|  |  |  |
| --- | --- | --- |
| Texto  Descripción generada automáticamente con confianza media | TECNICATURA UNIVERSITARIA EN DISEÑO INTEGRAL DE VIDEOJUEGOS  FACULTAD DE INGENIERÍA  Universidad Nacional de Jujuy |  |

*Profesor:*

*Mg. Ing. Ariel Alejandro Vega*

*Año 2024*

Trabajo Práctico N°1

Angelo Carla TUV000563

**FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS**

Indice

**Trabajo Práctico N°1: Operadores – Metodología de Programación**

*Sección Expresiones aritméticas y lógicas*

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

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

**Resolución:**

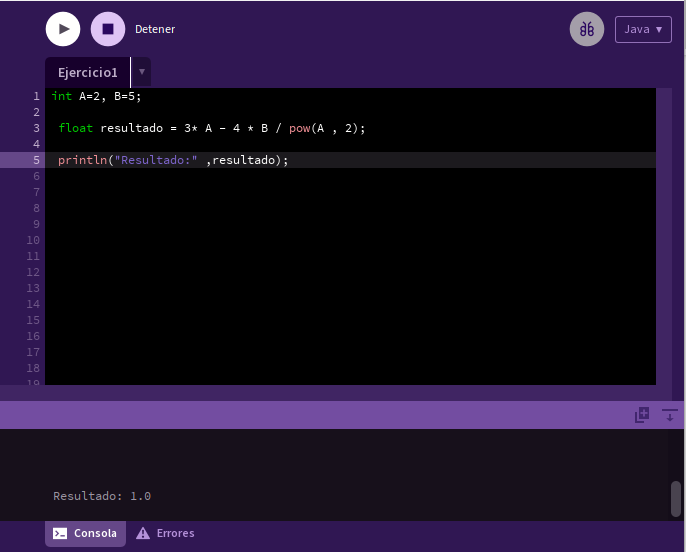
( 3\* 2 ) – ( 4 \* 5 / 2 ^ 2 )

6 - ( 20 / 4 )

6 – 5

**1**

**Processing:**



**Ejercicio 2:** Evaluar la siguiente expresión

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

**Resolución:**

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

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

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

1 + ( 3 / 25 / 4 \* 2 )

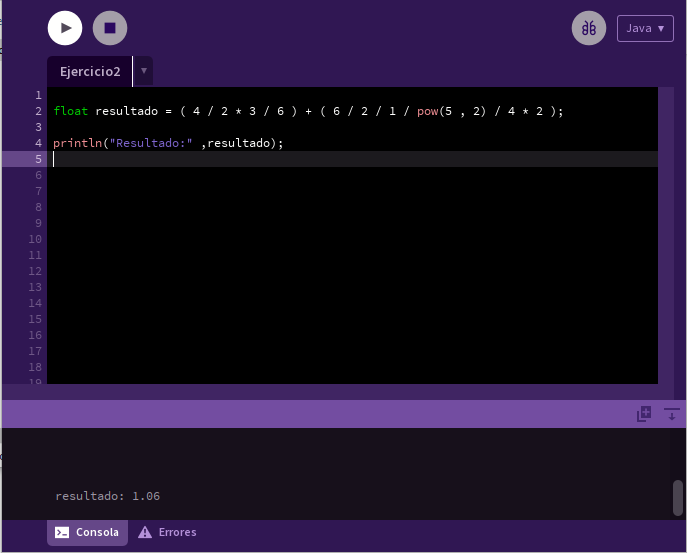
1 + ( 0.12 / 4 \* 2 )

1 + ( 0.03 \* 2 )

1 + 0.06

**1.06**

**Processing:**

****

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

a) b ^ 2 – 4 \* a \* c

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

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

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

**Resolución:**

1. b ^ 2 – 4 \* a \* c

( 2 ^ 2 ) − ( 4 \* 1\* 3 )

4 – ( 4 \* 3 )

4 – 12

**-8**

**𝑏² − 4. 𝑎. c**

1. 3 \* X ^ 4 – 5 \* X ^ 3 + X 12 – 17

( 3 \* 2 ^ 4 ) – ( 5 \* 2 ^ 3 ) + ( 2 \* 12 ) – 17

( 3 \* 16 ) – ( 5 \* 8 ) + ( 2 \* 12 ) – 17

48 – 40 + 24 – 17

8 + 24 – 17

32 – 17

**15**

**3x ⁴ - 5x ³ + x12 – 17**

1. (b + d) / (c + 4)

