



TECNICATURA UNIVERSITARIA EN DISEÑO
INTEGRAL DE VIDEOJUEGOS

FACULTAD DE INGENIERÍA
Universidad Nacional de Jujuy



FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS

Trabajo Práctico

N°1

Apellido y Nombre :

Ibáñez Esteban Agustín

Profesores:

Mg. Ing. Ariel Alejandro Vega

REGLAMENTO:

Crear una carpeta denominada TP01_XXXX donde XXXX es el apellido_nombre del estudiante. Al producto final, subirlo en su repositorio y compartir el enlace en formulario.

Sección Expresiones aritméticas y lógicas

Resolver cada ejercicio en un archivo Word y luego programarlo en Processing. En el caso

de la programación crear un archivo por ejercicio.

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

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

Resolución necesaria en Word:

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

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

$$6-5$$

$$1$$

Captura de Processing

```

1  int A=2, B=5;
2
3  float resultado = (3* A) - (4 * B) / (pow(A,2));
4
5  println(resultado);

```




Ojo: Colocar la captura, no reemplaza que deban agregar a la carpeta el archivo .pde que contiene el código programado.

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

Aritmética	Algebraica
$4 / 2 * 3 / 6 + 6 / 2 / 1 / 5 ^ 2 / 4 * 2$ $((4/2) * 3) / 6 + (((6/2) / 1) / (5^2)) / 4 * 2$ $1,0 + 0,06$ $1,06$	$\frac{\left(\frac{4}{2} \times 3\right)}{6} + \left(\frac{6}{\frac{2}{\frac{1}{5^2}}}\right)$

Captura del processing y resultado:

<pre>1 float answer = (((4/2) *3)/6)+((((6/2) /1)/ pow(5,2)/4)*2); 2 3 println(answer);</pre>	
---	--

Ejercicio 4: Evaluar las siguientes expresiones aritméticas, para lo cual indicar en el caso

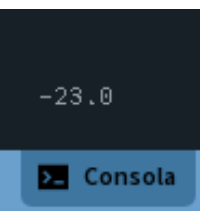
de las variables, el valor indicado. Luego escribirlas como expresiones algebraicas.

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

$A=2,B=3,C=4$

Aritmética	Algebraica
$b ^ 2 - 4 * a * c$ $((3^2)) - (4* 2* 4)$ $9 - 32$ -23	$b^2 - 4.a.c$ $3^2 - 4.2.4$ -23

Captura del processing y resultado:


<pre>1 int A=2,B=3,C=4; 2 3 float answer = ((pow(B,2)) -(4 * A *C)); 4 5 println(answer);</pre>	
---	---



b) $3 * X^4 - 5 * X^3 + X^{12} - 17$ $X=3$

Aritmética	Algebraica
$3 * X^4 - 5 * X^3 + X^{12} - 17$	$3 * 3^4 - 5 * 3^3 + 3 * 12 - 17$
$3 * 3^4 - 5 * 3^3 + 3 * 12 - 17$	$3 * 81 - 5 * 27 + 36 - 17$
$243 - 135 + 36 - 17$	$243 - 135 + 36 - 17$
$108 + 19$	$108 + 19$
127	127


Captura del processing y resultado:

<pre>1 int X=3; 2 3 float resultado = (3 * pow(X,4))-(5 * pow(X,3))+ ((X*12) -17) ; 4 5 println(resultado);</pre>	
---	---

c) $(b + d) / (c + 4)$ $b=2$ $c=6$ $d=4$

Aritmética	Algebraica
$(b + d) / (c + 4)$	$\frac{b + d}{c + 4}$
$(2 + 4) / (6 + 4)$	$2 + 4 / 6 + 4$
$6/10$	$6 / 10$
0.6	0,6

Captura del processing y resultado:


<pre>1 int b=2, d=4, c=6; 2 3 float respuesta = (b + d) / (c + 4.0); 4 5 println(respuesta);</pre>	
--	---

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

$x=5 \ y=10$

Aritmética	Algebraica
$(x^2 + y^2)^{1/2}$ $(5^2 + 10^2)^{1/2}$ $(125)^{1/2}$ 11,180	$(5^2 + 10^2)^{\frac{1}{2}}$ $(25+100)^{1/2}$ $125^{1/2}$ 11,180

Captura del processing y resultado:

<pre> 1 int x= 5, y=10; 2 3 float rta = pow(pow(x,2.0)+pow(y,2.0),0.5); 4 5 println(rta); 6 </pre>	 <p>11.18034</p> <p>>_ Consola</p>
--	---


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

Aritmética	Algebraica
$B * A - B^2 / 4 * C$ $20 - 25/4$ 13.75	$5 * 4 - 5^2 / 4 * 1$ $20 - 25/4$ 13.75


Captura del processing y resultado:

