



TECNICATURA UNIVERSITARIA EN DISEÑO
INTEGRAL DE VIDEOJUEGOS

FACULTAD DE INGENIERÍA
Universidad Nacional de Jujuy



FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS

Trabajo Práctico Nº 1
Canchi Mario Alejandro
LU: 6948

Profesores:
Mg. Ing. Ariel Alejandro Vega
2024

Indice

Ejercicio 01.....	01
Ejercicio 02.....	01
Ejercicio 04.....	02
Ejercicio 05.....	05
Ejercicio 06.....	06
Ejercicio 07.....	06
Ejercicio 08.....	07
Ejercicio 09.....	07
Ejercicio 10.....	07
Ejercicio 11.....	08
Ejercicio 12.....	08
Ejercicio 13.....	09
Ejercicio 14.....	11
Ejercicio 15.....	12
Ejercicio 16.....	14
Ejercicio 17.....	15
Ejercicio 18.....	18
Ejercicio 19.....	20
Ejercicio 20.....	22
Ejercicio 21.....	24
Ejercicio 22.....	26

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

$$3 * A - 4 * B / A^2$$

$$(3 * A) - (4 * B / (A^2))$$

$$6 - (4 * B / 4)$$

$$6 - 5$$

$$1$$

Ejercicio 01

```
int A=2;
int B=5;

float solucion = 3 * A - 4 * B / pow(A,2);
println("El resultado es: "+solucion);
```

Se ha guardado el sketch.

El resultado es: 1.0

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

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

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

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

$$(1) + ((0,12 / 4) * 2)$$

$$(1) + (0,03 * 2)$$

$$(1) + (0,06)$$

$$1,06$$

Ejercicio 02

```
float solucion = (( (4/2) * 3) / 6) + (((((6/2) / 1) / pow(5,2)) / 4)*2);
println("El resultado es: "+solucion);
```

Se ha guardado el sketch.

El resultado 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) $b^2 - 4 \cdot a \cdot c$

$a=4, b=6, c=2$

$$(6^2) - (4 \cdot 4 \cdot 2)$$

$$36 - 32$$

$$4$$

Expresión Algebraica:

$$b^2 - 4 \cdot a \cdot c$$

$$6^2 - 4 \cdot 4 \cdot 2$$

$$36 - 32 = 4$$

```
Ejercicio 04a
int a=4, b=6, c=2;

float solucion=pow(b,2)-4 * a * c;
println("El resultado es: "+solucion);
```

El resultado es: 4.0

b) $3 \cdot X^4 - 5 \cdot X^3 + X \cdot 12 - 17$
 $(3 \cdot (2^4)) - (5 \cdot (2^3)) + (2 \cdot 12) - 17$
 $(3 \cdot 16) - (5 \cdot 8) + 24 - 17$
 $48 - 40 + 24 - 17 = 15$

$x=2$

Expresion Algebraica:

$$3x^4 - 5x^3 + x \cdot 12 - 17$$

$$3 \cdot 2^4 - 5 \cdot 2^3 + 2 \cdot 12 - 17$$

$$3 \cdot 16 - 5 \cdot 8 + 24 - 17$$

$$48 - 40 + 24 - 17 = 15$$

```
Ejercicio 04b
int X = 2;

float solucion = 3 * pow(X, 4) - 5 * pow(X, 3) + X * 12 - 17;
println("El resultado es: "+solucion);
```

Se ha guardado el sketch.

El resultado es: 15.0

c) $(b + d) / (c + 4)$
 $(4 + 8) / (2 + 4)$

$b=4, c=2, d=8$

$$12 / 6 = 2$$

Expresion Algebraica:

$$\frac{(b + d)}{(c + 4)} = \frac{(4 + 8)}{(2 + 4)} = \frac{12}{6} = 2$$

```
Ejercicio 04c
int b = 4;
int d = 8;
int c = 2;

float solucion = (b + d) / float(c + 4);
println("El resultado es: "+solucion);
```

El resultado es: 2.0

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

$x=3, y=4$

$(3^2 + 4^2)^{1/2}$

$(9 + 16)^{1/2}$

$(25)^{1/2} = 5$

Expresión Algebraica:

$(x^2 + y^2)^{(\frac{1}{2})}$

$(3^2 + 4^2)^{(\frac{1}{2})}$

$(9 + 16)^{(\frac{1}{2})}$

$25^{(\frac{1}{2})} = 5$

Ejercicio 04d

```
int x = 3;
int y = 4;

float solucion = pow(pow(x, 2) + pow(y, 2), 0.5);
println("El resultado es: "+solucion);
```

El resultado es: 5.0

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

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

$20 - (25/4)$

$20 - 6,25 = 13,75$

Ejercicio 05a

```
int A = 4;
int B = 5;
int C = 1;

float solucion = B * A - (pow(B, 2) / 4) * C;
println("El resultado es: "+solucion);
```

