$y = 4$$

$$z = 1$$

$$R1 = 4 + 1$$

$$R1 = 5$$

$$R2 = 3 \geq 5$$



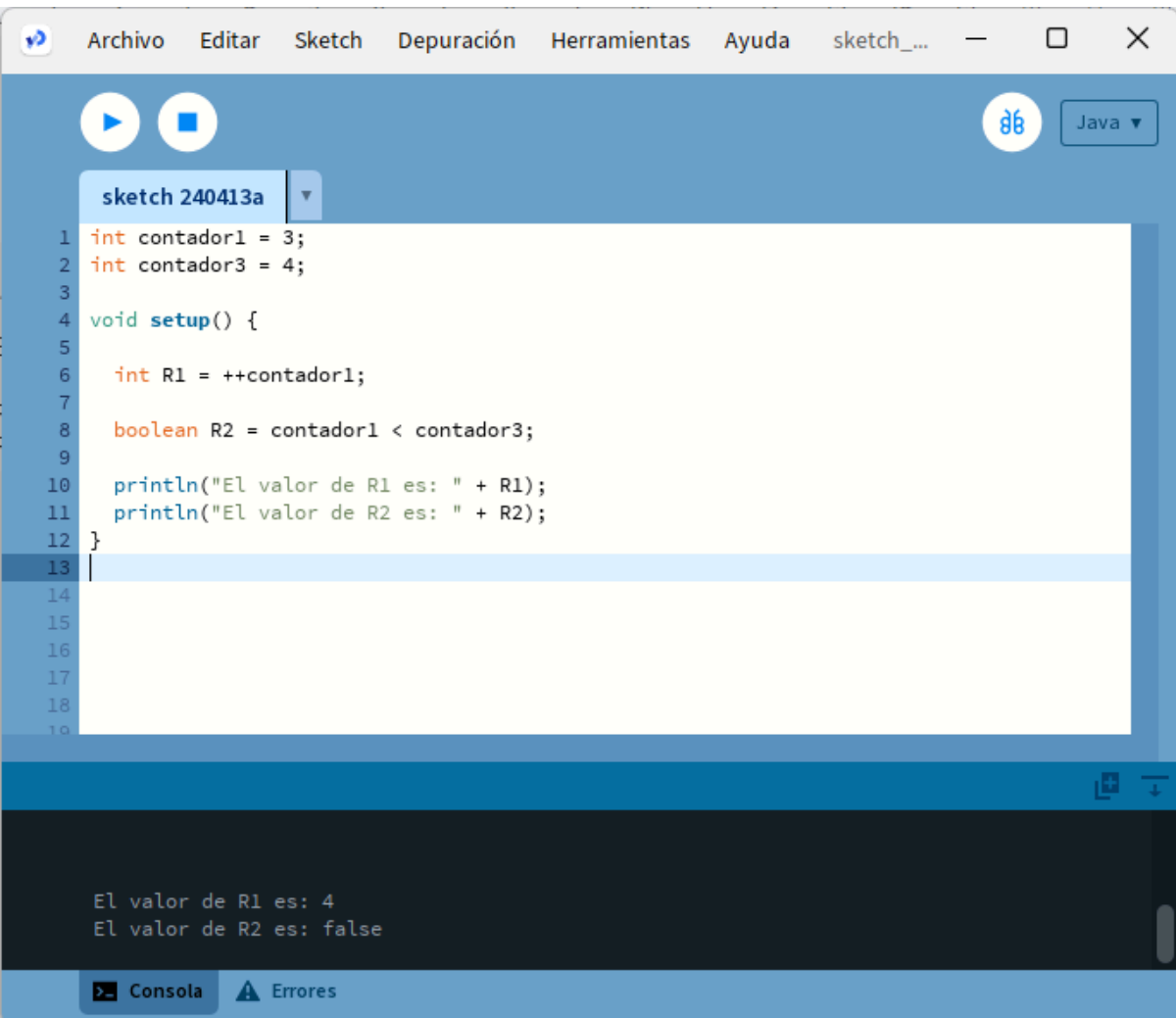
```
1 int x = 3;
2 int y = 4;
3 int z = 1;
4
5 void setup() {
6
7     int R1 = y + z;
8
9     boolean R2 = x >= R1;
10
11     println("El valor de R1 es: " + R1);
12     println("El valor de R2 es: " + R2);
13 }
14
15
16
17
18
19
```

El valor de R1 es: 5
El valor de R2 es: false

Ejercicio 7: Para contador1=3, contador3=4, evaluar el resultado de

R1 = ++contador1

R2 = contador1 < contador3



```
1 int contador1 = 3;
2 int contador3 = 4;
3
4 void setup() {
5
6   int R1 = ++contador1;
7
8   boolean R2 = contador1 < contador3;
9
10  println("El valor de R1 es: " + R1);
11  println("El valor de R2 es: " + R2);
12 }
13
14
15
16
17
18
19
```

El valor de R1 es: 4
El valor de R2 es: false

Ejercicio 8: Para $a=31$, $b=-1$; $x=3$, $y=2$, evaluar el resultado de $a+b-1 < x*y$

$$31 + (-1) - 1 < 3 * 2$$

$$29 < 6$$

false



```
1 int a = 31;
2 int b = -1;
3 int x = 3;
4 int y = 2;
5
6 void setup() {
7
8   boolean resultado = a + b - 1 < x * y;
9
10  println("El resultado de la expresión es " + resultado);
11 }
12
13
14
15
16
17
18
19
```

El resultado de la expresión es false

Ejercicio 9: Para $x=6$, $y=8$, evaluar el resultado de $!(x<5) \ \&\& \ !(y>=7)$

$!(6 < 5) \ \&\& \ !(8 >= 7)$

true && false

false



```
1 int x = 6;
2 int y = 8;
3
4 void setup() {
5
6   boolean resultado = !(x < 5) && !(y >= 7);
7
8   println("El resultado de la expresión es: " + resultado);
9 }
10
11
12
13
14
15
16
17
18
19
```

El resultado de la expresión es: false

Ejercicio 10: : Para i=22,j=3, evaluar el resultado de $!(i > 4) \parallel !(j \leq 6)$

