|  |
| --- |
| **FUNDAMENTOS DE LA PROGRAMACION ORIENTADO A OBJETOS**  **Fpoo 2024** |

**Trabajo práctico**

**N°1**

**Apellido y Nombre**

Flores Ruiz Rubén Claudio

**Profesor:**

Mg. Ing. Ariel Alejandro Vega

1

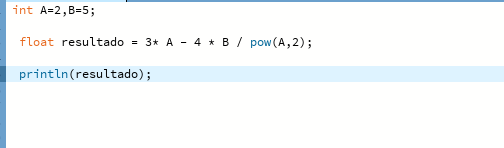
Evaluar(obtener resultado) la siguiente expresión para A = 2 y B = 5

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

6-(4\*B/4)

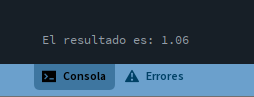
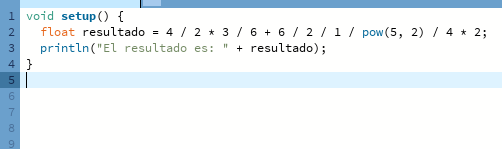
6-5 1

Resultado 1



2 Escribir las siguientes expresiones algebraicas como expresiones algorítmicas (en su forma aritmética dentro del algoritmo). En este caso no se pide evaluarlas ni programarlas

4 / 2 \* 3 / 6 + 6 / 2 / 1 / 5 ^ 2 / 4 \* 2

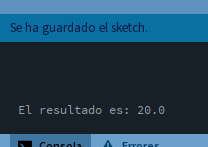
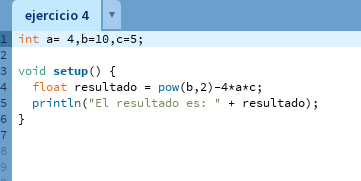


4

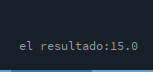
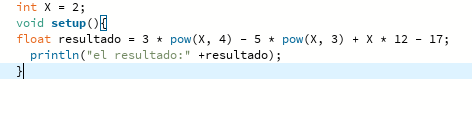
Evaluar las siguientes expresiones aritméticas, para lo cual indicar en el caso de las variables, el valor indicado. Luego escribirlas como expresiones algebraicas.

a) b ^ 2 – 4 \* a \* c

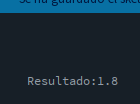
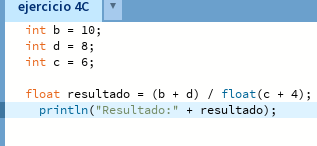
a= 3, b=10, c=5



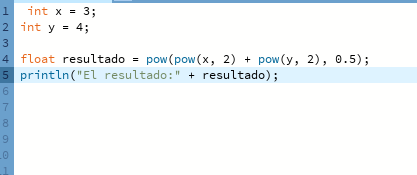
b) 3 \* X ^ 4 – 5 \* X ^ 3 + X 12 – 17 X=2



c) (b + d) / (c + 4) b =10 d =8 c=6



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

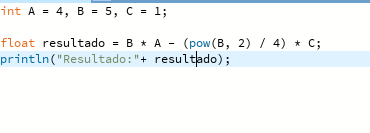
 

5

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

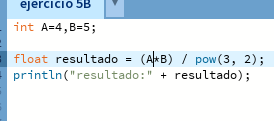
A :

B \* A – B ^ 2 / 4 \* C

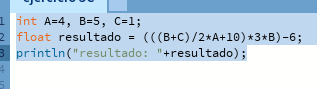
 

B:

(A \* B) / 3 ^ 2

C: (((B + C) / 2 \* A + 10) \* 3 \* B) – 6

6

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

R1 = y+z

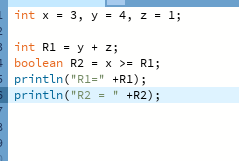
R2 = x >= R1

--------------------

R1= 4+1 = 5

R2= 3 >= R1

Falso

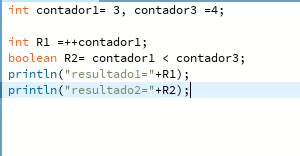
7

contador1=3, contador3=4,

evaluar elresultado de

R1 = ++contador1 : 4

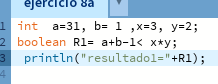
R2 = contador1 < contador2 : false

8

Para a=31, b=-1; x=3, y=2,

evaluar el resultado de a+b-1 < x\*y

9

Para x=6, y=8,

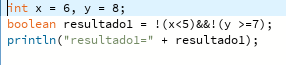
evaluar el resultado de ! (x<5) &&!(y>=7)

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

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

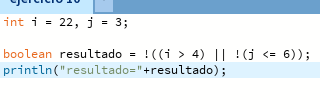
falso && falso

falso



10

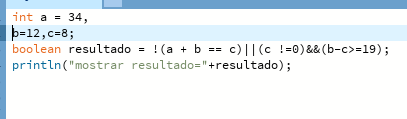
Para i=22 ,j=3, evaluar el resultado de

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

11

Para a=34, b=12,c=8,

evaluar el resultado de !(a+b==c) || (c!=0)CC(b-c>=19)



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

**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: nombre 🡪 cadena

Datos de Salida :saludo 🡪 cadena de texto

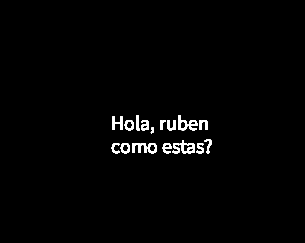
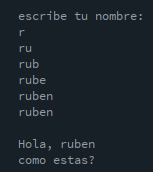
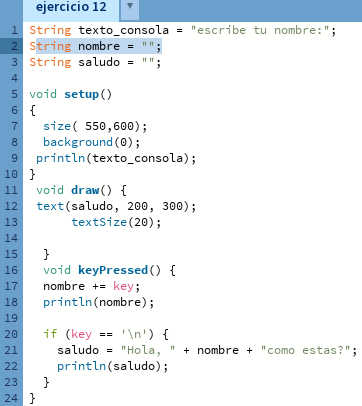
Proceso:

**¿Quien debe realizar el proceso?:** processing

**¿Cual es el proceso que resuelve?:** dar el saludo con el nombre dictado en el teclado ,mostrando un saludo en pantalla

**Diseño:**

|  |
| --- |
| **Entidad que resuelve el problema: processing\_algoritmo** |
| **Variables:**  **nombre: string 🡪** almacena el nombre ingresado  **mensaje: string 🡪** almacena una cadena de caracteres |
| **Nombre del Algoritmo:** Saludo |



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🡪 int

Datos de Salida :perímetro,area 🡪 int

Proceso:

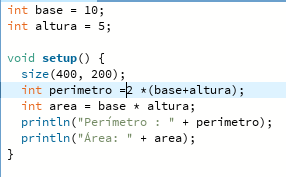
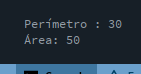
**¿Quien debe realizar el proceso?:** processing\_la calculadora

**¿Cual es el proceso que resuelve?:** dado base y altura calcular perímetro y área

P=2(base +altura) y A=base ⋅ altura

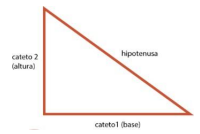
**Diseño:**

|  |
| --- |
| **Entidad que resuelve el problema: usuario** |
| **Variables:**  **base: int🡪**  **altura: int🡪**  **perimetro:int 🡪**  **area: int🡪** |
| **Nombre del Algoritmo:** Saludo |
| 1. **Dar base** 2. **Dar altura**   **El perimetro es: 2\*(base + altura)**   1. **area : base \* altura** 2. **Calcular el perímetro y área del triangulo** 3. **Mostrar perímetro y Area en la consola** 4. **fin** |

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 y 2🡪 int

Datos de Salida :hipotenusa 🡪 int

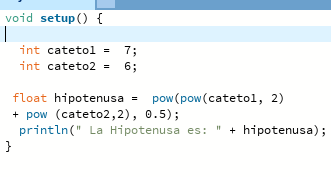
Proceso:

**¿Quien debe realizar el proceso?:** processing\_la calculadora

**¿Cual es el proceso que resuelve?:** dado dos catetos hallar la hipotenusa

**Diseño:**

|  |
| --- |
| **Entidad que resuelve el problema: programa** |
| **Variables:**  **Cateto1: int🡪 almacena valores enteros**  **Cateto2: int🡪 almacena valor entero**  **hipotenusa:float 🡪 almc valor de la operacion** |
| **Nombre del Algoritmo:** hallar\_hipotenusa |
| 1. **inicio** 2. **leer cateta1 y 2** 3. **hipotenusa = ← (a^2 + b^2 ) ^(0.5) pow(pow(cateto1, 2) + pow (cateto2,2), 0.5** 4. **fin** |

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: numeroA,numeroB🡪 int

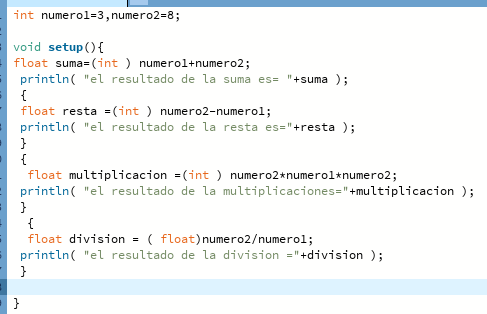
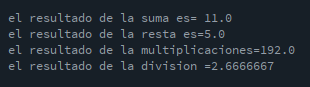
Datos de Salida :operaciones de suma,resta, multiplicación,division 🡪 int

Proceso:

**¿Quien debe realizar el proceso?:** processing\_la calculadora

**¿Cual es el proceso que resuelve?:** **Diseño: da**dado 2 numeros hallar, el resultado de una suma,resta,multiplicación y división

|  |
| --- |
| **Entidad que resuelve el problema: el programa** |
| **Variables:**  **Numero1: int🡪 almacena valores enteros**  **Numeor1 int🡪 almacena valor entero**  **suma:int 🡪 almc valor de la operación**  **resta: int 🡪 almacena valor de la operación**  **multiplicacion:int 🡪 almc valor de la operación**  **division:int 🡪 almc valor de la operacion** |
| **Nombre del Algoritmo:** hallar\_el\_resultado\_de\_la\_operaciones |
| 1. **inicio** 2. **leer numero1** 3. **leer numero2** 4. **calcular el resultado de la suma** 5. **calcular el resultado de la resta** 6. **calcular el resultado de la multiplicación** 7. **Calcular el resultado de la división** 8. **Dar resultados en la consola** 9. **fin** |

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: grados fahrenheit🡪 int

Datos de Salida : grados celcius 🡪 int

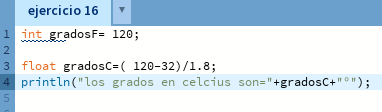
Proceso:

**¿Quien debe realizar el proceso?:** processing\_la calculadora

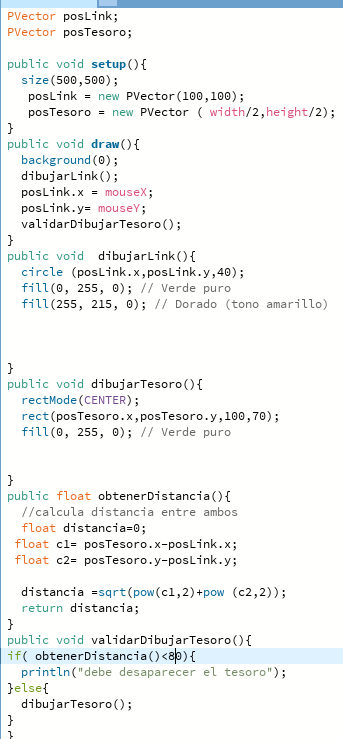
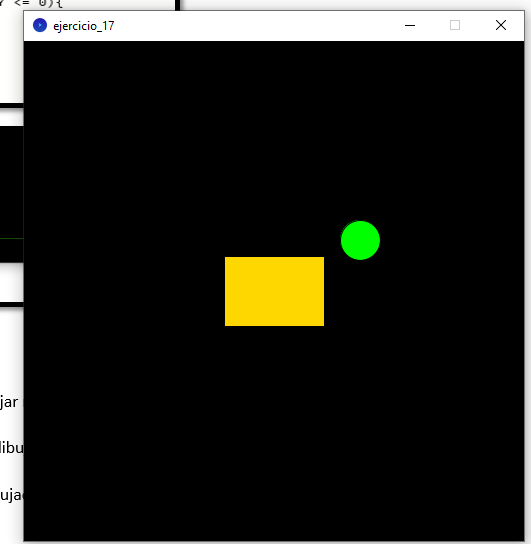
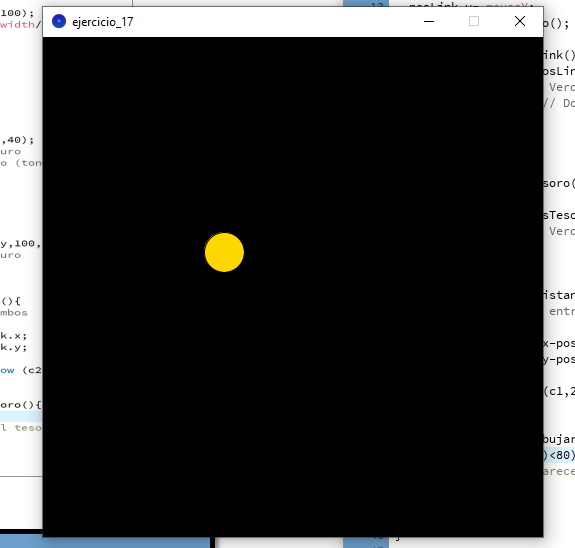
**¿Cual es el proceso que resuelve?:** dado los grados Fº convertirlos a celcius mediante la formula dada

**Diseño:**

|  |
| --- |
| **Entidad que resuelve el problema: usuario** |
| **Variables:**  **Grados farenheiht: int🡪 almacena valores de los grados Fº**  **Hallar grados celcius :int 🡪 almc valor de la operación para hallar ºC** |
| **Nombre del Algoritmo:** hallar\_hipotenusa |
| 1. **inicio** 2. **leer grados F** 3. **usar la formular para hallar C** 4. **Calcular 🡨(gradosF-32)/1.8** 5. **Mostrar resultado** 6. **Fin** |



-----------------------------------------------------------------------------------------------------------------------------------17 link tesoro

**Análisis:**

Datos de Entrada: posLink,PosTesoro🡪 PVector

Datos de Salida :hallar catetos,Calcular distancia 🡪 float

Proceso:

**¿Quien debe realizar el proceso?:** el programa

**¿Cual es el proceso que resuelve?:**hacer que un rectángulo “tesoro” desaparezca al acercarse con link “circulo”,calcular la distancia entre ambos sabiendo los catetos

**Diseño:**

|  |
| --- |
| **Entidad que resuelve el problema: usuario** |
| **Variables:** |
| **Nombre del Algoritmo:**Hallar\_tesoro |
| 1. Inicio 2. posLink🡪 pvector 3. posTesoro🡪pvector 4. lienzo🡪 500,500 5. **dibuja**r link 🡪circle (posLink.x,posLink.y,40) 6. **asignar** valores para link en Mause x 7. **asigna**r valores para link en Mause y 8. **dibuja**r tesoro🡪 rect(posTesoro.x,posTesoro.y,100,70) 9. *coordenada X ← x2 - x1* 10. *coordenada Y ← y2 – y1* 11. Hallar distancia sqrt(pow(c1,2)+pow (c2,2)) 12. ***Si*** *distancia< 80* ***entonces*** 13. **deb**e “desaparecer el tesoro” 14. mostrar 15. ***fin*** |

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: numero= int

Datos de Salida: Raíces de la ecuación cuadrática

Proceso:

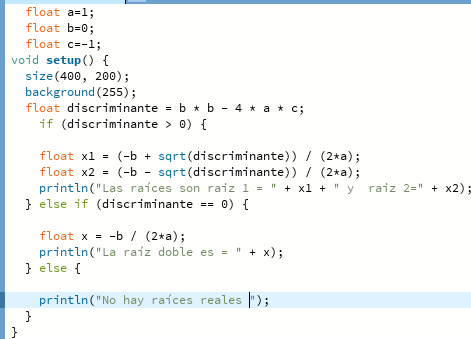
**¿Quién debe realizar el proceso?:** el programa

**¿Cuál es el proceso que resuelve?:** Calcular el discriminante para hallar las raíces en una ecuación cuadratica

**Diseño:**

|  |
| --- |
| **Entidad que resuelve el problema: el programador** |
| **Variables:**   * **a : float //** almacena un valor decimal * **b : float //** almacena un valor decimal * **c : float //** almacena un valor decimal * **discriminante: float** //almacena el valor de la operacion |

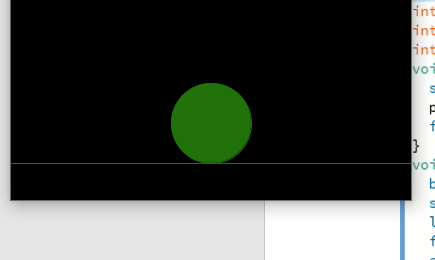
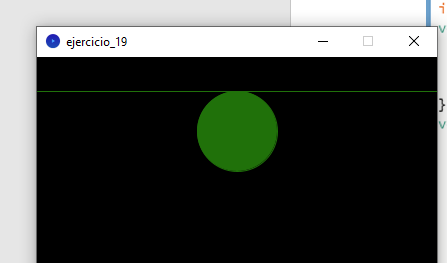
|  |
| --- |
| **Nombre del Algoritmo:** hallar\_raices |
| **Proceso del algoritmo:**   1. *inicio* 2. *Leer a* 3. *Leer b* 4. *Leer c* 5. *Size 🡨400,200* 6. *discriminante ← b \* b - 4 \* a \* c* 7. ***si*** (discriminante > 0) **entonces** 8. *x1 ← (-b + (discriminante))^0.5 /(2\*a)* 9. *X2 ← (-b - (discriminante))^0.5 /(2\*a)* 10. *mostrar “las raíces son raíz 1 = ” + x1 + “ y raíz 2 = ” + x2* 11. **si\_no** (discriminante == 0) **entonces** 12. **si** raiz *← -b / (2\*a)* 13. **mostrar** “la raíz doble es =“ + raiz 14. **si\_no** 15. mostrar “no hay raíces reales” 16. fin |





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 figura

20

**Análisis**

**Descripción del problema:** dibujar rectángulos usando estructuras iterativas

Datos de entrada: rectángulos dibujados con datos cargados de, ancho, alto y distancia entre el rectángulo

Datos de salida: rectángulos dibujados en el lienzo con bucle en X y Y respetando la distancia entre cada uno

Procesos:

¿Quién debe realizar el proceso? El proceso será realizado por el programa processing

¿Cuál es el proceso que resuelve? dibuja un bucle de rectángulos, en un lienzo cuyo tamaño es dado, mantendrá una distancia entre ellos, en forma vertical “Y” y horizontal “X”

Diseño

|  |
| --- |
| Entidad: Lienzo |
| Variables:  Cordenadas:cordenadas  Alto,ancho y distRec : enteros  Tamaño lienzo: enteros |

|  |
| --- |
| Nombre del algoritmo: Rectangulos\_Bucle |
| Proceso del algoritmo   1. Tamaño lienzo:440,420 2. DistRec: 20 3. Ancho 40 4. Alto 20 5. Para X cordenadas.x,**hasta** ancho lienzo,**con paso**(distRec+Ancho) 6. Para Y cordenadas.y,**hasta** alto lienzo,**con paso**(alto+distRec) 7. Dibujar un rectángulo en(x,y,ancho,alto) 8. mostrar 9. FIN |

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

**Análisis:**

Datos de Entrada: valores de x,y ancho escalon,alto escalon🡪 int

Datos de Salida : un bucle que dibuje una escalera con puntos en los extremos del escalon