( 2 + 4 ) / ( 3 + 4 )

6 / 7

**o.857**

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

( 2 ^ 2 + 1 ^ 2) ^ ( 1 / 2 )

( 4 + 1 ) ^ ( 1 / 2 )

5 ^ ( 1 / 2 )

**2.236**

**(x² + y²)¹/2**

**Processing:**

**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

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

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

**Resolución:**

a) ( 5 \* 4 ) – ( 5 ^ 2 / 4 \* 1 )

20 – ( 25 / 4 \* 1 )

20 – ( 6.25 \* 1 )

20 – 6.25

**13.75**

b) ( 4 \* 5 ) / 3 ^ 2

20 / 3 ^ 2

20 / 9

**2.2222**

c) ( ( ( 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

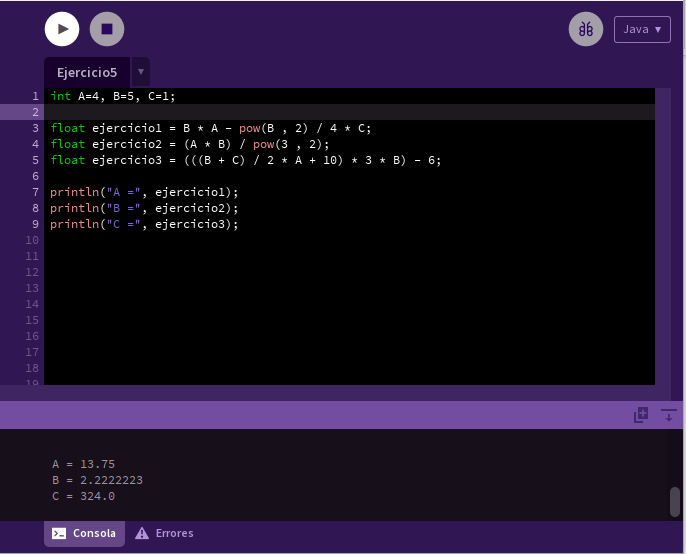
( 22 \* 3 \* 5 ) – 6

66 \* 5 – 6

330 – 6

**324**

**Processing:**



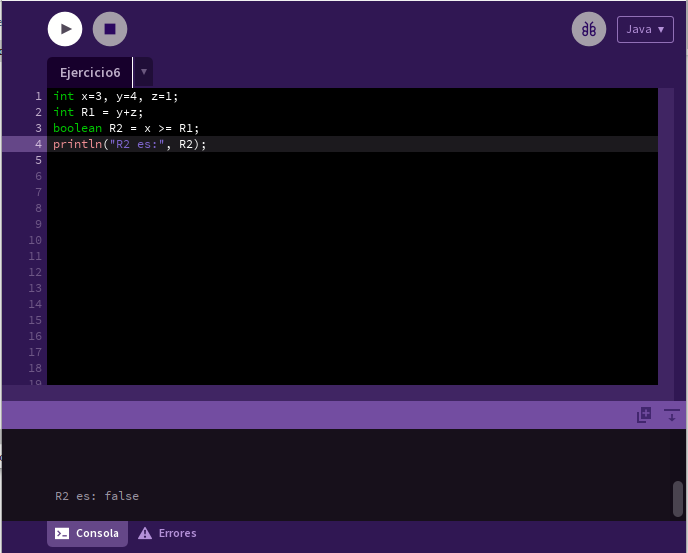
**Ejercicio 6:** Para x=3, y=4; z=1, evaluar el resultado de