<pre> 1 int A=4,B=5,C=1; 2 3 float respuesta= (B*A) - (pow(B,2)) / (4*C); 4 5 println(respuesta); 6 </pre>	 <p>13.75</p> <p>>_ Consola</p>
--	---

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

Aritmética	Algebraica
$(A * B) / 3^2$ $4 * 5 / 3^2$ $20/9$ $2,2...$	$(4 . 5) / 3^2$ $20 / 9$ $2,2...$


Captura del processing y resultado:

<pre>1 int A=4,B=5,C=1; 2 3 float respuesta = (A*B) /pow(3,2); 4 5 println(respuesta);</pre>	
--	---

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

Aritmética	Algebraica
$((B + C) / 2 * A + 10) * 3 * B - 6$ $((5 + 1) / 2 * 4 + 10) * 3 * 5 - 6$ $((6/2).4+10)*3*5) -6$ $((22.3.5) -6$ $330 - 6$ 324	$((5 + 1)/ 2 . 4 + 10) . 3 . 5) -6$ $\left(\left(\frac{5+1}{2} \cdot 4 \right) + 10 \right) \cdot 3 \cdot 5 - 6 = 324$ $((6/2).4+10).3.5)-6$ $22 . 3 . 5 - 6$ 324

Captura del processing y resultado:

<pre>1 int A=4,B=5,C=1; 2 3 float respuesta = (((B +C) /2 * A +10) * 3 * B) -6; 4 5 println(respuesta);</pre>	
---	---


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

$R1 = y+z$

$R2 = x \geq R1$

SOLUCIÓN
$R1 = 4 + 1 = 5$ $R2 = 3 \geq 5$ Falso

Captura del processing y resultado:

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

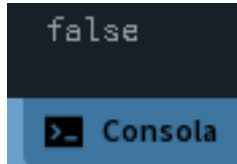
Ejercicio 7: Para contador1=3, contador3=4, evaluar el resultado de

$R1 = ++\text{contador1}$

$R2 = \text{contador1} < \text{contador2}$

SOLUCIÓN
$R2 = 4 < 4$ $R2 = \text{FALSO}$

Captura del processing y resultado:

<pre> 1 int contador1=3, contador2=4; 2 int R1 = ++contador1; 3 boolean R2 = contador1 < contador2; 4 println(R2); </pre>	
--	---

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

$a+b-1 < x*y$

SOLUCIÓN



$A+b-1 < x*y$
 $31 + (-1)-1 < 3 * 2$
 $29 < 12$
falso

Captura del processing y resultado:

```
1 int a=32,b=-1,x=3,y=2;  
2  
3 boolean resultado = a+b-1 < x* y;  
4 println(resultado);
```

false

>_ Consola

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

$!(x<5) \text{CC } !(y \geq 7)$

SOLUCIÓN

$!(x<5) \text{CC } !(y \geq 7)$

$!(6<5) \text{CC } !(8 \geq 7)$

Falso CC falso

falso

Captura del processing y resultado:

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

false

>_ Consola

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

$!((i>4) \text{ || } !(j \leq 6))$

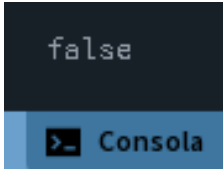
SOLUCION

$!((i>4) \text{ || } !(j \leq 6))$

$!((22>4) \text{ || } !(3 \leq 6))$

!(Verdadero || falso)
 !(verdadero)
 falso

Captura del processing y resultado:

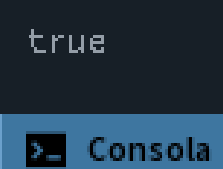
<pre> 1 int i=22,j=3; 2 3 boolean resultado = !((i>4) !(j<=6)); 4 5 println(resultado); </pre>	
--	---

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

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

<p>SOLUCION</p> <p>!(a+b==c) (c!=0)CC(b-c>=19) !(34+12==8) (8!=0)CC(12-8>=19) !(46==8) (8!=0)CC(4>=19) Verdadero verdadero falso Verdadero falso verdadero</p>

Captura del processing y resultado:

<pre> 1 int a=34, b=12, c=8; 2 3 boolean resultado = !(a+b == c) (c!=0) (b-c >= 19); 4 5 println(resultado); </pre>	
---	---

Sección Análisis – Diseño y Codificación de algoritmos – Aplicación de estructuras de control

Para cada ejercicio, en el archivo Word agregar las secciones de análisis y diseño, mientras que, para la codificación, crear el archivo de Processing.

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.

Análisis:

Datos de Entrada: // cadena
Datos de Salida: //cadena de texto
Proceso: ¿quien debe realizar el proceso?: La computadora ¿Cual es el proceso que resuelve? Ingresa caracteres que devolverá un saludo con los caracteres asignados Y su vista en pantalla.

