

# TRABAJO PRÁCTICO N° 1

F P O O  
( T U D I V J )

UNJU-Facultad de Ingeniería

Álvaro Gonzalo Ramos Aquino  
45764697  
TUV000746  
Año 2024

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

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

$$(2*3/6)+(3/1/25/4*2)$$

$$(6/6)+(3/25/4*2)$$

$$1+(0.12/4*2)$$

$$1+(0.03*2)$$

$$1+0.06$$

$$1.06$$

```
1 float resultado = (4/2*3/6)+(6./2/1/pow(5,2)/4*2);  
2 |  
3 println(resultado);
```

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 = 1, b = 2, c = 3, d = 4, X = 5, y = 6, x = 7

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

$$4-(4*c)$$

$$4-12$$

$$-8$$

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

$$(3*625)-((5*125)+(5*12))-17$$

$$1875-(625+60)-17$$

$$1875-685-17$$

$$1190-17$$

$$1173$$

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

$$6/7$$

$$0.85$$

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

$$(49+36)^{0.5}$$

$$85^{0.5}$$

$$9.21$$

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

b)  $3.X^4 - 5.X^3 + X.12 - 17$

c)  $(b + d) \div (c + 4)$

d)  $\sqrt{x^2+y^2}$

```

1 int a=1,b=2,c=3,d=4,X=5,y=6,x=7;
2
3 float resultadoA = pow(b,2)-(4*a*c);
4 float resultadoB = (3*pow(X,4))-(5*pow(X,3)+X*12)-17;
5 float resultadoC = (b+d)/(c+4.);
6 float resultadoD = pow((pow(x,2)+pow(y,2)),(1/2.));
7
8 println(resultadoA);
9 println(resultadoB);
10 println(resultadoC);
11 println(resultadoD);

```

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)  $(B \cdot A) - (B^{(2/4 \cdot C)})$   
 $20 - (B^{(0.5 \cdot C)})$   
 $20 - (B^{0.5})$   
 $20 - 2.23$   
 $17,76$
- b)  $(A \cdot B) / (3^2)$   
 $20/9$   
 $2.22$
- c)  $((((B+C)/2 \cdot A) + 10) \cdot 3 \cdot B) - 6$   
 $((6/2 \cdot A) + 10) \cdot 3 \cdot B - 6$   
 $((3 \cdot A) + 10) \cdot 3 \cdot B - 6$   
 $((12 + 10) \cdot 3 \cdot B) - 6$   
 $(22 \cdot 3 \cdot B) - 6$   
 $(66 \cdot B) - 6$   
 $330 - 6$   
 $324$

```

1 int A=4,B=5,C=1;
2
3 float resultadoA = (B*A)-pow(B,(2./4.*C));
4 float resultadoB = (A*B)/pow(3,2.);
5 float resultadoC = (((B+C)/2*A)+10)*3*B-6;
6
7 println(resultadoA);
8 println(resultadoB);
9 println(resultadoC);

```

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

R1 = y+z

R1 = 5

R2 = x >= R1

R2 = (3 >= 5)

R2= false

```

1 int x=3, y=4, z=1;
2
3 int R1 = y+z;
4 boolean R2= (x>=R1);
5
6 println(R2);

```

Ejercicio 7: Para contador1=3, contador2=4, evaluar el resultado de

R1 = ++contador1

R1= 4

R2 = contador1 < contador2

R2= 4<4

R2=false

```

1 int contador1=3, contador2=4 ;
2
3 int R1 = ++contador1;
4 boolean R2= (contador1<contador2);
5
6 println(R2);

```

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

a+b-1 < x\*y

31-1<6

30<6

False

```

1 int a=31, b=-1, x=3, y=2;
2
3 boolean ejercicio=(a+b-1<x*y);
4 println(ejercicio);

```

Ejercicio 9: Para x=6, y=8, evaluar el resultado de

!(x<5)&& !(y>=7)

!(false) AND !(true)

(true) AND (false)

False

```

1 int x=6, y=8;
2
3 boolean ejercicio=!(x<5)&& !(y>=7));
4 println(ejercicio);

```

Ejercicio 10: Para i=22,j=3, evaluar el resultado de

!((i>4) || !(j<=6))

!(true OR !(true))

!(true OR false)

!(true)

False

```

1 int i=22, j=3;
2
3 boolean ejercicio=!((i>4) || !(j<=6));
4 println(ejercicio);

```

Ejercicio 11: Para a=34, b=12,c=8, evaluar el resultado de

!(a+b==c) || (c!=0)&&(b-c>=19)