R1 = y+z

R2 = x >= R1

**Resolución:**

**Processing:**



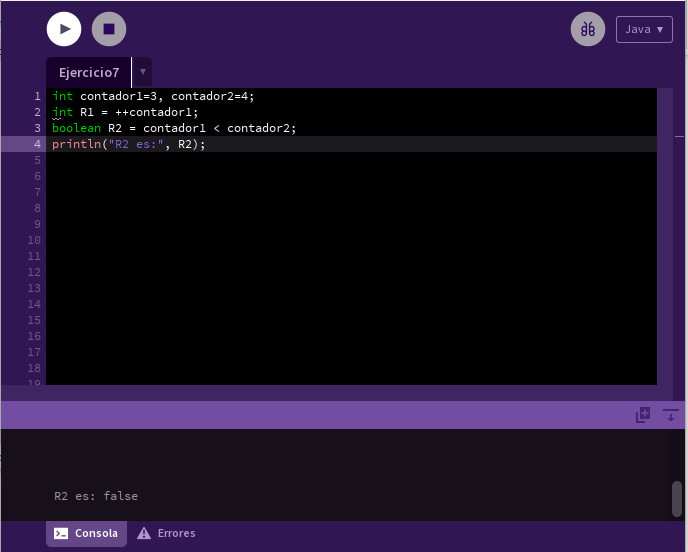
**Ejercicio 7:** Para contador1=3, contador2=4, evaluar el resultado de