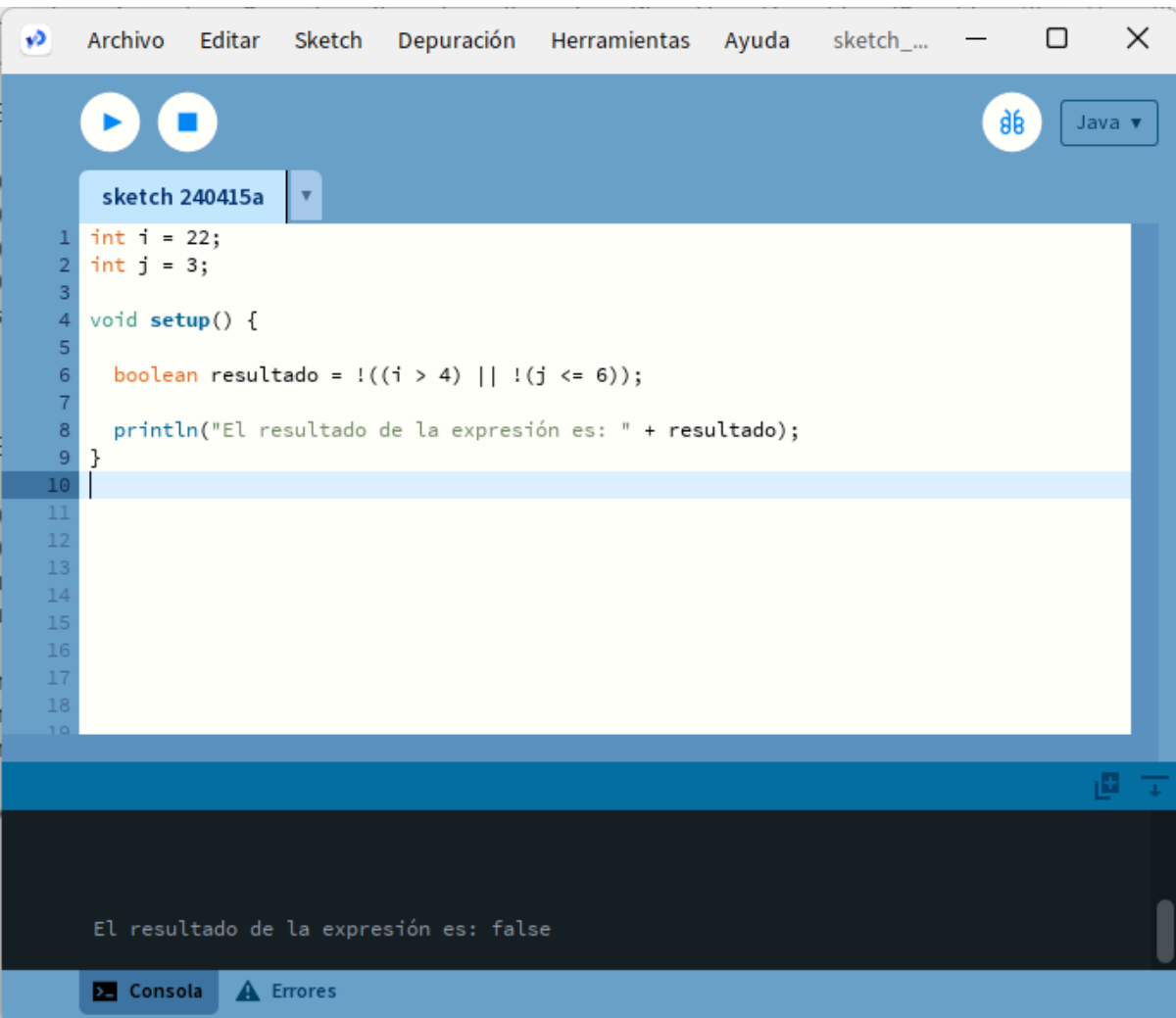
$!((22 > 4) \parallel !(3 \leq 6))$

$!((\text{true} \parallel !(\text{true}))$

$!((\text{true} \parallel \text{false})$

$!((\text{true})$

false



```
1 int i = 22;
2 int j = 3;
3
4 void setup() {
5
6   boolean resultado = !((i > 4) || !(j <= 6));
7
8   println("El resultado de la expresión es: " + resultado);
9 }
10
11
12
13
14
15
16
17
18
19
```

El resultado de la expresión es: false

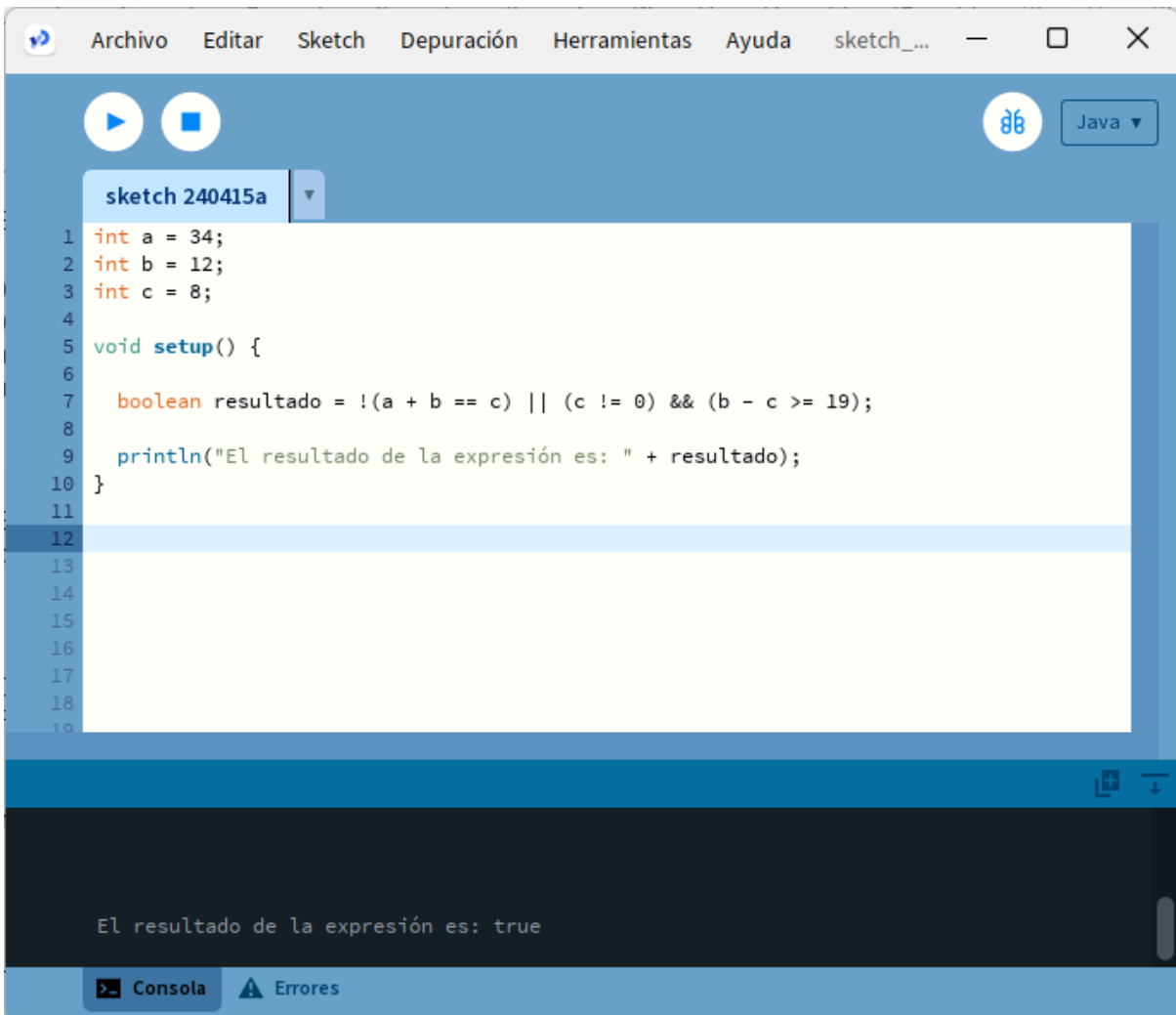
Ejercicio 11: Para a=34, b=12,c=8, evaluar el resultado de $!(a+b==c) \parallel (c!=0) \&\& (b-c>=19)$

$!(34 + 12 == 8) \parallel (8 != 0) \&\& (12 - 8 >= 19)$

$!(falso) \parallel (true) \&\& (false)$

$true \parallel false$

true



```

1 int a = 34;
2 int b = 12;
3 int c = 8;
4
5 void setup() {
6
7   boolean resultado = !(a + b == c) || (c != 0) && (b - c >= 19);
8
9   println("El resultado de la expresión es: " + resultado);
10 }
11
12
13
14
15
16
17
18
19

```

El resultado de la expresión es: true

Ejercicio 12: Un problema sencillo. Deberá pedir por teclado al usuario un nombre y posteriormente realizará la presentación en pantalla de un saludo con el nombre indicado.