El resultado es: 13.75

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

$(4 * 5) / 3^2$

$20 / 9 = 2,22 \dots$

Ejercicio 05b

```
int A = 4;
int B = 5;

float solucion = (A * B) / pow(3, 2);
println("El resultado es: "+solucion);
```

El resultado es: 2.2222223

c) $((B + C) / 2 * A + 10) * 3 * B - 6$
 $((5 + 1) / 2 * 4 + 10) * 3 * 5 - 6$
 $((6 / 2 * 4 + 10) * 3 * 5) - 6$
 $((3 * 4 + 10) * 3 * 5) - 6$
 $((12 + 10) * 3 * 5) - 6$
 $(22 * 3 * 5) - 6$
 $(66 * 5) - 6$
 $330 - 6 = 324$

Ejercicio 05c

```
int A=4;
int B=5;
int C=1;

float solucion = (((B+C)/2*A+10)*3*B)-6;
println("El resultado es: "+solucion);
```

Se ha guardado el sketch.

El resultado es: 324.0

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

$R1 = y+z$
 $R2 = x >= R1$
Resolución:
 $R1 = 4+1 = 5$
 $R2 = 3 >= R1$
 Falso

Ejercicio 06

```
int x = 3;
int y = 4;
int z = 1;

int R1 = y + z;
boolean R2 = 3 >= R1;
println("El resultado es: "+R2);
```

Se ha guardado el sketch.

El resultado es: false

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

$R1 = ++contador1$
 $R2 = contador1 < contador2$
Resolución:
 $R1 = 4$
 $R2 = 4 < 4$
 Falso

Ejercicio 07

```
int contador1 = 3;
int contador2 = 4;
int R1 = ++contador1;
boolean R2 = contador1 < contador2;
println("El resultado es: "+R2);
```

Se ha guardado el sketch.

El resultado es: false

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

$$a + b - 1 < x * y$$

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

$$29 < 6$$

Falso

```
Ejercicio 08
int a = 31;
int b = -1;
int x = 3;
int y = 2;
boolean valor = a + b - 1 < x * y;
println("El resultado es: "+valor);

Se ha guardado el sketch.

El resultado es: false
```

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

$$!(x < 5) \&\& !(y \geq 7)$$

Resolución:

$$!(6 < 5) \&\& !(8 \geq 7)$$

verdadero && falso

falso

```
Ejercicio 09
int x = 6;
int y = 8;
boolean valor = !(x < 5) && !(y >= 7);
println("El resultado es: "+valor);

Se ha guardado el sketch.

El resultado es: false
```

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

$$!((i > 4) \mid\mid !(j \leq 6))$$

Resolución:

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

!(verdadero $\mid\mid$ falso)

!(verdadero)

Falso

```
Ejercicio 10
int i = 22;
int j = 3;
boolean valor = !((i > 4) \mid\mid !(j <= 6));
println("El resultado es: "+valor);

El resultado es: false
```

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

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

Resolución:

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

$!(46==8) \ || \ (8!=0) \ \&\& \ (4>=19)$

verdadero $\ || \$ verdadero $\ \&\& \$ falso

verdadero $\ || \$ falso

verdadero

```

Ejercicio 11
int a = 34;
int b = 12;
int c = 8;
boolean valor = !(a + b == c) || (c != 0) && (b - c >= 19);
println("El resultado es: "+valor);

Se ha guardado el sketch.

El resultado 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

Especificación del Problema: Deberá pedir por teclado al usuario un nombre y posteriormente realizará la presentación en pantalla de un saludo con el nombre indicado.

- Análisis:

Datos de Entrada: nombre: String

Datos de Salida: mensaje de saludo: String

- **Proceso:** Ingresar un nombre por teclado y mostrar un saludo en pantalla.
- **Fase de diseño:**

ENTIDAD QUE RESUEVE EL PROBLEMA: algoritmo
VARIABLES nombre: string // almacena el nombre
NOMBRE DEL ALGORITMO: saludo PROCESO DEL ALGORITMO: 1. Leer nombre 2. Mostrar "Escribe tu nombre y presiona la tecla Enter" 3. $nombre \leftarrow keyPressed()$ 4. Mostrar "Saludos "+nombre+ "Bienvenido a Processing"

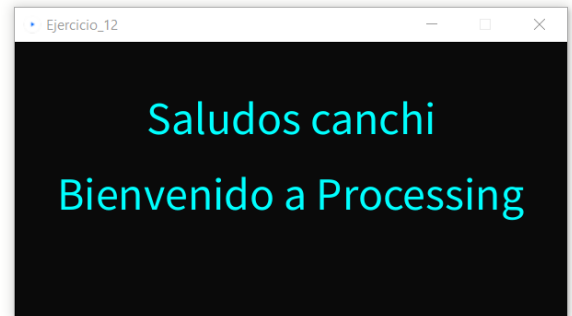
5. fin

```
String nombre="";
void setup(){
  size(600,300);
  println("Escribe tu nombre y presiona la tecla Enter");
}

void draw() {
  fill(10);
  rect(0,0,600,300);
  fill(0, 400, 590);
  textAlign(CENTER);
  textSize(50);
  text(nombre,300,150);
  textSize(30);
  text("Escribe tu nombre y presiona la tecla Enter",296,63);

  if (key == '\n'){
    fill(10);
    rect(0,0,600,300);
    textAlign(CENTER);
    textSize(50);
    fill(0, 400, 590);
    text("Saludos "+nombre+ "Bienvenido a Processing",300,100);
  }
}

void keyPressed(){
  nombre+= key;
  if (key == '\n'){
    println("Saludos "+nombre+ "Bienvenido a Processing");
  }
}
```



Depurador en funcionamiento...