R1 = ++contador1

R2 = contador1 < contador2

**Resolución:**

**Processing:**

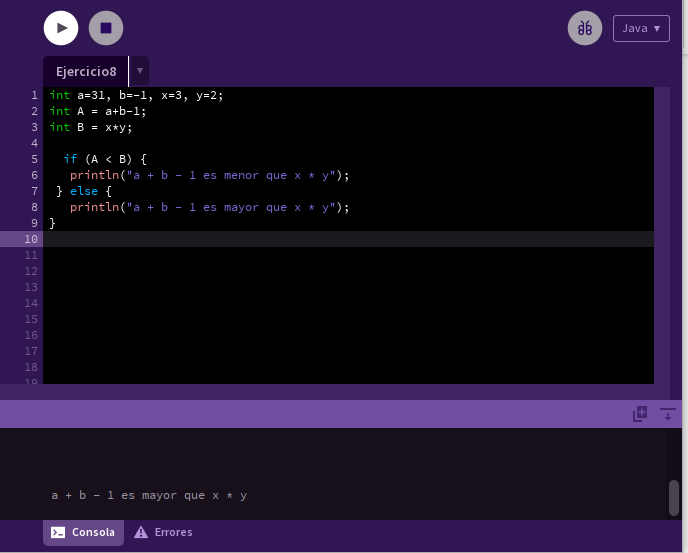


**Ejercicio 8:** Para a=31, b=-1; x=3, y=2, evaluar el resultado de

a+b-1 < x\*y

**Resolución:**

**Processing:**

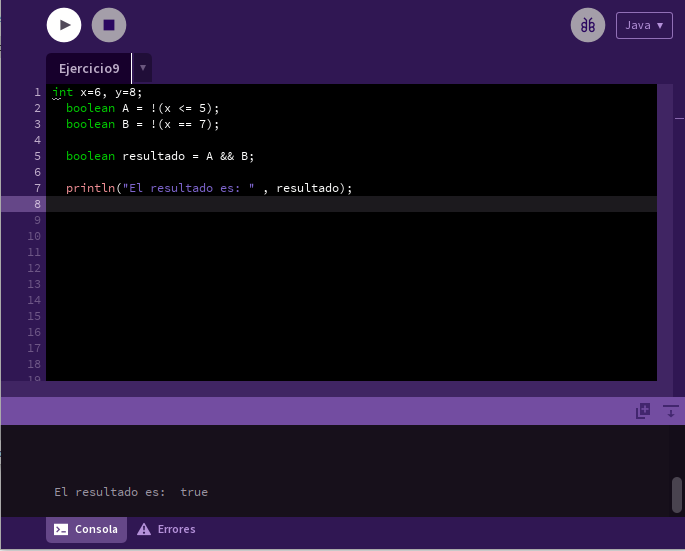


**Ejercicio 9:** Para x=6, y=8, evaluar el resultado de

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

**Resolución:**

**Processing:**

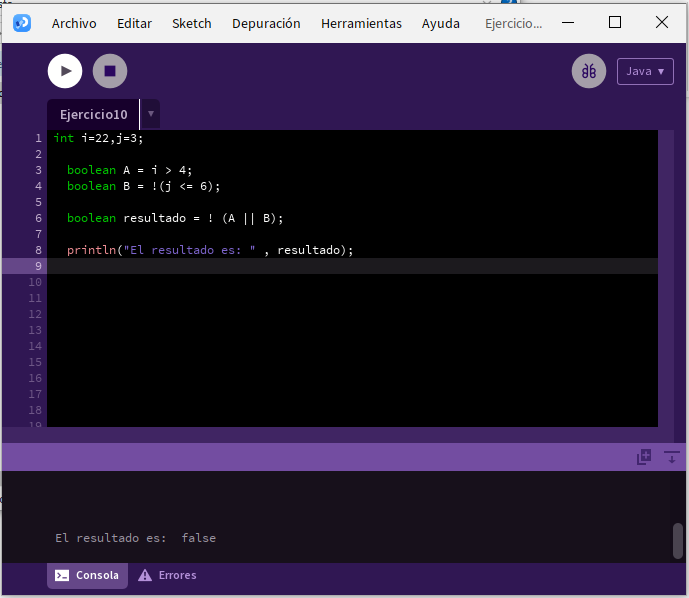


**Ejercicio 10:** Para i=22,j=3, evaluar el resultado de

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

**Resolución:**

**Processing:**

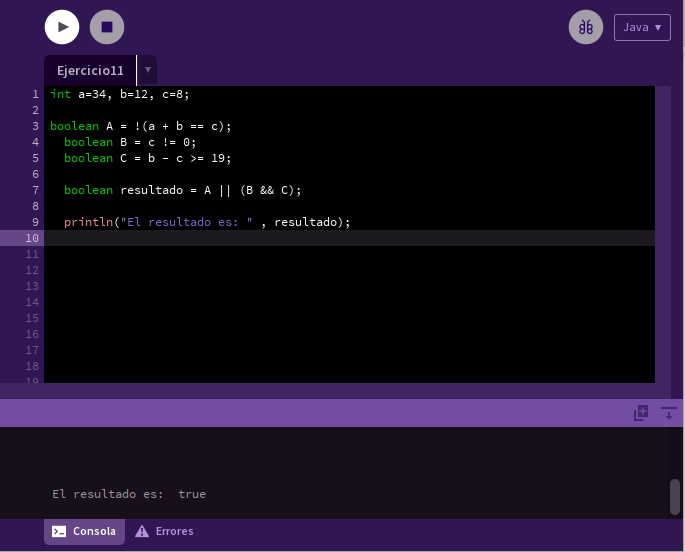


**Ejercicio 11:** Para a=34, b=12,c=8, evaluar el resultado de

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

**Resolución:**

**Processing:**



*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

**Resolución:**

*Definición del problema:* A partir de un nombre, saludara con el nombre indicado

**Análisis:**

*Datos de Entrada:* Un nombre

*Datos de Salida:* Un saludo con el nombre indicado

*Proceso:*

**Diseño:**

|  |
| --- |
| ***ENTIDAD QUE RESUELVE EL PROBLEMA:*** El algoritmo |
| ***VARIABLES:***  nombre: string //almacena el nombre |
| ***NOMBRE ALGORITMO:*** Ejercicio12  ***PROCESO DEL ALGORITMO:***  inicio  nombre ⭠ “Lulu” //ejemplo de nombre  saludo ⭠ “Hola” + nombre  *mostrar* saludo  fin |

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

**Resolución:**

*Definición del problema:*

**Análisis:**

*Datos de Entrada:*

*Datos de Salida:*

*Proceso:*

**Diseño:**

|  |
| --- |
| ***ENTIDAD QUE RESUELVE EL PROBLEMA:*** |
| ***VARIABLES:*** |
| ***NOMBRE ALGORITMO:***  ***PROCESO DEL ALGORITMO:*** |

**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

**Resolución:**

*Definición del problema:*

**Análisis:**

*Datos de Entrada:*

*Datos de Salida:*

*Proceso:*

**Diseño:**

|  |
| --- |
| ***ENTIDAD QUE RESUELVE EL PROBLEMA:*** |
| ***VARIABLES:*** |
| ***NOMBRE ALGORITMO:***  ***PROCESO DEL ALGORITMO:*** |

**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

**Resolución:**

*Definición del problema:*

**Análisis:**

*Datos de Entrada:*

*Datos de Salida:*

*Proceso:*

**Diseño:**

|  |
| --- |
| ***ENTIDAD QUE RESUELVE EL PROBLEMA:*** |
| ***VARIABLES:*** |
| ***NOMBRE ALGORITMO:***  ***PROCESO DEL ALGORITMO:*** |

**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 (𝑥1, 𝑦1) , mientras que las de la caja de tesoro se halla en la posición (𝑥2, 𝑦2). 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 lLnk con un Circulo, y al tesoro con un cuadrado. Además, mueva a Link mediante el mouse.