Fase de análisis

Datos de entrada: nombre = string

Proceso:

¿Quién debe realizar el proceso? Un programa

¿Cuál es el proceso que realiza? Leer los caracteres ingresados por el usuario y mostrarlos junto a un saludo.

Fase de diseño

ENTIDAD QUE RESUELVE EL PROBLEMA: Processing



VARIABLES

nombre = string

NOMBRE DEL ALGORITMO: mostrar_saludo

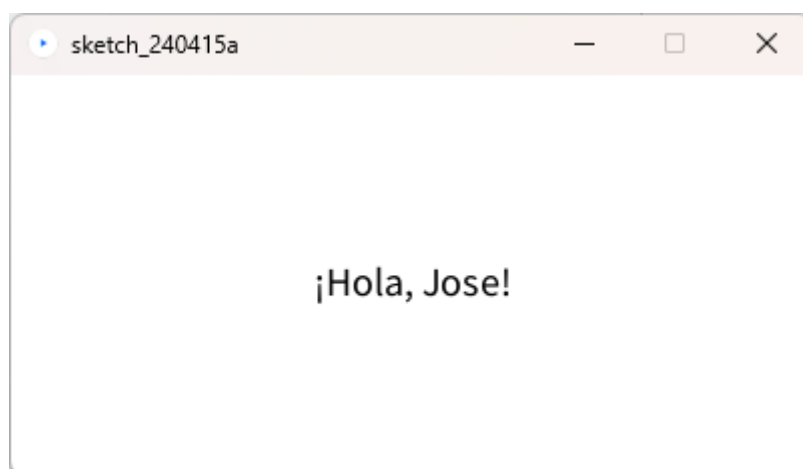
PROCESO DEL ALGORITMO

1. *escribir*("Por favor, ingresa tu nombre:")
2. *leer* nombre
3. nombre ← nombre + key
4. *escribir*("¡Hola, " + nombre + "!", width/2, height/2);

The screenshot shows the Arduino IDE interface with a Java sketch named 'sketch 240415a'. The code defines a String variable 'nombre' and implements the 'setup()' and 'draw()' functions. The 'setup()' function sets the window size to 400x200, aligns text to the center, sets the font size to 20, and prints the prompt 'Por favor, ingresa tu nombre:'. The 'draw()' function sets a white background, a black fill, and displays '¡Hola, ' followed by the current value of 'nombre' and an exclamation mark, centered on the screen. The 'keyTyped()' function appends the pressed key to the 'nombre' string. The console at the bottom shows the prompt 'Por favor, ingresa tu nombre:'.

```
1 String nombre = "";
2
3 void setup() {
4   size(400, 200);
5   textAlign(CENTER, CENTER);
6   textSize(20);
7   println("Por favor, ingresa tu nombre:");
8 }
9
10 void draw() {
11   background(255);
12   fill(#000000);
13   text("¡Hola, " + nombre + "!", width/2, height/2);
14 }
15
16 void keyTyped() {
17   nombre = nombre + key;
18 }
19
```

Por favor, ingresa tu nombre:



Ejercicio 13: Será común resolver problemas utilizando variables. Calcule el perímetro y área de un rectángulo dada su base y su altura.

Fase de análisis

Datos de entrada:

base = real

altura = real

Datos de salida

area = real

perimetro = real

Proceso:

¿Quién debe realizar el proceso? Un programa

¿Cuál es el proceso que realiza? Leer los datos del rectángulo y emplearlos para calcular su área y perímetro

Fase de diseño

ENTIDAD QUE RESUELVE EL PROBLEMA: Processing
VARIABLES base = int altura = int area = int perimetro = int
NOMBRE DEL ALGORITMO: calcular_area_y_perimetro PROCESO DEL ALGORITMO



1. *leer* base
2. *leer* altura
3. $\text{perimetro} \leftarrow 2 * (\text{base} + \text{altura})$
4. $\text{area} \leftarrow \text{base} * \text{altura}$
5. *mostrar* perimetro
6. *mostrar* area

```
1 int base = 10;
2 int altura = 5;
3
4 void setup() {
5
6     int perimetro = 2 * (base + altura);
7     int area = base * altura;
8
9     // Mostrar resultados
10    println("El perímetro del rectángulo es: " + perimetro);
11    println("El área del rectángulo es: " + area);
12 }
13
14
15
16
17
18
19
```

El perímetro del rectángulo es: 30
El área del rectángulo es: 50

Ejercicio 14: Una ayuda importante al momento de resolver problemas con algoritmos es asumir que su gran amigo son las matemáticas. Obtenga la hipotenusa de un triángulo rectángulo conociendo sus catetos

Fase de análisis

Datos de entrada:

catetoA = real



catetoB = real

Datos de salida

hipotenusa = real

Proceso:

¿Quién debe realizar el proceso? Un programa

¿Cuál es el proceso que realiza? Leer los catetos del triángulo y utilizarlos para calcular la hipotenusa del mismo.