!(46==c) OR (true) AND (4>=19)

!(false) OR (true) AND (false)

true OR (true AND false)

true OR false

true

```

1 int a=34, b=12, c=8;
2
3 boolean ejercicio=! (a+b==c) || (c!=0)&&(b-c>=19);
4 println(ejercicio);

```

ejercicio 12:

Definicion del problema: mostrar un saludo con el nombre del usuario

Analisis:

- Datos de Entrada:  
-nombre del usuario: string
- Datos de Salida:  
-saludo con nombre del usuario: string
- Proceso:

¿Quién debe realizar el proceso?: el usuario con el programa

¿Cuál es el proceso que realiza?:

Recibir el nombre del usuario para realizar un saludo con su nombre

**Diseño:**

Entidad: usuario
Variables: <ul style="list-style-type: none"><li>- nombreUsuario: string</li><li>- saludo: string</li></ul>
Nombre de Algoritmo: saludar_usuario
Proceso del Algoritmo: Inicio Importar JOptionPane del paquete Java Leer nombre Mostrar saludo en ("¡Hola"+nombre+"!") fin

```
1 import javax.swing.JOptionPane;
2
3 String nombreUsuario = JOptionPane.showInputDialog("Por favor, ingresa tu nombre:");
4
5 println("¡Hola "+nombreUsuario+"!");
```

**ejercicio 13:**

Definicion del problema: calcular el primetro y área de un rectangulo

**Analisis:**

- Datos de Entrada:
  - base: int
  - altura: int
- Datos de Salida:
  - perimetro: int
  - area: int
- Proceso:

¿Quién debe realizar el proceso?: el programa

¿Cuál es el proceso que realiza?:

Calcular el perímetro y área de un rectángulo con la base y altura dadas

**Diseño:**

Entidad: programa
Variables: <ul style="list-style-type: none"><li>- base: int</li><li>- altura: int</li><li>- perímetro: int</li></ul>

- area: int
<p>Nombre de Algoritmo: calcular_rectangulo</p> <p>Proceso del Algoritmo:</p> <p>Inicio</p> <p>importar JOptionPane del paquete Java</p> <p>Leer base</p> <p>Leer altura</p> <p>Mostrar perímetro del rectángulo igual a <math>(base*2+altura*2)</math></p> <p>Mostrar area del rectángulo igual a <math>(base*altura)</math></p> <p>fin</p>

```

1 import javax.swing.JOptionPane;
2
3 int base, altura;
4
5 base = int(JOptionPane.showInputDialog("Por favor, ingresa la base del rectangulo:"));
6 altura = int(JOptionPane.showInputDialog("Por favor, ingresa la altura del rectangulo:"));
7
8 println("perimetro del rectangulo="+ (base*2+altura*2));
9 println("area del rectangulo="+ (base*altura));

```

ejercicio 14:

Definicion del problema: obtener hipotenusa de un triangulo

Analisis:

- Datos de Entrada:
  - cateto1: int
  - cateto2: int
- Datos de Salida:
  - hipotenusa: float
- Proceso:
  - ¿Quién debe realizar el proceso?: el programa
  - ¿Cuál es el proceso que realiza?:
    - Calcular la hipotenusa de un triangulo con sus dos catetos dados

**Diseño:**

Entidad: programa
<p>Variables:</p> <ul style="list-style-type: none"> <li>- cateto1: int</li> <li>- cateto2: int</li> <li>- hipotenusa: float</li> </ul>
<p>Nombre de Algoritmo: calcular_hipotenusa</p> <p>Proceso del Algoritmo:</p> <p>Inicio</p> <p>importar JOptionPane del paquete Java</p>

```

Leer cateto1
Leer cateto2
hipotenusa ← raíz de (cateto1^2 + cateto2^2)
Mostrar hipotenusa del triangulo
fin

```

```

1 import javax.swing.JOptionPane;
2
3 int cateto1, cateto2;
4
5 cateto1 = int(JOptionPane.showInputDialog("Por favor, ingresa el cateto 1 del triangulo:"));
6 cateto2 = int(JOptionPane.showInputDialog("Por favor, ingresa el cateto 2 del triangulo:"));
7
8 float hipotenusa = sqrt(pow(cateto1,2.)+pow(cateto2,2.));
9
10 println("hipotenusa del triangulo="+hipotenusa);

```

ejercicio 15:

Definicion del problema: calcular la suma, resta, multiplicacion y división de dos números dados

Analisis:

- Datos de Entrada:
  - numeroA: int
  - numeroB: int
- Datos de Salida:
  - resultadoSuma: float
  - resultafoResta: float
  - resultadoMulti: float
  - resultadoDiv: float
- Proceso:
  - ¿Quién debe realizar el proceso?: el programa

¿Cuál es el proceso que realiza?:

Calcular la suma, resta, multiplicación y división de dos números dados

**Diseño:**

Entidad: programa
Variables: <ul style="list-style-type: none"> <li>- numeroA: int</li> <li>- numeroB: int</li> <li>- resultadoSuma: float</li> <li>- resultafoResta: float</li> <li>- resultadoMulti: float</li> <li>- resultadoDiv: float</li> </ul>
Nombre de Algoritmo: calcular_resultado  Proceso del Algoritmo: Inicio importar JOptionPane del paquete Java



```

Leer numeroA
Leer numeroB
resultadoSuma ← (numeroA+numeroB)
resultadoResta ← (numeroA-numeroB)
resultadoMulti ← (numeroA*numeroB)
resultadoDiv ← (numeroA/numeroB)
Mostrar resultadoSuma
Mostrar resultadoResta
Mostrar resultadoMulti
Mostrar resultadoDiv
fin

```

```

1 import javax.swing.JOptionPane;
2
3 int numeroA, numeroB;
4
5 numeroA = int(JOptionPane.showInputDialog("Por favor, ingresa el primer numero:"));
6 numeroB = int(JOptionPane.showInputDialog("Por favor, ingresa el segundo numero:"));
7
8 float resultadoSuma = (numeroA+numeroB);
9 float resultadoResta = (numeroA-numeroB);
10 float resultadoMulti = (numeroA*numeroB);
11 float resultadoDiv = float(numeroA)/numeroB;
12
13 println("resultado de suma="+resultadoSuma);
14 println("resultado de resta="+resultadoResta);
15 println("resultado de multiplicacion="+resultadoMulti);
16 println("resultado de division="+resultadoDiv);

```

ejercicio 16:

Definicion del problema: convertir temperatura Fahrenheit en grados Celsius

Analisis:

- Datos de Entrada:  
-gradoFah: int
- Datos de Salida:  
-gradoCel: float
- Proceso:  
¿Quién debe realizar el proceso?: el programa

¿Cuál es el proceso que realiza?:

Convertir los grados Fahrenheit dados en grados Celsius

**Diseño:**

Entidad: programa
Variables: <ul style="list-style-type: none"> <li>- gradoFah: int</li> <li>- gradoCel: float</li> </ul>

Nombre de Algoritmo: transformar\_celsius

Proceso del Algoritmo:

inicio  
importar JOptionPane del paquete Java  
*Leer* gradoFah  
 $\text{gradoCel} \leftarrow (\text{gradoFah} - 32) / 1.8$   
*Mostrar* gradoCel  
fin

```
1 import javax.swing.JOptionPane;  
2  
3 int gradoFah;  
4  
5 gradoFah = int(JOptionPane.showInputDialog("Por favor, ingresa la temperatura en Fahrenheit:"));  
6  
7 float gradoCel = (gradoFah-32.)/1.8;  
8  
9 println("temperatura en Celsius="+gradoCel);
```

ejercicio 17:

Definicion del problema: calcular la distancia entre Link y la caja

Analisis:

- Datos de Entrada:
  - ancho, alto de Lienzo: int
  - coordenadasLink: coordenadas cartesianas
  - coordenadasCaja: coordenadas cartesianas
  - ancho, alto de Link: int
  - ancho, alto de Caja: int
  - catetoA, catetoO: coordenadas cartesianas
  - colorLink: color
  - colorCaja: color
- Datos de Salida:
  - distanciaLinkCaja: float
- Proceso:
  - ¿Quién debe realizar el proceso?: el programa
  - ¿Cuál es el proceso que realiza?:
    - Colocar en un lienzo un cuadrado estatico y un circulo controlado por el mouse que dependiendo la posición de este se calcule la distancia entre ambos

Diseño:

Entidad: programa

Variables:

- anchoLienzo, altoLienzo: int
- coordenadasLink: coordenadas cartesianas
- coordenadasCaja: coordenadas cartesianas

- anchoLink, altoLink: int
- anchoLink, altoCaja: int
- catetoA, catetoO: coordenadas cartesianas
- colorLink: color
- colorCaja: color
- distanciaLinkCaja: float