```
Escribe tu nombre y presiona la tecla Enter
Saludos canchi
Bienvenido a Processing
```

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:

- **Especificación del Problema:** Calcule el perímetro y área de un rectángulo dada su base y su altura.

- Análisis:

Datos de Entrada: base: int

altura: int

Datos de Salida: perímetro: int

área: int

- **Proceso:** calcular el perímetro y área de un rectángulo dado su base y altura usando la formula: $\text{Perimetro} = 2(\text{base} + \text{altura})$ $\text{Area} = \text{base} * \text{altura}$

- Fase de diseño:

ENTIDAD QUE RESUEVE EL PROBLEMA: algoritmo
VARIABLES base: int // almacena la base del rectángulo altura: int // almacena la altura del rectángulo perimetro: int // almacena el perímetro del rectángulo area: int // almacena el área del rectángulo
NOMBRE DEL ALGORITMO: perimetro_area 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 "El perimetro es: "+perimetro+", el área es: "+area" 6. fin

Ejercicio 13

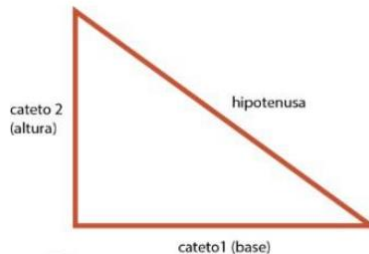
```
int base=2;
int altura=4;

int perimetro= 2*(base+altura);
int area= base*altura;
println("El perimetro es: "+perimetro+", el área es: "+area);
```

Se ha guardado el sketch.

El perimetro es: 12, el área es: 8

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:**

- **Especificación del Problema:** Obtenga la hipotenusa de un triángulo rectángulo conociendo sus catetos

- **Análisis:**

Datos de Entrada: base: int

altura: int

Datos de Salida: hipotenusa: float

- **Proceso:** calcular la hipotenusa de un triángulo rectángulo dado su base y altura usando la fórmula: $(a^2 + b^2)^{0.5}$

- **Fase de diseño:**

ENTIDAD QUE RESUEVE EL PROBLEMA: algoritmo
VARIABLES a: int // almacena la base del triángulo b: int // almacena la altura del triángulo hipotenusa: float // almacena la hipotenusa del triángulo
NOMBRE DEL ALGORITMO: calculo_hipotenusa PROCESO DEL ALGORITMO: 1. Leer a 2. Leer b 3. $hipotenusa \leftarrow (a^2 + b^2)^{0.5}$ 4. Mostrar hipotenusa 5. fin

Ejercicio 14

```
int a = 6;
int b = 8;

float hipotenusa = (int) pow(pow(a, 2) + pow(b, 2), 0.5);
println("La hipotenusa tiene un valor de : "+hipotenusa);
```

Se ha guardado el sketch.

La hipotenusa tiene un valor de : 10.0

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:**

- **Especificación del Problema:** Dados dos números permita calcular la suma, resta, multiplicación y división de estos.

- **Análisis:**

Datos de Entrada: a: int

b: int

Datos de Salida: suma: int

resta: int

división: float

multiplicación: int

- **Proceso:** calcular la suma, resta, multiplicación y división de dos números usando la formula: $\text{suma} = a + b$ $\text{resta} = a - b$ $\text{división} = a / b$ $\text{multiplicación} = a * b$

- **Fase de diseño:**

ENTIDAD QUE RESUEVE EL PROBLEMA: algoritmo
VARIABLES
a: int // almacena un numero entero
b: int // almacena un numero entero
suma: int // almacena la suma de a y b
resta: int // almacena la resta de a y b

multiplicacion: int // almacena la multiplicacion de a y b

division: float // almacena la division de a y b

NOMBRE DEL ALGORITMO: suma

PROCESO DEL ALGORITMO:

1. Leer a
2. Leer b
3. suma $\leftarrow a+b$
4. Mostrar suma
5. fin

NOMBRE DEL ALGORITMO: resta

PROCESO DEL ALGORITMO:

1. Leer a
2. Leer b
3. resta $\leftarrow a-b$
4. Mostrar resta
5. fin

NOMBRE DEL ALGORITMO: multiplicacion

PROCESO DEL ALGORITMO:

1. Leer a
2. Leer b
3. multiplicacion $\leftarrow a*b$
4. Mostrar multiplicacion
5. fin

NOMBRE DEL ALGORITMO: division

PROCESO DEL ALGORITMO:

1. Leer a
2. Leer b
3. division $\leftarrow a/b$
4. Mostrar division
5. fin

Ejercicio 15