Fase de diseño

ENTIDAD QUE RESUELVE EL PROBLEMA: Processing

VARIABLES

catetoA = float

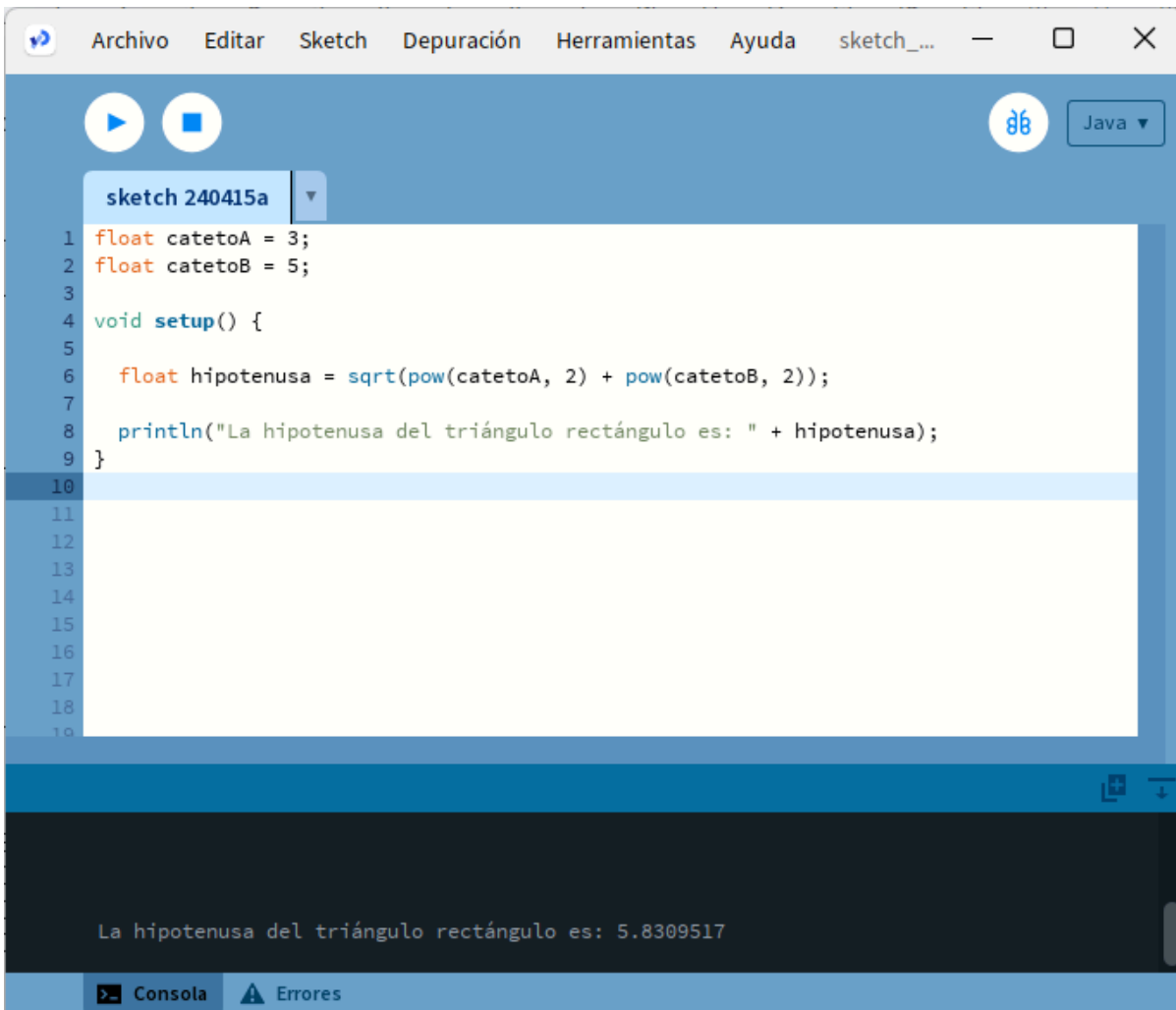
catetoB = float

hipotenusa = float

NOMBRE DEL ALGORITMO: calcular_hipotenusa

PROCESO DEL ALGORITMO

1. *leer* catetoA
2. *leer* catetoB
3. $hipotenusa \leftarrow \sqrt{\text{pow}(\text{catetoA}, 2) + \text{pow}(\text{catetoB}, 2)}$
4. *mostrar* hipotenusa



```
1 float catetoA = 3;
2 float catetoB = 5;
3
4 void setup() {
5
6   float hipotenusa = sqrt(pow(catetoA, 2) + pow(catetoB, 2));
7
8   println("La hipotenusa del triángulo rectángulo es: " + hipotenusa);
9 }
10
11
12
13
14
15
16
17
18
19
```

La hipotenusa del triángulo rectángulo es: 5.8309517

Ejercicio 15: Si viste algo de los apuntes y vídeos, esto debería ser muy fácil de resolver. Dados dos números permita calcular la suma, resta, multiplicación y división de estos. Considere que cada una de estas operaciones es un algoritmo cuando realice el diseño. Obviamente muestre los resultados.

Fase de análisis

Datos de entrada:

numero1 = real

numero2 = real



Datos de salida

suma = real

resta = real

multiplicacion = real

division = real

	<p>FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS TECNICATURA UNIVERSITARIA EN DISEÑO INTEGRAL DE VIDEOJUEGOS FACULTAD DE INGENIERÍA Universidad Nacional de Jujuy PENSAMIENTO COMPUTACIONAL y PROGRAMACIÓN: Problema y Solución – PC y P – Algoritmos – Principio de la P</p>	
---	---	---


Proceso:

¿Quién debe realizar el proceso? Un programa

¿Cuál es el proceso que realiza? Leer ambos números y realizar la suma, la resta, la multiplicación y la división de los mismos.

Fase de diseño

ENTIDAD QUE RESUELVE EL PROBLEMA: Processing
<p>VARIABLES</p> <pre> numero1 = float numero2 = float suma = float resta = float multiplicacion = float division = float </pre>
<p>NOMBRE DEL ALGORITMO: realizar_calculos</p> <p>PROCESO DEL ALGORITMO</p> <ol style="list-style-type: none"> 1. <i>leer</i> numero1 2. <i>leer</i> numero2 3. $\text{suma} \leftarrow \text{numero1} + \text{numero2}$ 4. <i>mostrar</i> suma 5. $\text{resta} \leftarrow \text{numero1} - \text{numero2}$ 6. <i>mostrar</i> resta 7. $\text{multiplicacion} \leftarrow \text{numero1} * \text{numero2}$ 8. <i>mostrar</i> multiplicacion 9. $\text{division} \leftarrow \text{numero1} / \text{numero2}$ 10. <i>mostrar</i> division



```

1 float numero1 = 6;
2 float numero2 = 9;
3
4 void setup() {
5
6   float suma = numero1 + numero2;
7   println("El resultado de la suma es: " + suma);
8
9   float resta = numero1 - numero2;
10  println("El resultado de la resta es: " + resta);
11
12  float multiplicacion = numero1 * numero2;
13  println("El resultado de la multiplicación es: " + multiplicacion);
14
15  float division = numero1 / numero2;
16  println("El resultado de la división es: " + division);
17 }
18

```

El resultado de la suma es: 15.0
El resultado de la resta es: -3.0
El resultado de la multiplicación es: 54.0
El resultado de la división es: 0.6666667

Ejercicio 16: Necesitamos convertir una temperatura Fahrenheit en grados Celsius. Si no conoce la forma en la que se realiza esta conversión, debería investigarlo; para eso sirve la etapa de análisis. Pero como somos buenos, daremos una ayuda: $\text{temperaturaCelsius} = (\text{temperaturaFahrenheit} - 32) / 1.8$

Fase de análisis

Datos de entrada:

temperaturaFahrenheit = real

Datos de salida

temperaturaCelsius = real

Proceso:

¿Quién debe realizar el proceso? Un programa

¿Cuál es el proceso que realiza? Leer el valor de la temperatura en Fahrenheit y convertirla a Celsius implementando la fórmula establecida.



Fase de diseño

ENTIDAD QUE RESUELVE EL PROBLEMA: Processing

VARIABLES

temperaturaFahrenheit = float
temperaturaCelsius = float

NOMBRE DEL ALGORITMO: calcular_temperatura

PROCESO DEL ALGORITMO

1. *leer* temperaturaFahrenheit
2. $\text{temperaturaCelsius} \leftarrow (\text{temperaturaFahrenheit} - 32) / 1.8$
3. *mostrar* temperaturaCelsius



The screenshot shows the Processing IDE interface. The menu bar includes Archivo, Editar, Sketch, Depuración, Herramientas, Ayuda, and sketch_... The toolbar has a play button, a stop button, and a Java language selector. The sketch is named 'sketch 240416a'. The code in the editor is as follows:

```
1 float temperaturaFahrenheit = 86;
2
3 void setup() {
4
5     float temperaturaCelsius = (temperaturaFahrenheit - 32) / 1.8;
6
7     println("Temperatura en Fahrenheit: " + temperaturaFahrenheit);
8     println("Temperatura en Celsius: " + temperaturaCelsius);
9 }
10
11
12
13
14
15
16
17
18
19
```

The console at the bottom shows the output of the program:

```
Temperatura en Fahrenheit: 86.0
Temperatura en Celsius: 30.0
```

The console tabs are labeled 'Consola' and 'Errores'.