Nombre de Algoritmo: calcular\_distanciaLinkCaja

Proceso del Algoritmo:

```

inicio
anchoLienzo ← 400
altoLienzo ← 400
coordenadaXCaja ← 200
coordenadaYCaja ← 300
altoLink ← 20
anchoLink ← 20
altoCaja ← 20
anchoCaja ← 20
catetoA ← xCaja - xLink
catetoB ← yCaja - yLink
distanciaLinkCaja ← raíz de (catetoA^2 + catetoB^2)
Leer coordenadasLink
Mostrar distanciaLinkCaja
  
```

```

1 float yLink, xLink;
2 float yCaja, xCaja;
3
4 void setup() {
5   size(400, 400);
6
7   xLink= 100;
8   yLink= 100;
9
10  xCaja=200;
11  yCaja=300;
12 }
13
14 void draw() {
15   background(255);
16
17   float catetoA = xCaja - xLink;
18   float catetoO = yCaja - yLink;
19
20   float distancia = sqrt(pow(catetoA, 2)+pow(catetoO, 2));
21
22   fill(255, 0, 0);
23   ellipse(xLink, yLink, 20, 20);
24
25   fill(0, 0, 255);
26   rectMode(CENTER);
27   rect(xCaja, yCaja, 20, 20);
  
```

```

22 fill(255, 0, 0);
23 ellipse(xLink, yLink, 20, 20);
24
25 fill(0, 0, 255);
26 rectMode(CENTER);
27 rect(xCaja, yCaja, 20, 20);
28
29 println("La distancia entre Link y la caja es de "+distancia);
30 }
31 void mouseMoved() {
32     xLink = mouseX;
33     yLink = mouseY;
34 }

```

ejercicio 18:

Definicion del problema: crear un algoritmo que resuelva raíces de ecuaciones de segundo grado y analize su discriminante

Analisis:

- Datos de Entrada:
  - numeroA: float
  - numeroB: float
  - numeroC: float
- Datos de Salida:
  - resultadoEcu: float
  - discriminante: float
  - x1, x2, x: float
- Proceso:
  - ¿Quién debe realizar el proceso?: el programa

¿Cuál es el proceso que realiza?:

Calcular la ecuación de segundo grado dada y analizar su discriminante

**Diseño:**

Entidad: programa
Variables: <ul style="list-style-type: none"> <li>- numeroA: float</li> <li>- numeroB: float</li> <li>- numeroC: float</li> <li>- resultadoEcu: float</li> <li>- discriminante: float</li> <li>- x1, x2, x: float</li> </ul>
Nombre de Algoritmo: calcular_ecuacion  Proceso del Algoritmo: <ul style="list-style-type: none"> <li>Inicio</li> <li>importar JOptionPane del paquete Java</li> <li>Leer numeroA</li> </ul>

```

Leer numeroB
Leer numeroC
discriminante ← numeroB^2 - 4*numeroA*numeroC
si la discriminante es mayor a cero se realizara
x1 ← (-numeroB+raíz del(discriminante))/(2*a)
x2 ← (-numeroB-raíz del(discriminante))/(2*a)
Mostrar las raíces reales y distintas x1, x2

si la discriminante es igual a cero se realizara
x ← -numeroB/(2*a)
Mostrar la raíz real e igual x

si no cumple ninguna de las condiciones anteriores se realizara
Mostrar las raíces son complejas
fin

```

```

1 import javax.swing.JOptionPane;
2 float a, b, c;
3
4 a= float(JOptionPane.showInputDialog("Escribir la variable a"));
5 b= float(JOptionPane.showInputDialog("Escribir la variable b"));
6 c= float(JOptionPane.showInputDialog("Escribir la variable c"));
7
8 float discriminante = pow(b, 2) - 4*a*c;
9 println("la discriminante es igual:" + discriminante);
10
11 if (discriminante > 0){
12     float x1= (-b + sqrt(discriminante)) / (2 * a);
13     float x2= (-b - sqrt(discriminante)) / (2 * a);
14     print("Las raices son reales y distintas: " + "x1="+ x1 + " x2="+ x2);
15 } else if (discriminante == 0){
16     float x= -b/(2 * a);
17     print("las raices son reales e iguales: " + "x= " + x);
18 } else {
19     print("las raices son complejas");
20 }

```

ejercicio 19:

Definicion del problema: dibujar una línea que toque la parte superior de un círculo y hacer que los dos se muevan juntos indefinidamente de arriba a abajo de los bordes del lienzo

Analisis:

- Datos de Entrada:
  - anchoLienzo, altoLienzo: int
  - yLlinea, yElipse, velocidad: float
  - direccionLinea, direcciónElipse: int
- Datos de Salida:
  - movimiento de círculo con linea
- Proceso:
  - ¿Quién debe realizar el proceso?: el programa

¿Cuál es el proceso que realiza?:

En un lienzo muestra una línea y un círculo a los cuales se les otorga cierta velocidad y la capacidad de rebotar entre los límites del lienzo

#### Diseño:

Entidad: programa
Variables: <ul style="list-style-type: none"><li>- anchoLienzo, altoLienzo: int</li><li>- yLinea, yElipse, velocidad: float</li><li>- direccionLinea, direcciónElipse: int</li></ul>
Nombre de Algoritmo: rebotar_circulo_linea
Proceso del Algoritmo:  Inicio velocidad $\leftarrow$ 2 direccionLinea $\leftarrow$ 1 direccionElipse $\leftarrow$ 1 yElipse $\leftarrow$ 240 yLinea $\leftarrow$ altura de elipse/2 coordenada inicio para la elipse respecto a y coordenada inicio para la linea respecto a y velocidad por la que se moverá la línea por el eje y si la línea es igual o mayor a la altura del lienzo rebotara invirtiendo su movimiento a -1 si la elipse es igual o mayor a la altura del lienzo rebotara invirtiendo su movimiento a -1

```
1 float yLinea, yElipse, velocidad = 2;
2 int direccionLinea = 1, direccionElipse = 1;
3
4 void setup() {
5     size(400, 400);
6     yLinea = height / 2;
7     yElipse = 240;
8 }
9 void draw() {
10    background(0);
11    fill(0, 200, 0);
12    ellipse(width/2, yElipse, 80, 80);
13    stroke(255);
14    line(0, yLinea, width, yLinea);
15    yLinea += direccionLinea * velocidad;
16
17    if (yLinea >= height || yLinea <= 0) {
18        direccionLinea *= -1;
19    }
20    yElipse += direccionElipse * velocidad;
21    if (yElipse >= height || yElipse <= 0) {
22        direccionElipse *= -1;
23    }
24 }
```

ejercicio 20:

Definicion del problema: dibujar una serie de rectángulos idénticos en un lienzo

Analisis:

- Datos de Entrada:
  - coordenadasRect: coordenadas cartesianas
  - ancho, alto, distanciaEntreRect: enteros
  - anchoLienzo, altoLienzo: enteros
  - rect\_color: color
- Datos de Salida:
  - rectangulos dibujados
- Proceso:
  - ¿Quién debe realizar el proceso?: el programa
  - ¿Cuál es el proceso que realiza?:  
Dibujar una serie de rectángulos con determinado espaciado y color entre ellos en un lienzo con un determinado tamaño

**Diseño:**

Entidad: programa
Variables: <ul style="list-style-type: none"><li>- coordenadasRect: coordenadas</li><li>- ancho, alto, distanciaEntreRect: enteros</li><li>- anchoLienzo, altoLienzo: enteros</li><li>- rect_color: color</li></ul>
Nombre de Algoritmo: dibujar_rectangulos  Proceso del Algoritmo:  inicio anchoLienzo ← 440 altoLienzo ← 420 distanciaEntreRect ← 20 anchoz ← 40 alto ← 20 color ← color(255, 165, 0) para x ← coordenadasRect.x hasta anchoLienzo con paso (ancho+distanciaEntreRect) hacer para y ← coordenadasRect.y hasta altoLienzo con paso (alto+distanciaEntreRect) hacer rellenar con color los rectangulos dibujar un rectángulo en (x.coordenadasRect.y) con dimensiones ancho y alto

fin para  
fin para  
fin

```
rectangulo
1 PVector coordenadasRect;
2 int ancho, alto, distEntreRect;
3 color rectColor;
4
5 public void setup(){
6     size(440,420);
7     distEntreRect = 20;
8     ancho=40;
9     alto=20;
10    coordenadasRect = new PVector(distEntreRect,distEntreRect);
11    rectColor = color(255, 165, 0);
12 }
13
14 public void draw(){
15     dibujarRectangulos();
16 }
17
18 public void dibujarRectangulos(){
19     for(float x=coordenadasRect.x;x<width;x+= (ancho+distEntreRect)){
20         for(float y=coordenadasRect.y;y<height;y+= (alto+distEntreRect)){
21             fill(rectColor);
22             rect(x,y,ancho,alto);
23         }
24     }
25 }
26
27
```

ejercicio 21:

Definicion del problema: dibujar una serie de escalones donde en cada borde hay un punto rojo

Analisis:

- Datos de Entrada:
  - puntoA
  - puntoB
  - puntoC
  - puntoD
- Datos de Salida:
  - EL dibujo en la línea horizontal
  - El dibujo en la línea vertical
  - El dibujo del punto rojo
- Proceso:
  - ¿Quién debe realizar el proceso?: el programa
  - ¿Cuál es el proceso que realiza?:
    - Dibujar una serie de escalones con un punto rojo al borde de cada escalon dentro de un lienzo

**Diseño:**



Entidad: programa

Variables:

- puntoA
- puntoB
- puntoC
- puntoD
- distLinea: entero

Nombre de Algoritmo: dibujar\_escalera

Proceso del Algoritmo:

inicio

dibujar una linea horizontal entre los puntos A y B, con distancia distLinea

dibujar una linea vertical entre los puntos B y C, con distancia distLinea

dibujar\_circulo

dibujar un punto en la siguiente posicion: x= posicion en x de B, y = posicion en y de B - 10

fin

```
1 PVector puntoA, puntoB, puntoC, puntoD;
2 int distlinea;
3
4 public void setup(){
5     size(500, 500);
6     distlinea= 60;
7     puntoA= new PVector(0, distlinea);
8
9     while(puntoA.y < height){
10         dibujarEscalon();
11         actualizarCoordenadasA();
12     }
13 }
14 public void dibujarEscalon(){
15     stroke(#1AEAFF);
16     strokeWeight(4);
17     puntoB= new PVector(puntoA.x+distlinea,puntoA.y);
18     line(puntoA.x,puntoA.y,puntoB.x,puntoB.y);
19     puntoC= new PVector(puntoB.x,puntoB.y+distlinea);
20     line(puntoB.x,puntoB.y, puntoC.x, puntoC.y);
21     dibujarPunto();
22 }
23 public void dibujarPunto(){
24     stroke(255, 0, 0);
25     strokeWeight(10);
26     puntoD = new PVector(puntoB.x, puntoB.y-10);
27     point(puntoD.x,puntoD.y);
28 }
29
30 public void actualizarCoordenadasA(){
31     puntoA.x= puntoC.x;
32     puntoA.y= puntoC.y;
33 }
```

## ejercicio 22:

Definicion del problema: dibujar una serie de líneas de las cuales algunas tendrán círculos de colores arriba suyo

### Analisis:

- Datos de Entrada:
  - distanciaEntreRect: coordenadas cartesianas
  - alto, ancho: int
  - lineaX, Y: coordenadas cartesianas
  - circuloX, Y: coordenadas cartesianas
  - distanciaCirculo:

- Datos de Salida:
  - lineas con círculos dibujado

- Proceso:
  - ¿Quién debe realizar el proceso?: el programa

¿Cuál es el proceso que realiza?:

Dibujar una serie de de 6 líneas de manera horizontal a través del lienzo, de las cuales solamente en las líneas impares tendrán círculos de colores arriba suyo

### Diseño:

Entidad: programa

#### Variables:

- coordenadasRect: coordenadas
- ancho, alto, distanciaEntreRect: enteros
- anchoLienzo, altoLienzo: enteros
- rect\_color: color

Nombre de Algoritmo: dibujar\_lineas\_pelotas

#### Proceso del Algoritmo:

```
anchoLienzo ← 600
altoLienzo ← 600
distanciaRect ← 20
ancho ← 40
alto ← 20
hacer circulo ← distanciaCirculo
hacer línea(lineaX, lineaY, ancho, lineaY)
    circulo(circuloX, circuloY, 50, 50)
Mientras(circulo < ancho)
    lineaX += 100
    circuloY += 200;
Mientras(lineaY < alto)
fin
```

```
1 int alto, ancho;
2
3 void setup(){
4     size(600,600);
5     int lineaX = 0;
6     int lineaY = 100;
7     int circuloY = 75;
8     int distanciaCirculo = 30;
9
10    do{
11        int circuloX = distanciaCirculo;
12
13    do{
14        stroke(#008DFC);
15        line(lineaX,lineaY,width,lineaY);
16        fill(random(255), random(255), random(255));
17        stroke(0);
18        strokeWeight(2);
19        ellipse(circuloX,circuloY,50,50);
20        circuloX += distanciaCirculo*2;
21
22    }while(circuloX < width);
23        lineaY += 100;
24        circuloY += 200;
25
26    }while(lineaY < height);
27 }
```