```
int a=100;
int b=50;
int suma = a + b;
int resta = a - b;
int multiplicacion = a * b;
float division = a / b;
println("El resultado de la suma es: " + suma);
println("El resultado de la resta es: " + resta);
println("El resultado de la multiplicación es: " + multiplicacion);
println("El resultado de la división es: " + division);
```

```
El resultado de la suma es: 150
El resultado de la resta es: 50
El resultado de la multiplicación es: 5000
El resultado de la división es: 2.0
```

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

Especificación del Problema: Necesitamos convertir una temperatura Fahrenheit en grados Celsius

- Análisis:

Datos de Entrada: temperaturaFahrenheit: float

Datos de Salida: temperaturaCelsius: float

- **Proceso:** Calcular la temperatura Celsius dada una temperatura Fahrenheit usando la formula: $\text{temperaturaCelsius} = (((\text{temperaturaFahrenheit} - 32) * 5) / 9)$
- **Fase de diseño:**

ENTIDAD QUE RESUEVE EL PROBLEMA: algoritmo
VARIABLES temperaturaFahrenheit: float // almacena la temperatura Fahrenheit temperaturaCelsius: float // almacena la temperatura Celsius
NOMBRE DEL ALGORITMO: calculo_temperatura PROCESO DEL ALGORITMO: 1. Leer temperaturaFahrenheit 2. $\text{temperaturaCelsius} \leftarrow (((\text{temperaturaFahrenheit} - 32) * 5) / 9)$

3. Mostrar "La temperatura Celsius es: " + temperaturaCelsius

4. fin

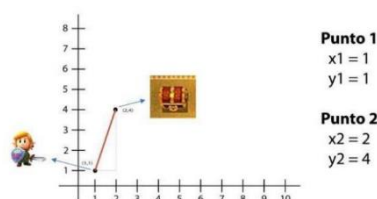
Ejercicio 16

```
float temperaturaFahrenheit = 50;
float temperaturaCelsius = (((temperaturaFahrenheit - 32)*5)/9);
println("La temperatura en Celsius es: " + temperaturaCelsius);
```

Se ha guardado el sketch.

La temperatura en Celsius es: 10.0

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 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 Circulo, y al tesoro con un cuadrado. Además, mueva a Link mediante el mouse.

- Fase de Análisis

Especificación del Problema: Calcular la distancia entre Link y el Tesoro

- Análisis:

Datos de Entrada: Movimiento del Mouse

Datos de Salida: un círculo y un rectángulo que representan a Link y el Tesoro

- **Proceso:** Calcular la distancia de las coordenadas x, y que nos darán los catetos formados por ambos.
- **Fase de diseño:**

ENTIDAD QUE RESUEVE EL PROBLEMA: usuario
VARIABLES posLink: float // almacena la posición de Link posTesoro: float // almacena la posición del Tesoro distancia: float // almacena el resultado de un calculo cateto1: float // almacena el valor de un cateto cateto2: float // almacena el valor de un cateto
NOMBRE DEL ALGORITMO: dibujarLink PROCESO DEL ALGORITMO: 1. Leer posLink.x 2. Leer posLink.y 3. $Link \leftarrow (posLink.x, posLink.y, 40)$ 4. fin
NOMBRE DEL ALGORITMO: dibujarTesoro PROCESO DEL ALGORITMO: 1. Leer posTesoro.x 2. Leer posTesoro.y 3. $Tesoro \leftarrow (posTesoro.x, posTesoro.y, 100, 70);$ 4. fin
NOMBRE DEL ALGORITMO: obtenerDistancia PROCESO DEL ALGORITMO: 1. Leer posTesoro.x



2. Leer posTesoro.y

3. Leer posLink.x

4. Leer posLink.y

5. cateto1 ← posTesoro.x-posLink.x

6. cateto2 ← posTesoro.y-posLink.y

7. distancia ← $\sqrt{\text{pow}(\text{cateto1},2)+\text{pow}(\text{cateto2},2)}$;

8. fin

NOMBRE DEL ALGORITMO: validarDibujarTesoro

PROCESO DEL ALGORITMO:

1. si (obtenerDistancia()<70) entonces

2. *Mostrar*("Debe desaparecer el tesoro")

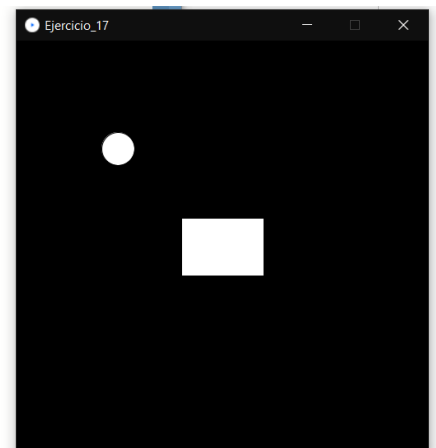
3. sino

4. *dibujarTesoro* fin si

5. fin