	<p>FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS TECNICATURA UNIVERSITARIA EN DISEÑO INTEGRAL DE VIDEOJUEGOS FACULTAD DE INGENIERÍA Universidad Nacional de Jujuy PENSAMIENTO COMPUTACIONAL y PROGRAMACIÓN: Problema y Solución – PC y P – Algoritmos – Principio de la P</p>	
---	---	---

Ejercicio 17: Si queremos representar personajes o power ups (premios) en la pantalla debemos primero ubicarlos en alguna posición dentro de la pantalla. Imagine que está en un juego donde un power up desaparece porque el personaje se acerca a una distancia de x unidades, sin importar por donde se acerque. Por tanto, para que desaparezca, en primer lugar, hay que determinar esa distancia. La forma de representar la posición de un objeto en la pantalla es a través de las coordenadas de un punto. Suponga que la posición de Link está representada por la coordenada (x_1, y_1) , mientras que las de la caja de tesoro se halla en la posición (x_2, y_2) . Si se observa con detenimiento se observa la conformación de un triángulo rectángulo, por lo que es posible aplicar Pitágoras para obtener la distancia. Para esto debe calcular el tamaño de los catetos y luego aplicar el teorema. Halle la distancia entre ambos objetos. Cuando programe, represente a Link con un Círculo, y al tesoro con un cuadrado. Además, mueva a Link mediante el mouse.

Fase de análisis

Datos de entrada:

x_1 = real

y_1 = real

x_2 = real

y_2 = real

Datos de salida

distancia = real

Proceso:

¿Quién debe realizar el proceso? Un programa

¿Cuál es el proceso que realiza? Aplicar el teorema de pitágoras con el fin de hallar la distancia entre ambos objetos.

Fase de diseño

ENTIDAD QUE RESUELVE EL PROBLEMA: Processing
<p>VARIABLES</p> <p>x_1 = real y_1 = real x_2 = real y_2 = real distancia = real</p>
<p>NOMBRE DEL ALGORITMO: calcular_distancia</p> <p>PROCESO DEL ALGORITMO</p>



FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS
TECNICATURA UNIVERSITARIA EN DISEÑO INTEGRAL DE VIDEOJUEGOS

FACULTAD DE INGENIERÍA

Universidad Nacional de Jujuy

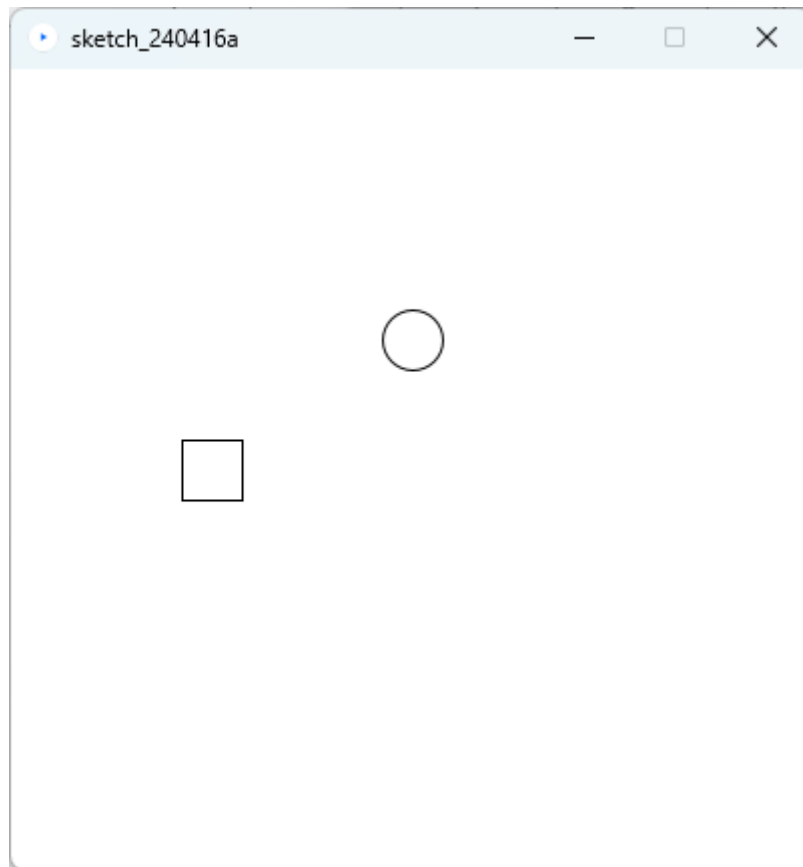
PENSAMIENTO COMPUTACIONAL y PROGRAMACIÓN:

Problema y Solución – PC y P – Algoritmos – Principio de la P



1. *leer* x1
2. *leer* y1
3. *leer* x2
4. *leer* y2
5. $\text{distancia} \leftarrow \sqrt{\text{pow}(x2 - x1, 2) + \text{pow}(y2 - y1, 2)}$;
6. *escribir* ("Distancia entre Link y la caja de tesoro: " + distancia)





Ejercicio 18: Desarrolle el análisis y diseño de un algoritmo que permita obtener las raíces de una ecuación de segundo grado. Además, utilice la estructura según para el análisis de la discriminante de la ecuación cuadrática. Obviamente codifique en Processing.

Fase de análisis

Datos de entrada:

a = real

b = real

c = real

Datos de salida

discriminante = real

x1 = real

x2 = real

x = real

Proceso:

¿Quién debe realizar el proceso? Un programa

¿Cuál es el proceso que realiza? Leer los coeficientes de la ecuación para luego evaluar la discriminante y encontrar sus raíces.