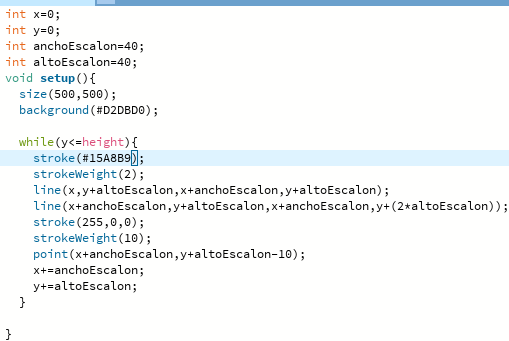
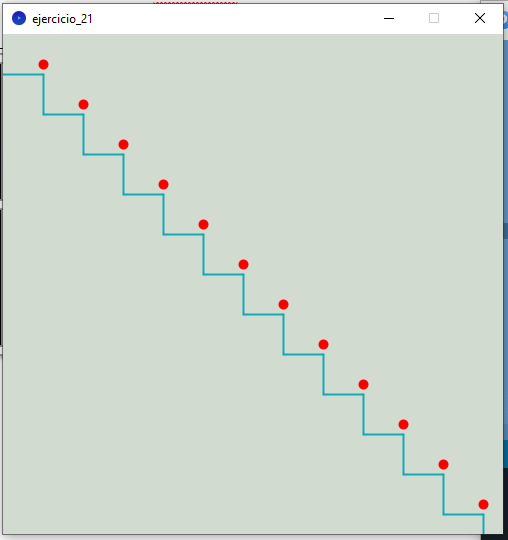
Proceso:

**¿Quien debe realizar el proceso?:** processing\_el algoritmo

**¿Cual es el proceso que resuelve?:** dado valores a X y E iniciar un bucle while que dibuje escalones

**Diseño:**

|  |
| --- |
| **Entidad que resuelve el problema: programa** |
| **Variables:**  **x🡪 cordenadas de tipo entero**  **y🡪cordenadas de tipo entero**  **ancho🡪tamaño de escalon en X**  **alto🡪 alto del escalon en Y** |
| **Nombre del Algoritmo:** Escalones\_punto |
| 1. Inicio 2. Tamaño lienzo 3. Colo lienzo 4. Color linea 5. Grosor linea 6. Dibujar una linea que forme la parte superior del escalon 7. Dibujar una linea horizontal que dibuje la parte derecha del escalon 8. Establecer el color del punto 9. Establecer el grosor del punto 10. Incrementar cordenadas horizontales 11. Incrementar cordenadas verticales 12. fin |

22

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

Análisis:

Datos de Entrada: valores de la linea, y circulo

Datos de Salida: el lienzo dividido en 6 partes iguales ,en ellas una hilera en el eje X de círculos con colores random

Proceso:

¿Quien debe realizar el proceso?: el programa

¿Cual es el proceso que resuelve?: un lienzo divido en 6 partes iguales en el eje Y con ciruculos en el eje x con una separación especifica en la linea x, y usando colores random

|  |
| --- |
| Entidad que resuelve el problema: processing |
| Variables:   * valores de Y : int //almacena un valor entero * distanciaLineas: int //almacena un valor entero * anchoLienzo, altoLienzo: int //almacenan valores enteros * color lienzo //almacena color del lienzo |

|  |
| --- |
| **Nombre del Algoritmo: l**ienzo\_dividido |
| **Proceso del algoritmo:**   1. *Inicio* 2. *Valor de Y= int* 3. *Valor de Y distancia* 4. *anchoLienzo = 600* 5. *altoLienzo = 600* 6. *asignar entero a Y* 7. *color fondo* *(#BCBCBC)* 8. ***dibujar*** una linea (0,y+separacionLineas,**width**,y+separacionLineas) 9. ***mientras*** Y sea menor o igual que la altura 10. ***dar***  valores de color tipo random 11. ***dibujar*** un circulo que repita en el eje x (i\*56,80,40,40); 12. ***fin*** |