```
Ejercicio 17
1 PVector posLink;
2 PVector posTesoro;
3
4 public void setup(){
5     size(500,500);
6     posLink = new PVector (100,100);
7     posTesoro = new PVector(width/2, height/2);
8 }
9
10 public void draw(){
11     background(0);
12     dibujarLink();
13     posLink.x=mouseX;
14     posLink.y=mouseY;
15     validarDibujarTesoro();
16 }
17
18 public void dibujarLink(){
19     circle(posLink.x, posLink.y,40);
20 }
21
22 public void dibujarTesoro(){
23     rectMode(CENTER);
24     rect(posTesoro.x,posTesoro.y,100,70);
25 }
26
27 public float obtenerDistancia(){
28     float distancia;
29     float cateto1=posTesoro.x-posLink.x;
```

```
30
31     float cateto2=posTesoro.y-posLink.y;
32     distancia = sqrt(pow(cateto1,2)+pow(cateto2,2));
33     return distancia;
34 }
35
36 public void validarDibujarTesoro(){
37     if(obtenerDistancia()<70){
38         println("Debe desaparecer el tesoro");
39     }else{
40         dibujarTesoro();
41     }
42 }
43 }
```



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

Especificación del Problema: Obtener las raíces de una ecuación de segundo grado

- Análisis:

Datos de Entrada: a, b, c

Datos de Salida: x1, x2

-Proceso: : Se ingresarán los valores de a, b y c, luego se procederá a calcular la discriminante ($\text{discriminante} \leftarrow b^2 - 4 \cdot a \cdot c$), en base al resultado se realizarán diferentes procesos para obtener las raíces de la ecuación de segundo grado usando las siguientes formulas:

$$x1 = \frac{-b + \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a}$$

$$x2 = \frac{-b - \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a}$$

- Fase de diseño:

ENTIDAD QUE RESUEVE EL PROBLEMA: La Calculadora
VARIABLES a: float // almacena el valor de la variable a. b: float // almacena el valor de la variable b. c: float // almacena el valor de la variable c. discriminante: float // almacena el valor de la discriminante estado: int // almacena un estado de la discriminante y usarse en la estructura Segun x1: float // almacena el valor de la variable x1. x2: float // almacena el valor de la variable x2.
NOMBRE DEL ALGORITMO: calcular_raices PROCESO DEL ALGORITMO: 1. Mostrar "Ingrese el valor de a: " 2. Leer a 3. Mostrar "Ingrese el valor de b: "

4. Leer b

5. Mostrar "Ingrese el valor de c: "

6. Leer c

7. discriminante $\leftarrow b^2 - 4*a*c$

8. según_sea (discriminante) hacer

9. caso 1:

10. si (discriminante > 0) entonces

11. $x1 \leftarrow (-b + \text{sqrt}(\text{discriminante})) / (2*a)$

12. $x2 \leftarrow (-b - \text{sqrt}(\text{discriminante})) / (2*a)$

13. Mostrar "El valor de x1 es: " + x1

14. Mostrar "El valor de x2 es: " + x2

15. fin_si

16. caso 2:

17. si (discriminante = 0) entonces

18. $x1 \leftarrow -b / (2*a);$

19. Mostrar "El valor de x1 es: " + x1

20. Mostrar "El valor de x2 es: " + x1

21. fin_si

22. caso 3:

23. si (discriminante < 0) entonces

24. Mostrar "No existe solución en el conjunto de números Reales"

25. fin_si

26. fin_según

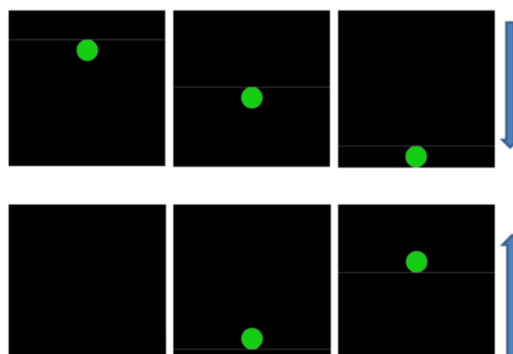
```