Fase de diseño

ENTIDAD QUE RESUELVE EL PROBLEMA: Processing

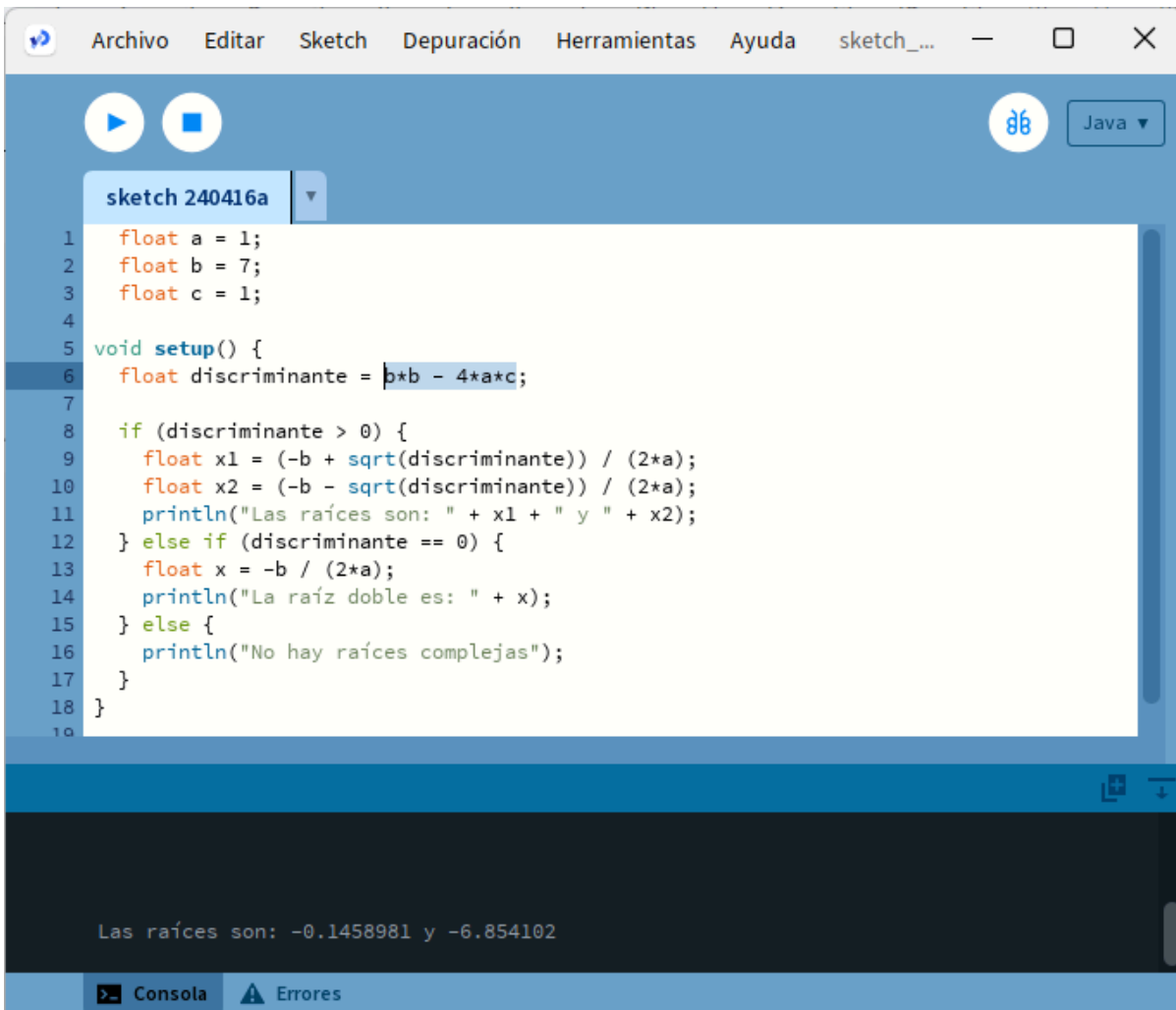
VARIABLES

a = real
b = real
c = real
discriminante = real
x1 = real
x2 = real
x = real

NOMBRE DEL ALGORITMO: calcular_raices

PROCESO DEL ALGORITMO

1. *leer* a
2. *leer* b
3. *leer* c
4. $\text{discriminante} \leftarrow b*b - 4*a*c$
5. **si** (discriminante > 0)
6. $x1 \leftarrow (-b + \text{sqrt}(\text{discriminante})) / (2*a)$
7. $x2 \leftarrow (-b - \text{sqrt}(\text{discriminante})) / (2*a)$
8. *escribir* ("Las raíces son: " + x1 + " y " + x2)
9. **si_no si** (discriminante == 0)
10. $x \leftarrow -b / (2*a)$
11. *escribir* ("La raíz doble es: " + x)
12. **si_no**
13. *escribir* ("No hay raíces complejas")



```

1  float a = 1;
2  float b = 7;
3  float c = 1;
4
5  void setup() {
6    float discriminante = b*b - 4*a*c;
7
8    if (discriminante > 0) {
9      float x1 = (-b + sqrt(discriminante)) / (2*a);
10     float x2 = (-b - sqrt(discriminante)) / (2*a);
11     println("Las raíces son: " + x1 + " y " + x2);
12   } else if (discriminante == 0) {
13     float x = -b / (2*a);
14     println("La raíz doble es: " + x);
15   } else {
16     println("No hay raíces complejas");
17   }
18 }

```

Las raíces son: -0.1458981 y -6.854102

Ejercicio 19: Declare las variables necesarias para dibujar una línea que se dibuja desde las coordenadas iniciales del lienzo y se extiende por todo el ancho. Sobre el punto medio de la línea y a una distancia de 40px (en sentido vertical desde la línea) dibuje una elipse que tenga como ancho 80px y de alto 80px. Dentro de la función draw(), actualice las variables necesarias para que la línea desde su inicio se mueva en dirección hacia abajo arrastrando la elipse. Mantenga en cero el valor para background(). Cuando la línea supere la posición de la altura del lienzo, debe invertir su sentido, es decir dirigirse hacia arriba arrastrando la elipse. Cuando la línea alcance nuevamente el valor 0 para su posición en y, el desplazamiento debe ser hacia abajo y así sucesivamente. El lienzo debería verse como en las siguientes figuras

Fase de análisis

Datos de entrada:
posY = real



direccion = real
radio = real

Proceso:

- ¿Quién debe realizar el proceso? Un programa
- ¿Cuál es el proceso que realiza? Utilizar los valores ingresados para calcular y alterar la posición y la dirección tanto de la línea como del círculo.

Fase de diseño

ENTIDAD QUE RESUELVE EL PROBLEMA: Processing

VARIABLES

posY = real
direccion = real
radio = real

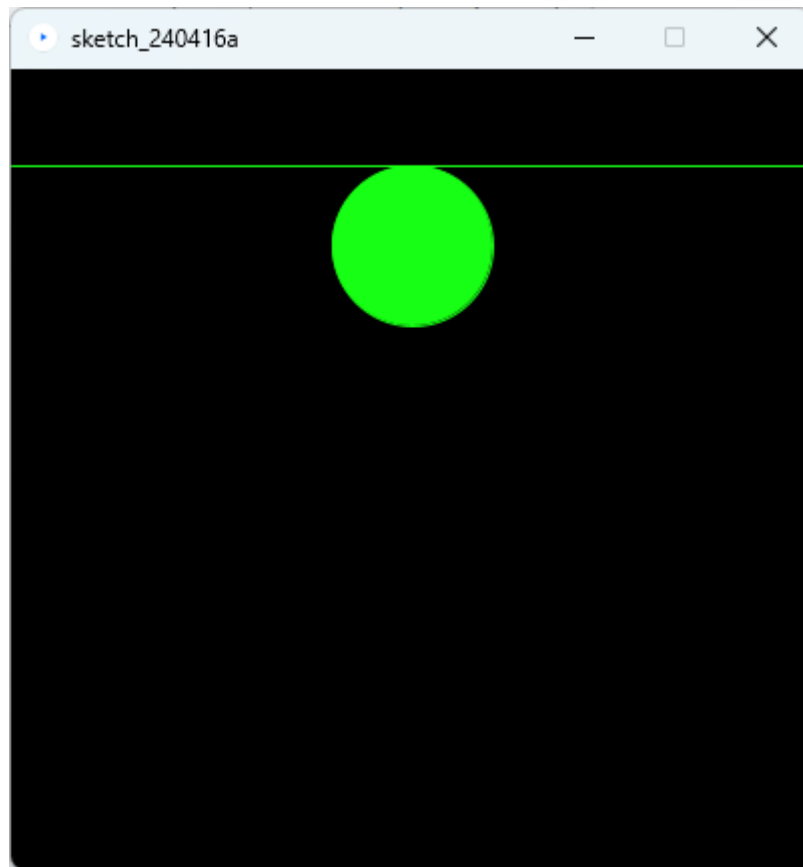
NOMBRE DEL ALGORITMO: calcular_posiciones