Diseño:

Entidad que resuelve el problema: Algoritmo
Variables: Nombre:String //almacena nombre Saludo: String// almacena la cadena de saludo
Nombre del Algoritmo: saludar_nombre
Proceso del algoritmo: 1. Leer nombre 2. Saludo "Bienvenido Estimado " + nombre + "tenga un buen dia." 3. Mostrar saludo 4. fin

Captura del processing y Resultado:



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

Análisis:

Datos de Entrada: base , altura // decimal

Datos de Salida: área, perímetro// decimal

Proceso:

¿Quien debe realizar el proceso?

El usuario o calculadora

¿Cual es el proceso que resuelve?

Calcula el perímetro y el área del rectángulo utilizando una formula especifica

Diseño:

Entidad que resuelve el problema: persona

Variables:

- Base y Altura : int // almacena valores enteros

- Perimetro : float // almacena valores decimales

- Area : float // almacena valores decimales

- Perimetro1 : cadena //

- Area1 :cadena //

- Mostrar Resultados de Perimetro1 , Area1 : Println // mostrar consola



Nombre del Algoritmo: `perimetro_y_area_del_rectangulo`

Proceso del algoritmo:

1. Leer base y altura
2. $\text{Perimetro} \leftarrow 2 * (\text{base} + \text{altura})$
3. $\text{Area} \leftarrow \text{base} * \text{altura}$
4. $\text{Perimetro1} \leftarrow \text{"El perimetro es de:"} + \text{perimetro}$
5. $\text{area1} \leftarrow \text{"El area es de:"} + \text{area}$
6. Mostrar Perimetro1
7. Mostrar Area1
8. fin

Captura del processing y Resultado:

```
1 int base=7, altura=12;
2
3 float perimetro= 2*(base*altura);
4 float area = base*altura;
5 String perimetrol= "El perimetro es de:" + perimetro;
6 String areal = "El area es de:" + area;
7 println (perimetrol);
8 println(areal);
```

```
El perimetro es de:168.0
El area es de:84.0
```

Consola Errores

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.

Análisis:

Datos de Entrada: cateto1, cateto2

Datos de Salida: hipotenusa

Proceso:

- ¿Quién debe realizar el proceso?
El usuario o calculadora
- ¿Cuál es el proceso que resuelve?



Para calcular la hipotenusa de un triangulo rectangulo se requiere los valores de ambos catetos que brinda el resultado y se utiliza el teorema de pitagoras $h^2 = a^2 + b^2$

Diseño:

Entidad que resuelve el problema: persona

Variables:

- Cateto1: int // almacena valores enteros
- Cateto2: int // almacena valores enteros
- Hipotenusa: float // almacena un valor de calculos

Nombre del Algoritmo: perimetro_area_rectangulo

Proceso del algoritmo

1. Leer cateto1
2. Leer cateto2
3. Hipotenusa $\leftarrow (a^2 + b^2)^{0.5}$
4. Mostrar hipotenusa
5. fin

Captura del processing y Resultado:

```
1 int cateto1 = 12 ;//base
2 int cateto2 = 36 ;//altura
3
4 float hipotenusa = sqrt(pow(cateto1,2)+ pow(cateto2,2));
5
6 println("hipotenusa:" + hipotenusa);
```

hipotenusa:37.94733

Consola Errores

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.

Análisis:



Datos de Entrada: Numerox, NumeroY

Datos de Salida: suma , resta , multiplicación, división

Proceso:

¿Quien debe realizar el proceso?

El usuario o calculadora

¿Cual es el proceso que resuelve?

Resuelve problemas como suma , resta , division y multiplicacion a base de 2 numeros

Diseño:

Entidad que resuelve el problema: persona

Variables:

- Numerox: int // almacena valor entero
- Numeroy: int // almacena valor entero
- Suma: float // almacena valor de suma
- Resta: float // almacena valor de resta
- Multiplicacion: float // almacena un valor de multiplicación
- División: float // almacena un valor de división

Nombre del Algoritmo : Calculador

Proceso del algoritmo

1. Leer numerox
2. Leer numeroy
3. $\text{Suma} \leftarrow \text{numeroX} + \text{numeroY}$
4. $\text{Mostrar} \leftarrow \text{"Resultado de suma:"} + \text{suma}$
5. $\text{Resta} \leftarrow \text{numeroX} - \text{numeroY}$
6. $\text{Mostrar} \leftarrow \text{"Resultado de resta:"} + \text{resta}$
7. $\text{Multiplicación} \leftarrow \text{numeroX} * \text{numeroY}$
8. $\text{Mostrar} \leftarrow \text{"Resultado de multiplicación:"} + \text{multiplicación}$
9. $\text{división} \leftarrow \text{numeroX} / \text{numeroY}$
10. $\text{Mostrar} \leftarrow \text{"Resultado de división:"} + \text{division}$
11. fin

Captura del processing y Resultado:

```
1 int numeroX= 50, numeroY= 25;
2
3 float suma = numeroX + numeroY;
4 println("Resultado de suma:" + suma);
5
6 float resta = numeroX - numeroY;
7 println("Resultado de resta:" + resta);
8
9 float multiplicacion = numeroX * numeroY;
10
11 println("Resultado de multiplicacion:" + multiplicacion);
12
13 float division = numeroX / numeroY;
14
15 println("Resultado de division:" + division);
```

```
Resultado de suma:75.0
Resultado de resta:25.0
Resultado de multiplicacion:1250.0
Resultado de division:2.0
```

>_ Consola

! Errores

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.