Ejercicio 18
float a=1;
float b=2;
float c=1;
float discriminante;
float x1;
float x2;
int estado=1;

discriminante = pow(b,2) - 4*a*c;
if(discriminante > 0){
    estado=1;
}else if(discriminante==0){
    estado=2;
}else if(discriminante<0){
    estado=3;
}
switch(estado) {
    case 1:
        x1 = (-b + sqrt(discriminante)) / (2*a);
        x2 = (-b - sqrt(discriminante)) / (2*a);
        println("El valor de x1 es: "+x1);
        println("El valor de x2 es: "+x2);
        break;
    case 2:
        x1 = -b / (2*a);
        println("El valor de x1 es: "+ x1);
        println("El valor de x2 es: "+ x1);
        break;
    case 3:
        println("No existe solución en el conjunto de números Reales");
        break;
}

```

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

Especificación del Problema: Declare las variables necesarias para dibujar una línea que se dibuja desde las coordenadas iniciales del lienzo y se extiende por todo el ancho

- Análisis:

Datos de Entrada: Radio

Datos de Salida: una Línea y un Circulo

- Fase de diseño:

ENTIDAD QUE RESUEVE EL PROBLEMA: Lienzo
VARIABLES posY: int incremento: int distanciaRadio: int
NOMBRE DEL ALGORITMO: dibujar_circulo PROCESO DEL ALGORITMO: 1. Leer incremento 2. Leer distanciaRadio 3. posY \leftarrow incremento 4. si (posY >= height o posY <= 0) entonces 5. incremento \leftarrow incremento * -1 6. distanciaRadio \leftarrow distanciaRadio * -1 7. fin si 8. fin

Ejercicio 19

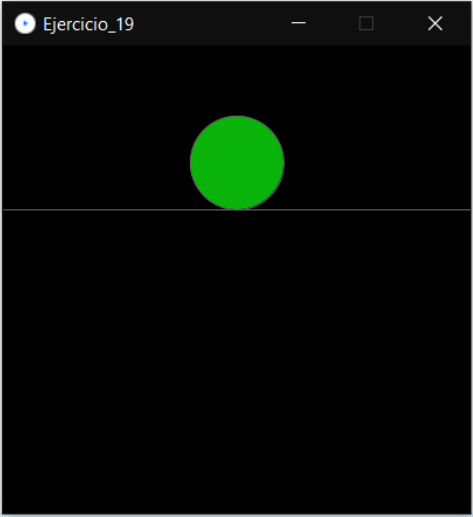
```

int posY;
int incremento=1;
int distanciaRadio=40;

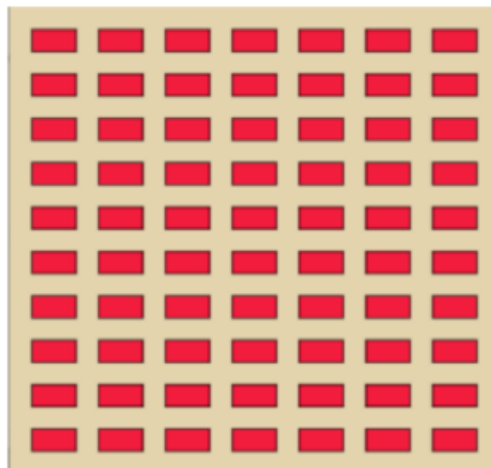
void setup(){
    size(400,400);
    posY=0;
}

void draw(){
    background(0);
    stroke(100);
    line(0,posY,width,posY);
    fill(10,180,10);
    ellipse(width/2,posY+distanciaRadio,80,80);

    posY+=incremento;
    if (posY>=height || posY<=0){
        incremento*=-1;
        distanciaRadio*=-1;
    }
}
    
```



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:



Análisis:

Descripción del problema: Dibujar en Processing la imagen indicada

Análisis:

Datos de Entrada: rectAlto, rectAncho, espacio.

Datos de Salida: El lienzo dibujado.

Proceso: Dibujar los rectángulos con un espaciado de 20 pixeles entre ellos.

Diseño:

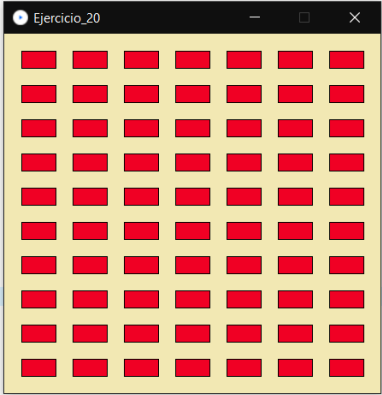
Entidad que resuelve el Problema: Lienzo
Variables: <ul style="list-style-type: none"> - rectAlto: int // Almacena el alto del rectángulo - rectAncho: int // Almacena el ancho del rectángulo - espacio: int // Almacena el espacio entre rectángulos - height: int // Representa el alto del lienzo - width: int // Representa el ancho del lienzo
Nombre del Algoritmo: rectángulos_espaciados Proceso: <ol style="list-style-type: none"> 1. inicio 2. Leer width \leftarrow 440 3. Leer height \leftarrow 420 4. Leer rectAncho \leftarrow 40 5. Leer rectAlto \leftarrow 20 6. Leer espacio \leftarrow 20 7. para y espacio hasta y < height – rectAlto con paso y += rectAlto + espacio 8. hacer 9. para x = espacio hasta x < width – rectAncho con paso x += rectAncho + espacio 10. hacer 11. Mostrar rect (x,y, rectAncho, rectAlto) 12. fin-para 13. fin-para 14. fin

Ejercicio 20

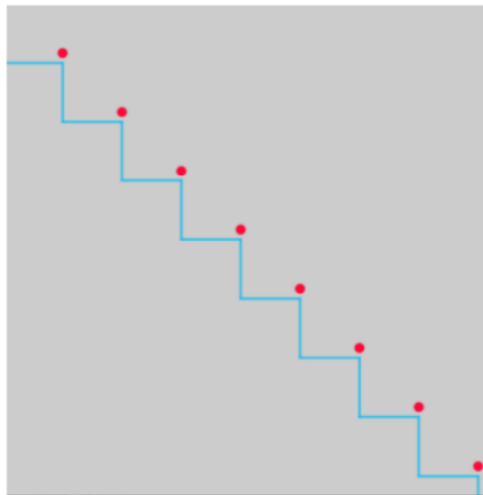
```
void setup() {
    size(440, 420);
}