PROCESO DEL ALGORITMO

1. leer posY
2. leer direccion
3. leer radio
4. $\text{posY} \leftarrow \text{posY} + \text{direccion}$
5. **si** ($\text{posY} \leq 0 \parallel \text{posY} \geq \text{height}$)
6. $\text{direccion} \leftarrow (-\text{direccion})$
7. $\text{radio} \leftarrow (\text{radio} * -1)$



```
1 int posY;
2 int direccion = 1;
3 int radio = 40;
4
5 void setup() {
6   size(400, 400);
7   posY = 0;
8 }
9
10 void draw() {
11   background(#000000);
12
13   fill(#17FF15);
14   stroke(#17FF15);
15   line(0, posY, width, posY);
16   ellipse(width/2, posY+radio, radio*2, radio*2);
17
18   posY = posY + direccion;
19
20   if (posY <= 0 || posY >= height) {
21     direccion = -direccion;
22     radio = radio * -1;
23   }
24 }
25
```

Ejercicio 20: Dibuje en toda la extensión del lienzo de (440, 420) rectángulos de idénticas medidas (40 ancho y 20 de alto) y que mantengan una distancia de 20 pixeles entre ellos tanto horizontal como verticalmente. Utilice la estructura de control repetitiva for. El lienzo debería verse así:

Fase de análisis

Datos de entrada:

ancho = real

alto = real

distancia = real

Proceso:

¿Quién debe realizar el proceso? Un programa

¿Cuál es el proceso que realiza? Utilizar los valores ingresados para dibujar los rectángulos en el lienzo

Fase de diseño

ENTIDAD QUE RESUELVE EL PROBLEMA: Processing



VARIABLES

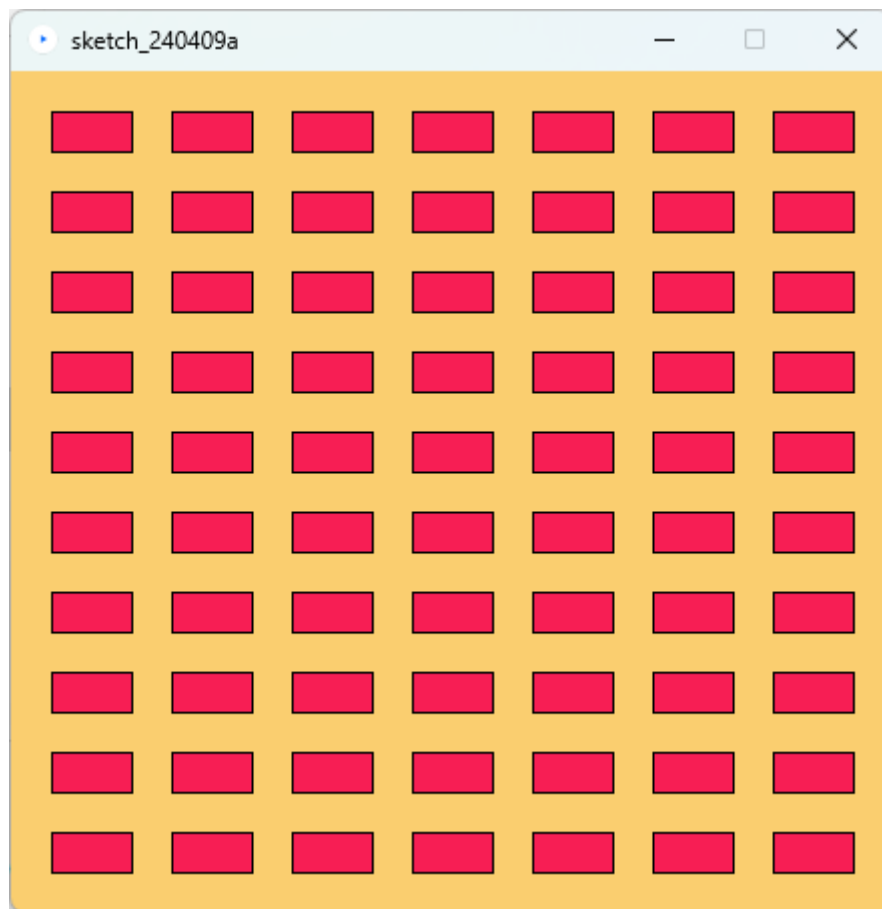
```
ancho = int  
alto = int  
distancia = int
```

NOMBRE DEL ALGORITMO: dibujar_rectangulos

PROCESO DEL ALGORITMO

8. leer ancho
9. leer alto
10. leer distancia
11. **para** (int y = 20; y <= height - alto; y += (alto + distancia))
12. **para** (int x = 20; x <= width - ancho; x += (ancho + distancia))
13. dibujar rect(x, y, ancho, alto)

```
Archivo  Editar  Sketch  Depuración  Herramientas  Ayuda  ejercicio ...  Java ▼  
sketch 240409a ▼  
1 int ancho = 40;  
2 int alto = 20;  
3 int distancia = 20;  
4  
5 public void setup(){  
6   size(440, 420);  
7   background(#FACE6F);  
8 }  
9  
10 public void draw(){  
11   fill(#F71E54);  
12   for (int y = 20; y <= height - alto; y += (alto + distancia)) {  
13     for (int x = 20; x <= width - ancho; x += (ancho + distancia)) {  
14       rect(x, y, ancho, alto);  
15     }  
16   }  
17 }  
18  
19  
20  
You are running Processing revision 01292, the latest build is 01293.  
Consola  Errores
```



Ejercicio 21: Utilizando la estructura de control repetitiva while() dibuje la siguiente imagen utilizando líneas que forman escalones y sobre cada borde de escalón se dibuje un punto de color rojo El tamaño del lienzo es size(500,500). La estructura while() se ejecuta dentro de la función setup(). La condición es que solo se dibuje dentro del lienzo. Utilice variables que puedan ayudar a la construcción del dibujo, por ej: x, y, anchoEscalon, altoEscalon, etc

Fase de análisis

Datos de entrada:

distancia = real
x = real
y = real

Proceso:

¿Quién debe realizar el proceso? Un programa
¿Cuál es el proceso que realiza? Utilizar los valores correspondientes para dibujar los escalones junto a los puntos hasta llenar el lienzo.