Análisis:

Datos de Entrada:temperatura en grados Fahrenheit

Datos de Salida:temperatura grados Celsius

Proceso:

¿Quien debe realizar el proceso?

Por una calculadora o programa matematico.

¿Cual es el proceso que resuelve?

Resuelve el procedimiento de llevar una temperatura en grados fahrenheit a grados celsius.

Diseño:

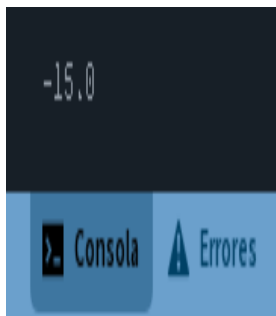
Entidad que resuelve el problema: persona
Variables: - TempFahrenheit :float // almacena valor decimal - TempCelsius:float // almacena valor decimal
Nombre del Algoritmo: conversor_de_temperatura_celsius
Proceso del algoritmo 1. Leer tempFahrenheit 2. $\text{TempCelsius} \leftarrow (\text{tempFahrenheit} - 32)/1.8$ 3. Mostrar tempCelsius 4. fin

Captura del processing y Resultado:

```

1 float tempFarhenheit = 5;
2 float tempCelsius = (tempFarhenheit - 32)/1.8;
3 println(tempCelsius);

```



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 (??1, ??1) , mientras que las de la caja de tesoro se halla en la posición (??2, ??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.

Análisis:

Datos de Entrada: Coordenadas de link, Coordenadas cofre
Datos de Salida: Distancia entre Link y tesoro
Proceso:
¿Quien debe realizar el proceso?
El programa informático o la computadora
¿Cual es el proceso que resuelve?
Calculamos las diferencias en las coordenadas x;y entre los dos puntos

Diseño:

Entidad que resuelve el problema: persona
Variables: <ul style="list-style-type: none"> - X1: float // almacena un valor decimal - Y1 : float // almacena un valor decimal - X2 : float // almacena un valor decimal - Y2 : float // almacena un valor decimal - CoordenadaX : float // almacena el resultado de un calculo - CoordenadaY : float // almacena el resultado de un calculo - Distancia : float // almacena el resultado de un calculo - Cofre : float // almacena un valor
Nombre del Algoritmo: distancia
Proceso del algoritmo <ol style="list-style-type: none"> 1. Leer x1 2. Leer y1 3. Leer x2 4. Leer y2 5. Cofre \leftarrow 25 6. CoordenadaX \leftarrow x2 - x1 7. CoordenadaY \leftarrow y2 - y1 8. Distancia \leftarrow ((CoordenadaX)²+ (CoordenadaY)²)² 9. Mostrar "La distancia es de:" + distancia 10. Si distancia <= cofre 11. Mostrar distancia "Premio Recibido" 12. fin

Captura del processing y Resultado:

```

1 float x1= 100 , y1 = 100 , x2= 200, y2= 400;
2 float cofre = 25;
3
4 void setup(){
5   size(750,500);
6 }
7
8 void draw() {
9   background(0);
10  float CoordenadaX = x2 - x1;
11  float CoordenadaY = y2 - y1;
12
13  float distancia = sqrt(pow(CoordenadaX, 2)+ pow(CoordenadaY, 2));
14  String textoDistancia = "La distancia es de:" + distancia;
15  println(textoDistancia);
16  if (distancia <= cofre) {
17    println("Premio Recibido");
18  }
19
20  println("Premio Recibido");
21  }
22  fill(#10F026);
23  ellipse(x1,y1,50,50);
24  fill(#F01810);
25  rect(x2,y2,50,50);
26  String coordenadas = "X1: " + mouseX + ", Y1: " + mouseY;
27  fill(255);
28  textSize(20);
29  textAlign(RIGHT, TOP);
30  text(coordenadas, width, 0);
31
32 void mouseMoved(){
33   x1=mouseX;
34   y1=mouseY;
35 }

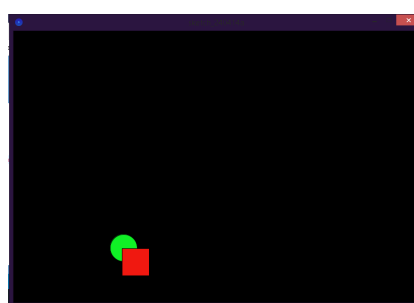
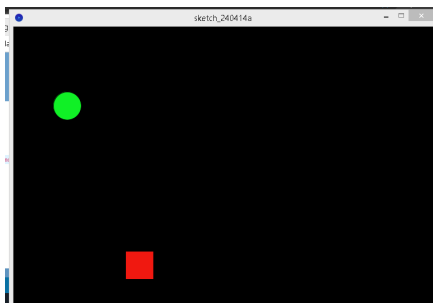
```

```

Premio Recibido
La distancia es de:7.0
Premio Recibido
La distancia es de:7.0
Premio Recibido

```

Consola Errores



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.