void draw() {
    background(#F2E8B3);
    fill(#F00024);
    int rectAncho = 40;
    int rectAlto = 20;
    int espacio = 20;

    for (int y = espacio; y < height - rectAlto; y += rectAlto + espacio) {
        for (int x = espacio; x < width - rectAncho; x += rectAncho + espacio) {
            rect(x, y, rectAncho, rectAlto);
        }
    }
}
```



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.

Análisis:

Descripción del problema: Dibujar en Processing la imagen indicada

Análisis:

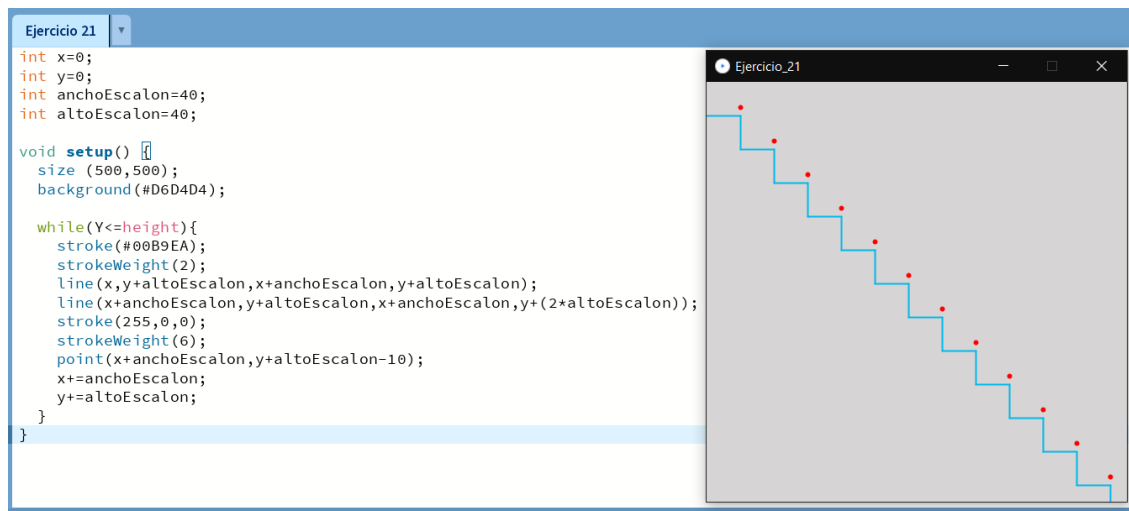
Datos de Entrada: x, y, anchoEscalon, altoEscalon.

Datos de Salida: El lienzo dibujado.

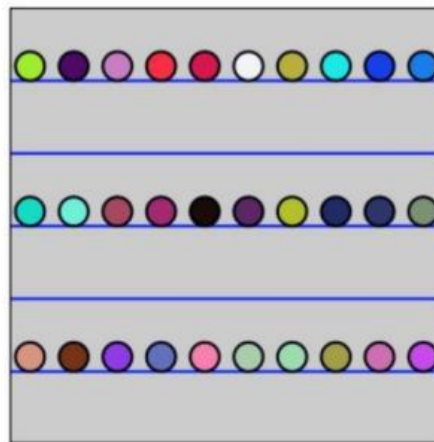
Proceso: Dibujar líneas que forman escalones y sobre cada borde de escalón se dibuje un punto de color rojo.

Diseño:

Entidad que resuelve el Problema: Lienzo
Variables: <ul style="list-style-type: none"> - x: int - y: int - anchoEscalon: int - altoEscalon: int
Nombre del Algoritmo: dibujar_escalones Proceso: <ol style="list-style-type: none"> 1. inicio 2. Leer $x \leftarrow 0$ 3. Leer $y \leftarrow 0$ 4. Leer $\text{anchoEscalon} \leftarrow 40$ 5. Leer $\text{altoEscalon} \leftarrow 40$ 6. Mientras ($Y < \text{height}$) hacer 7. Mostrar linea $\leftarrow (x, y + \text{altoEscalon}, x + \text{anchoEscalon}, y + \text{altoEscalon})$; 8. Mostrar linea $\leftarrow (x + \text{anchoEscalon}, y + \text{altoEscalon}, x + \text{anchoEscalon}, y + (2 * \text{altoEscalon}))$ 9. Mostrar punto $\leftarrow (x + \text{anchoEscalon}, y + \text{altoEscalon} - 10)$ 10. fin



Ejercicio 22: Utilizando la estructura de control repetitiva do-while. Replique la siguiente imagen