**Resolución:**

*Definición del problema:*

**Análisis:**

*Datos de Entrada:*

*Datos de Salida:*

*Proceso:*

**Diseño:**

|  |
| --- |
| ***ENTIDAD QUE RESUELVE EL PROBLEMA:*** |
| ***VARIABLES:*** |
| ***NOMBRE ALGORITMO:***  ***PROCESO DEL ALGORITMO:*** |

**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

**Resolución:**

*Definición del problema:*

**Análisis:**

*Datos de Entrada:*

*Datos de Salida:*

*Proceso:*

**Diseño:**

|  |
| --- |
| ***ENTIDAD QUE RESUELVE EL PROBLEMA:*** |
| ***VARIABLES:*** |
| ***NOMBRE ALGORITMO:***  ***PROCESO DEL ALGORITMO:*** |

**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

**Resolución:**

*Definición del problema:*

**Análisis:**

*Datos de Entrada:*

*Datos de Salida:*

*Proceso:*

**Diseño:**

|  |
| --- |
| ***ENTIDAD QUE RESUELVE EL PROBLEMA:*** |
| ***VARIABLES:*** |
| ***NOMBRE ALGORITMO:***  ***PROCESO DEL ALGORITMO:*** |

**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

**Resolución:**

*Definición del problema:* Dibujar rectángulos a lo largo y ancho del lienzo utilizando la estructura iterativa for

**Análisis:**

*Datos de Entrada:* el alto y ancho de los rectángulos y la distancia entre ellos que es de 20 pixeles

*Datos de Salida:* Los rectángulos dibujados

*Proceso:* Dibujar los rectángulos con estructura iterativa

**Diseño:**

|  |
| --- |
| ***ENTIDAD QUE RESUELVE EL PROBLEMA:*** El algoritmo |
| ***VARIABLES:***  coordRect: PVector  distRect, anchoLienzo, altoLienzo, ancho, alto : entero |
| ***NOMBRE ALGORITMO:*** dibujar\_rectangulos  ***PROCESO DEL ALGORITMO:***  Inicio  anchoLienzo ***⭠*** 440  altoLienzo ***⭠*** 420  distRect ***⭠*** 20  ancho ***⭠*** 40  alto ***⭠*** 20  coordRect***⭠*** PVector(distRect,distRect)  rectangulos ***⭠***  *para.x* ***⭠*** coordRect.x hasta anchoLienzo con paso (ancho+distRect) hacer  *para.y****⭠*** coordRect.y hasta altoLienzo con paso (alto+distRect) hacer  *dibujar* rectángulo (x, y) con dimensiones ancho y alto  *fin\_para*  *fin\_para*  *mostrar* rectangulos  fin |

**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

**Resolución:**

*Definición del problema:*

**Análisis:**

*Datos de Entrada:*

*Datos de Salida:*

*Proceso:*

**Diseño:**

|  |
| --- |
| ***ENTIDAD QUE RESUELVE EL PROBLEMA:*** |
| ***VARIABLES:*** |
| ***NOMBRE ALGORITMO:***  ***PROCESO DEL ALGORITMO:*** |

**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.

**Resolución:**

*Definición del problema:*

**Análisis:**

*Datos de Entrada:*

*Datos de Salida:*

*Proceso:*

**Diseño:**

|  |
| --- |
| ***ENTIDAD QUE RESUELVE EL PROBLEMA:*** |
| ***VARIABLES:*** |
| ***NOMBRE ALGORITMO:***  ***PROCESO DEL ALGORITMO:*** |

Conclusión

Párrafos de las conclusiones

Fuentes bibliográficas

Se deben enunciar las fuentes (apuntes de la materia, páginas web, videos de youtube, libro (nombre, autores, año), etc)