Análisis:

Datos de Entrada: Coeficientes de la ecuación cuadrática: a,b y c.

Datos de Salida: Raíces de la ecuación cuadrática

Proceso:

¿Quién debe realizar el proceso?

El programa informático o la computadora



¿Cual es el proceso que resuelve?

Calcular discriminante de la ecuacion cuadratica.

Diseño:

Entidad que resuelve el problema: persona

Variables:

- A: float // almacena un valor decimal
- B: float // almacena un valor decimal
- C: float // almacena un valor decimal
- Discriminante: float //almacena un valor de calculos

Nombre del Algoritmo encontrar_raices

Proceso del algoritmo

1. Leer a
2. Leer b
3. Leer c
4. $\text{Discriminante} \leftarrow b^2 - 4 \cdot a \cdot c$
5. Si (discriminante > 0) entonces
6. $\text{Raiz 1} \leftarrow (-b + (\text{discriminante})^{1/2}) / (2 \cdot a)$
7. $\text{Raiz2} \leftarrow (-b - (\text{discriminante})^{1/2}) / (2 \cdot a)$
8. Mostrar "las raices son" + raiz1+ y + raiz2
9. Si_no si (discriminante == 0) entonces
10. Mostrar "la raiz doble es:" + raiz
11. si_no
12. Mostrar "no hay raices reales"
13. fin

Captura del processing y Resultado:

```

1 float a=8,b=3,c=-10;
2
3 void setup(){
4     float discriminante = pow(b,2)-4*a*c;
5
6     if(discriminante>0){
7         float x1= (-b+sqrt(discriminante))/(2*a);
8         float x2= (-b-sqrt(discriminante))/(2*a);
9         println("Raices: " + x1 + " y " +x2);
10    }
11    else if (discriminante==0){
12        float x=-b / (2*a);
13        println("Raiz"+x);
14    }
15    else
16        println("No existen raices");
17 }
18
19

```

Se ha guardado el sketch.

Raices: 0.9461473 y -1.3211473

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.

Análisis:

Datos de Entrada: línea , esfera

Datos de Salida: bucle de la línea y círculo

Proceso:

¿Quién debe realizar el proceso?

El programa informático o la computadora

¿Cuál es el proceso que resuelve?

Que la línea del círculo empiece desde 1 y llega hasta la máxima posición otorgada, como no puede pasar, rebota y vuelve a su estado original

Diseño:

Entidad que resuelve el problema: lienzo
Variables: - Linea: entero // almacena valor entero - Esfera : entero // almacena valor entero
Nombre del Algoritmo: linea_en_movimiento
Proceso del algoritmo 1. Leer linea 2. Leer esfera 3. Ancholienzo \leftarrow 400 4. AltoLienzo \leftarrow 400 5. Para v \leftarrow 0 hasta alto incremento 1 6. Linea \leftarrow linea + esfera 7. Si ((linea \geq ancholienzo linea \leq 0)) 8. Esfera \leftarrow esfera * -1 9. fin_si 10. Mostrar linea 11. Dibujar linea esfera, linea,altolienzo, linea 12. Dibujar altolienzo/2 , linea + 40,80,80 13. fin

Captura del processing y Resultado:

```

1  int linea;
2  int esfera = 1;
3
4  void setup() {
5    size(400, 400);
6    linea = 200;
7  }
8
9  void draw() {
10   background(1);
11
12   for (int v = 0; v < 1; v++){
13     linea = linea + esfera;
14   }
15   if (linea >= height || linea <= 0){
16     esfera = esfera * -1;
17   }
18   println(linea);
19
20   stroke(255);
21   fill(0,255,0);
22   line(esfera, linea, width, linea);
23   ellipse(width/2 , linea + 40,80,80);
24 }

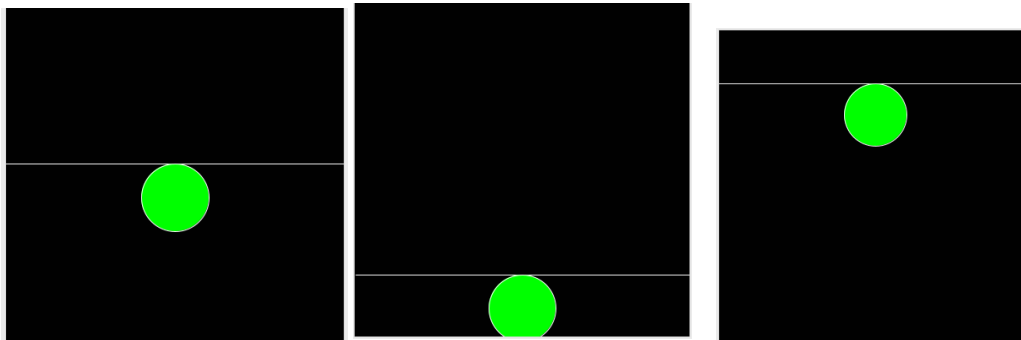
```

```

3
2
1
0
1
2
3
4
5
6
7
8
9
10

```

Consola



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: Dibujar en toda la intencional de un lienzo de (440, 420) rectángulos de idéntica medida.