La imagen debe ser construida desde la función setup(). Defina el tamaño del lienzo en size(600,600), verticalmente se divide el lienzo en franjas de igual medida, se deben dibujar los círculos sobre cada línea de por medio es decir en la línea 1 se dibujan círculos con distanciamiento, en la línea 2 no se dibuja y así sucesivamente. Las líneas tienen un color fijo, los círculos asumen colores aleatorios.

Análisis:

Descripción del problema: Dibujar los círculos sobre cada línea de por medio

Análisis:

Datos de Entrada: y, círculos, radio.

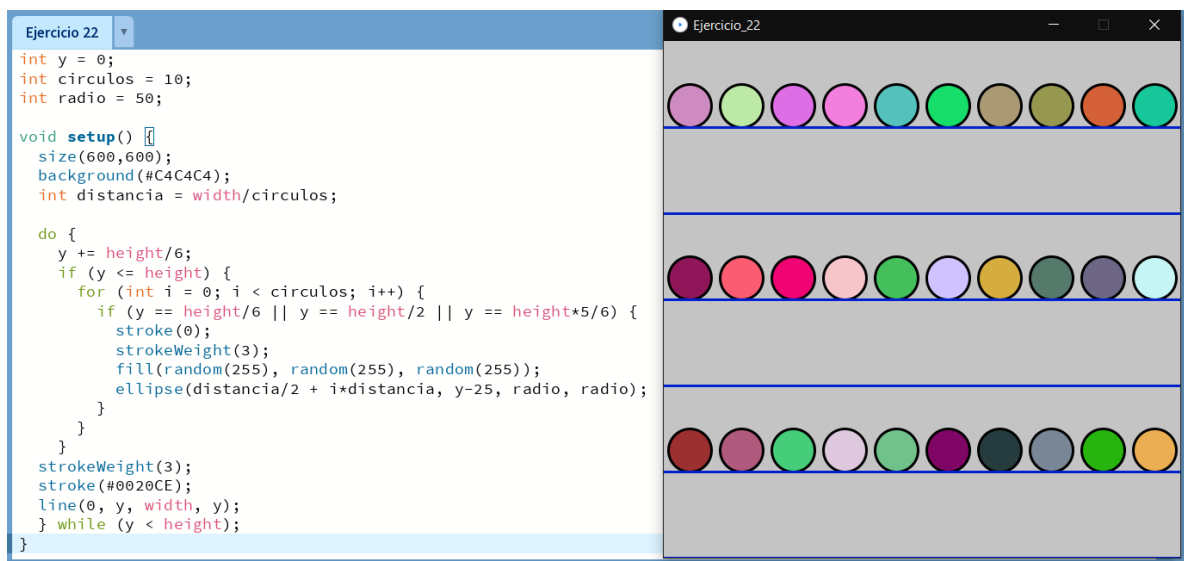
Datos de Salida: El lienzo dibujado.

Proceso: Dibujar los círculos de diferentes colores sobre cada línea de por medio.

Diseño:

Entidad que resuelve el Problema: Lienzo
Variables: <ul style="list-style-type: none"> - x: int - círculos: int - radio: int - distancia: int
Nombre del Algoritmo: dibujar_círculos Proceso: <ol style="list-style-type: none"> 1. inicio 2. Leer y $\leftarrow 0$ 3. Leer círculos $\leftarrow 10$

4. Leer radio $\leftarrow 50$
5. Leer distancia $\leftarrow \text{width}/\text{circulos}$
6. **Mientras** ($y < \text{height}$) **hacer**
7. $y \leftarrow y + \text{height}/6$
8. **Si** ($y \leq \text{height}$) **entonces**
9. **Para** $i \leftarrow 0$ **hasta** $i \leftarrow \text{circulos}$ **con paso** $i++$ **hacer**
10. **Mostrar** circulo $\leftarrow (\text{distancia}/2 + i * \text{distancia}, y-25, \text{radio}, \text{radio})$
11. **Fin para**
12. **Fin si**
13. **Mostrar** linea $\leftarrow (0, y, \text{width}, y)$
14. **Fin Mientras**
15. **Fin**



Conclusión

El Trabajo Practico pone a prueba los conocimientos básicos sobre Processing, aprendí nuevos conceptos y nuevas formas de relacionar las estructuras repetitivas.

Fuentes bibliográficas

[Fundamentos de Programación Orientada a Objetos](#)

[Processing Reference](#)