Fase de diseño



ENTIDAD QUE RESUELVE EL PROBLEMA: Processing

VARIABLES

distancia = intl

x = float

y = float

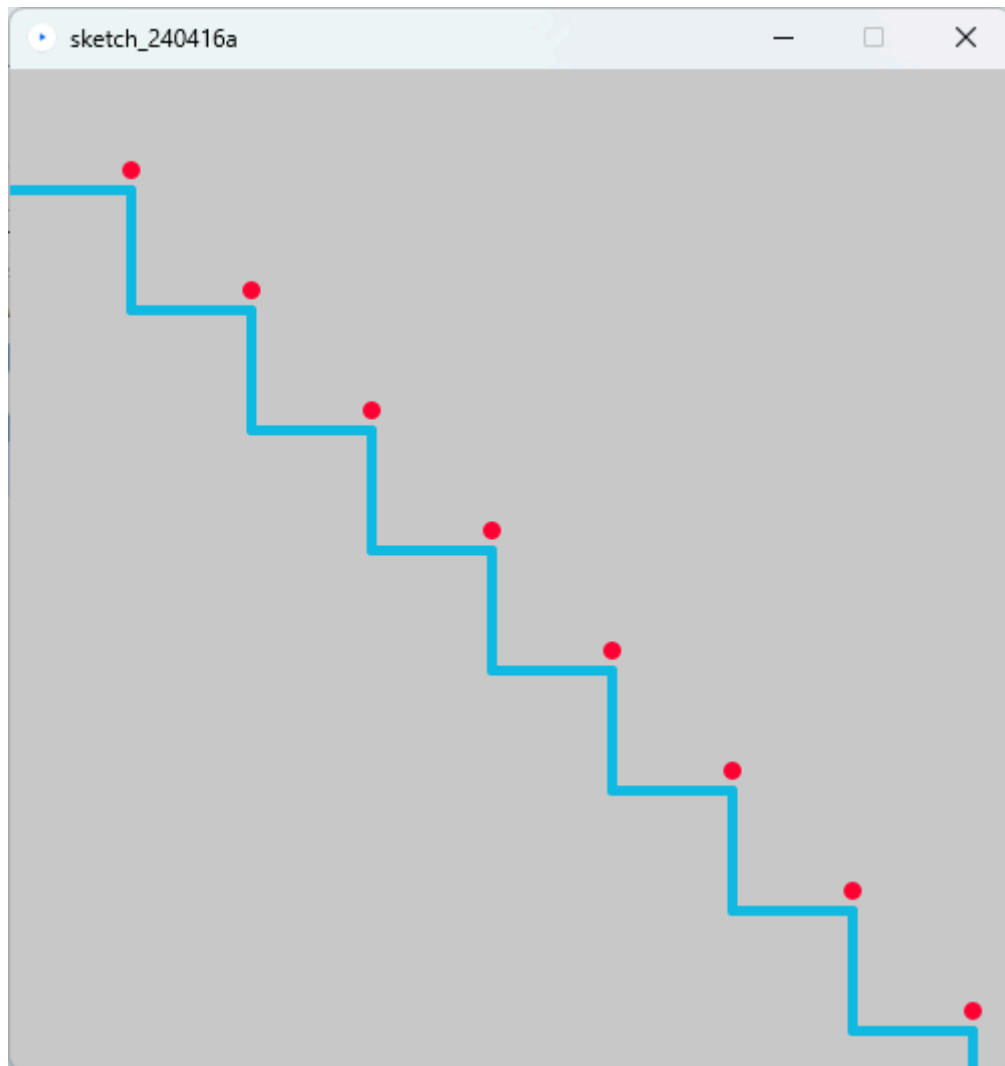
NOMBRE DEL ALGORITMO: dibujar_escalones

PROCESO DEL ALGORITMO

1. *leer* x
2. *leer* y
3. *leer* distancia
4. **mientras** (y <= height)
5. *dibujar* line(x, y, x + distancia, y)
6. *dibujar* line(x + distancia, y, x + distancia, y + distancia)
7. *dibujar* point(x + distancia, y - 10)
8. $x \leftarrow x + distancia$
9. $y \leftarrow y + distancia$



```
1 void setup() {
2   size(500, 500);
3   int distancia = 60;
4   float x = 0;
5   float y = distancia;
6
7   while (y <= height) {
8     stroke(#10BCE5);
9     strokeWeight(5);
10    line(x, y, x + distancia, y);
11    line(x + distancia, y, x + distancia, y + distancia);
12
13    stroke(#FF0033);
14    strokeWeight(9);
15    point(x + distancia, y - 10);
16
17    x += distancia;
18    y += distancia;
19  }
20 }
21
```



Ejercicio 22: Declare las variables necesarias para dibujar una línea que se dibuja desde las coordenadas iniciales del lienzo y se extiende por todo el ancho. Sobre el punto medio de la línea y a una distancia de 40px (en sentido vertical desde la línea) dibuje una elipse que tenga como ancho 80px y de alto 80px. Dentro de la función draw(), actualice las variables necesarias para que la línea desde su inicio se mueva en dirección hacia abajo arrastrando la elipse. Mantenga en cero el valor para background(). Cuando la línea supere la posición de la altura del lienzo, debe invertir su sentido, es decir dirigirse hacia arriba arrastrando la elipse. Cuando la línea alcance nuevamente el valor 0 para su posición en y, el desplazamiento debe ser hacia abajo y así sucesivamente. El lienzo debería verse como en las siguientes figuras

Fase de análisis

Datos de entrada:
lineaY = real



contador = real
circuloX = real

Proceso:

¿Quién debe realizar el proceso? Un programa
¿Cuál es el proceso que realiza? Utilizar los valores ingresados para dibujar las líneas y, posteriormente, determinar cuáles tendrán círculos de colores aleatorios encima.

Fase de diseño

ENTIDAD QUE RESUELVE EL PROBLEMA: Processing

VARIABLES

lineaY = real
contador = real
circuloX = real

NOMBRE DEL ALGORITMO: calcular_posiciones

PROCESO DEL ALGORITMO

1. leer lineaY
2. leer contador
3. **hacer**
4. dibujar line(0, lineaY, width, lineaY)
5. circuloX \leftarrow 30
6. **mientras** (circuloX < width - 15)
7. dibujar ellipse(circuloX, lineaY*2-120, 40, 40)
8. circuloX \leftarrow circuloX + 60
9. **mientras** (contador < 6)
10. lineaY \leftarrow lineaY + 100
11. contador \leftarrow contador + 1



```
1 void setup() {
2   size(600, 600);
3   int lineaY = 100;
4   int contador = 0;
5
6   do {
7     stroke(#3856FA);
8     strokeWeight(4);
9     line(0, lineaY, width, lineaY);
10
11    int circuloX = 30;
12
13    while (circuloX < width - 15) {
14      stroke(#000000);
15      strokeWeight(4);
16      fill(random(255), random(255), random(255));
17      ellipse(circuloX, lineaY*2-120, 40, 40);
18      circuloX = circuloX + 60;
19    }
20    lineaY = lineaY + 100;
21    contador = contador + 1;
22  } while (contador < 6);
23 }
24
25
```




FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS
TECNICATURA UNIVERSITARIA EN DISEÑO INTEGRAL DE VIDEOJUEGOS

FACULTAD DE INGENIERÍA

Universidad Nacional de Jujuy

PENSAMIENTO COMPUTACIONAL y PROGRAMACIÓN:

Problema y Solución – PC y P – Algoritmos – Principio de la P