Datos de Entrada: Rectángulos dibujados en el lienzo a base de las especificaciones dadas

Datos de Salida: Los Rectángulos dibujados en el lienzo a base de las especificaciones dadas

Proceso:

¿Quién debe realizar el proceso?

:El proceso es realizado por un programa como processing

¿Cuál es el proceso que resuelve?

Dibujar varios rectángulos en el lienzo con un tamaño específico respetando las distancias verticales y horizontales entre ellos, utilizando la variable **for** para dibujar los rectángulos

Diseño:

Entidad que resuelve el problema: Lienzo

Variables:

- **coordenadas:** float // almacena un valor de coordenadas cartesianas
- **ancho y la altura del lienzo:** size // almacena valores enteros de la posición
- **ancho, alto y la distancia:** int // almacena valores enteros de la posición

Nombre del Algoritmo : repetición_rectángulos

Proceso del algoritmo

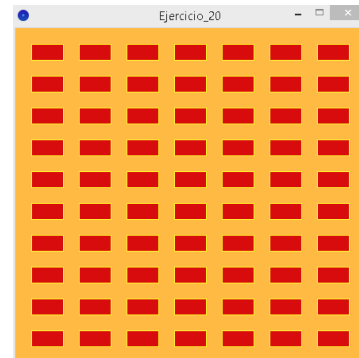
1. Leer La Altura, ancho y la distancia // determinando valores enteros
2. Leer el tamaño asignado (x, y) (440, 420)
3. Leer distancia
4. Leer ancho
5. Leer Altura
6. Leer Coordenadas de PVector
7. Para X coordenadas.x hasta ancho del lienzo (ancho + distancia)
8. Para Y coordenadas.y hasta la altura del lienzo (altura+distancia)
9. Dibujar rectángulo en (x,y,ancho,altura)
10. fin

Captura del processing y Resultado:

```

Ejercicio 20
1 PVector coordenadas;
2 int altura, ancho, distancia;
3
4 //altura , ancho y distancia esta relacionado al dibujo del rectangulo//
5
6 void setup(){
7   size(440,420);
8   distancia = 20;
9   ancho = 40;
10  altura = 20;
11  coordenadas = new PVector(distancia,distancia);
12 }
13
14 void draw(){
15   background(#FFBA44);
16   fill(#D80C0C);
17   stroke(#FFF929);
18   dibujar();
19 }
20
21 void dibujar(){
22   for(float x=coordenadas.x;x<width;x+=(ancho+distancia)){
23     for(float y= coordenadas.y;y<height;y+=(altura+distancia)){
24       rect(x,y,ancho,altura);
25     }
26   }
27 }

```



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:

Datos de Entrada: PuntoM,PuntoN,PuntoO,PuntoP,DISTANCIA

Datos de Salida: Una imagen que consiste en escalones con puntos en los bordes

Proceso:

¿Quien debe realizar el proceso?

El programa informático o la computadora

¿Cual es el proceso que resuelve?

El proceso que resuelve consiste en repetir while() para dibujar escalones y puntos.

Diseño:

Entidad que resuelve el problema: programa

Variables:

PuntoM,PuntoN,PuntoO,PuntoP: int // almacena un vector

Distancia : int // almacena un valor entero

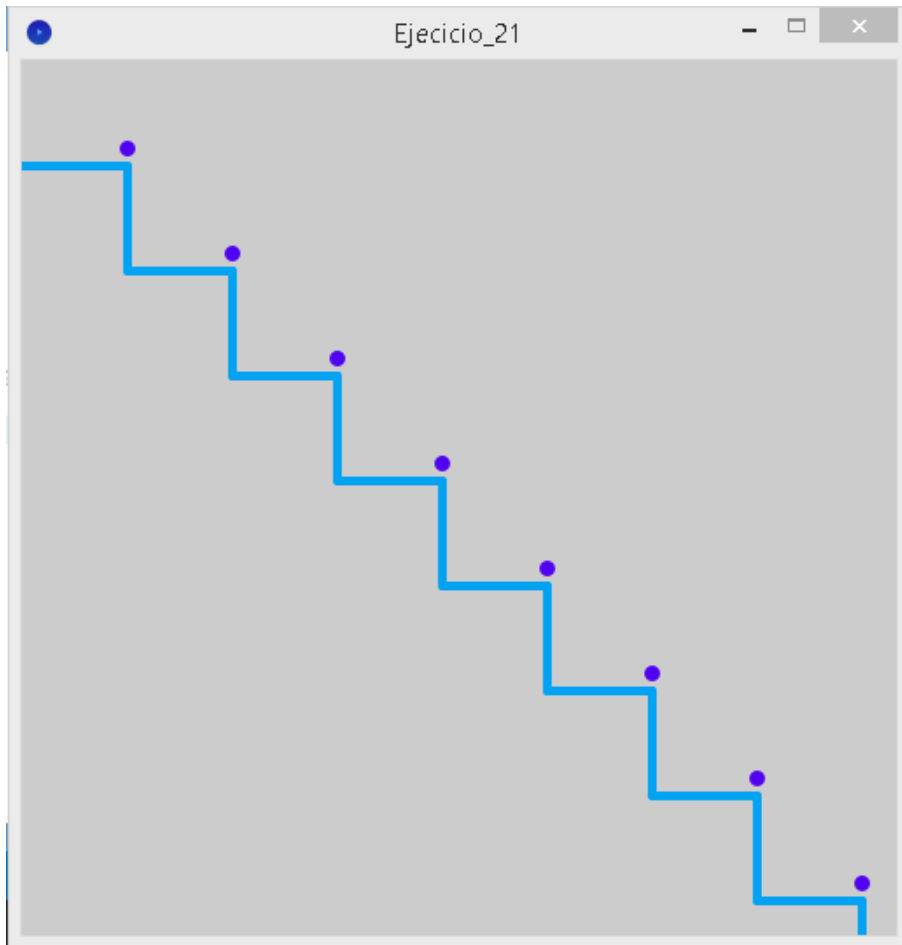
Nombre del Algoritmo: escalones_puntos

Proceso del algoritmo:

1. Ancho lienzo \leftarrow 500
2. Alto lienzo \leftarrow 500
3. Distancia \leftarrow 60
4. Mientras PuntoM.y sea menor o igual ancholienzo
5. Dibujar linea horizontal en PuntoM.y,PuntoN.x,PuntoN.y
6. Dibujar linea vertical en PuntoN.x,PuntoN.y,PuntoO.x,PuntoO.y
7. Dibujar crculo en (PuntoP.x,PuntoP.y)
8. PuntoM.x \leftarrow PuntoO.x
9. PuntoM.y \leftarrow PuntoO.y
10. fin

Captura del processing y Resultado:


```
1  int distancia;
2  PVector PuntoM,PuntoN,PuntoO,PuntoP;
3
4  void setup() {
5      size(500,500);
6      distancia=60;
7      PuntoM = new PVector(0,distancia);
8
9      while(PuntoM.y <= height){
10         escalera();
11         punto();
12         repeat();
13     }
14 }
15 void escalera(){
16     stroke(#01A2F2);
17     strokeWeight(5);
18     PuntoN = new PVector(PuntoM.x+distancia, PuntoM.y);
19     line(PuntoM.x, PuntoM.y,PuntoN.x,PuntoN.y);
20     PuntoO = new PVector(PuntoN.x,PuntoN.y+60);
21     line(PuntoN.x,PuntoN.y,PuntoO.x,PuntoO.y);
22 }
23 void punto(){
24     stroke(#5101F2);
25     strokeWeight(9);
26     PuntoP = new PVector(PuntoN.x,PuntoN.y-10);
27     point(PuntoP.x,PuntoP.y);
28 }
29
30 void repeat(){
31     PuntoM.x = PuntoO.x;
32     PuntoM.y = PuntoO.y;
33 }
```



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:

Datos de Entrada: líneas y círculos

Datos de Salida: círculos de colores aleatorios sobre una línea.

Proceso:

¿Quién debe realizar el proceso?

El programa en este caso es Processing

¿Cuál es el proceso que resuelve?

El lienzo establecido ocupa una serie de líneas de forma vertical de misma medida.



En cada línea se dibujan varios círculos con diversos colores a lo largo de la línea, respetando un distanciamiento. En las líneas se alternan de por medio que línea contendrá círculos y cual no, así sucesivamente

Diseño:

Entidad que resuelve el problema: programa processing

Variables:

-linex ,liney ,circley ,circlex,distancecircle :int // almacenan valores enteros
-Ancholiendo, alto lienzo : int // almacenan valores enteros

Nombre del Algoritmo: circulos_repetidos

Proceso del algoritmo

1. Ancho lienzo \leftarrow 600
2. Alto lienzo \leftarrow 600
3. Linex \leftarrow 0
4. Liney \leftarrow 100
5. Distancecircle \leftarrow 30
6. Circley \leftarrow 85
7. do
8. circlex \leftarrow distancecircle
9. do
10. Dibujar line (linex,liney,ancholiendo,liney)
11. Dibujar circle(circlex,circley,30)
12. Circlex \leftarrow circlex + distancecircle*2
13. Fin_do
14. While(circlex < ancholiendo)
15. Liney \leftarrow liney + 100
16. Cricley \leftarrow circley + 200
17. Fin_do
18. While (liney < altoliendo)
19. fin

Captura del processing y Resultado:

```
1 void setup(){
2   size(600,600);
3   int linex = 0 ,liney = 100,circley = 85,distancecircle = 30;
4   do{
5     int circlex = distancecircle;
6     do{
7       stroke(#09F2CB);
8       line(linex,liney,width,liney);
9       fill (random(255), random(255), random(255));
10      stroke(0);
11      strokeWeight(5);
12      circle(circlex,circley,30);
13      circlex += distancecircle*2;
14      println(circlex); //va sumando cada 30 unidades//
15    }while(circlex < width);
16    liney += 100;
17    circley +=200;
18  }while(liney < height);
19 }
20 }
21 }
```

450.0

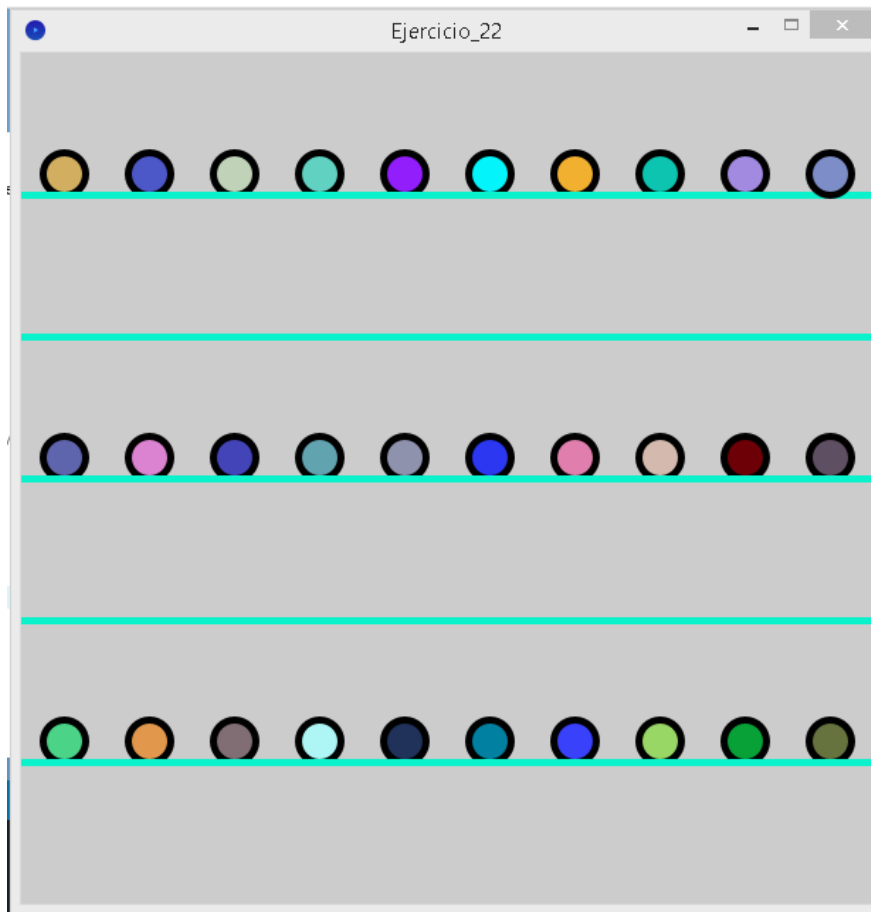
510.0



570.0

630.0



Consola



	<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>	
---	--	---

Fuentes bibliográficas

Se deben enunciar las fuentes (apuntes de la materia, páginas web, videos de youtube, libro

- Tutorial de processing: matemáticas

([PROCESSING: 4.4 Matemáticas I \[En español\] - YouTube](#))

- Tutorial de proccesing : font text

(https://processing.org/reference/textFont_.html)

-Colores RGB con codigos

https://www.rapidtables.org/web/color/RGB_Color.html

-Calculadora

<https://mathsolver.microsoft.com/es/solveproblem/3%20%3C%20%206?r=bi&ref=bi>

-Perimetro y area:

https://content.nroc.org/DevelopmentalMath/U07L2T2_RESOURCE/U07_L2_T2_text_final_es.html

-Perimetro y area de un rectangulo:

<https://www.neurochispas.com/wiki/perimetro-y-area-de-un-rectangulo-formulas-y-ejercicios/>

-Sacar hipotenusa :

<https://www.unprofesor.com/matematicas/como-sacar-la-hipotenusa-de-un-trianguulo-6083>.

-Aplicar Raiz cuadrada en processing:

https://processing.org/reference/quad_.html