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

Trabajo Práctico N°1

*FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS*

*FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS*



Apellido y Nombre

Córdoba Agustina Ayelén

LU / TUV000471

*Profesores:*

*Mg. Ing. Ariel Alejandro Vega*

*Año 2024*

indice

*Ejercicio 1:*

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

Un dibujo de un perro

Descripción generada automáticamente con confianza media

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

Resolución necesaria en Word:

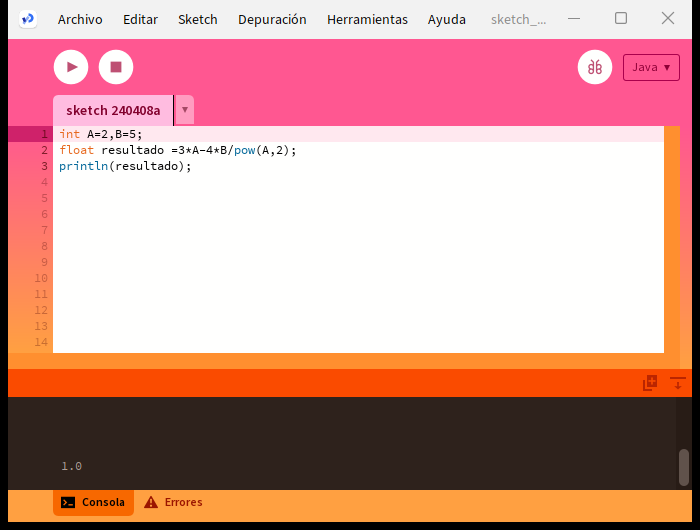
(3\*A) -(4\*B/(A^2))

6-(4\*B/4)

6-5

1

*Captura de Processing*



*Ejercicio 2:*

Evaluar la siguiente expresión



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

1 + 3/1/25

1 + 3/25

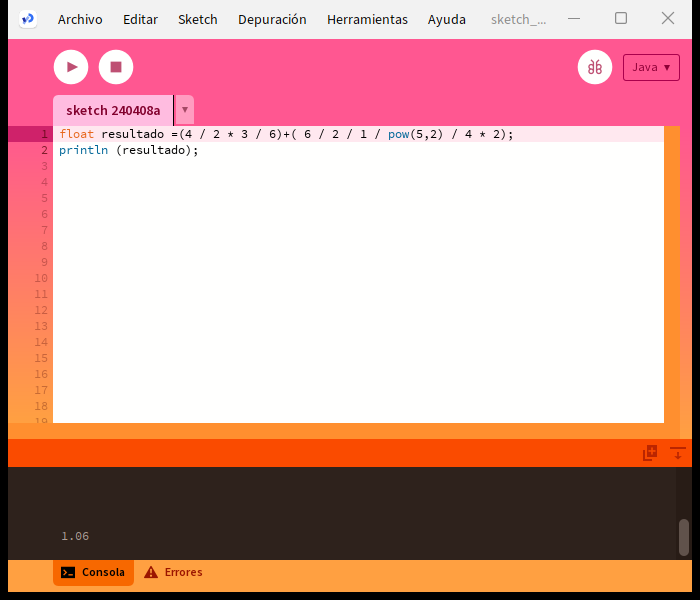
1 + 0,12/4

1 + 0,3\*2

1 + 0,06

1,06

*Captura de Processing*



*Ejercicio 3:*

Escribir las siguientes expresiones algebraicas como expresiones algorítmicas

(en su forma aritmética dentro del algoritmo). En este caso no se pide evaluarlas ni

programarlas.

*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.

Un dibujo de un perro

Descripción generada automáticamente con confianza media

a=16 b=4 c=2

1. b ^ 2 – 4 \* a \* c

|  |  |
| --- | --- |
| Aritmética | Algebraica |
| b ^ 2 – 4 \* a \* c  (16^ 2) - (4\* 4 \* 2)  256– 32  248 | 256 |

Un dibujo de un perro

Descripción generada automáticamente con confianza media

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

x=8

|  |  |
| --- | --- |
| Aritmética | Algebraica |
| 3 \* X ^ 4 – 5 \* X ^ 3 + X 12 – 17  (3\*(8^4)) -(5\*(8^3))+(8\*12)-17  12.288 - 2560 + 96 -17  9.807 | 12.288 - 2560 + 96 -17  9.807 |

Un dibujo de un perro

Descripción generada automáticamente con confianza media

b=6 c=2 d=30

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

|  |  |
| --- | --- |
| Aritmética | Algebraica |
| (b+d)/(c+4)  (6+30)/(2+4)  36/6  6 | =6 |

1. Un dibujo de un perro

   Descripción generada automáticamente con confianza media(x ^ 2 + y ^ 2) ^ (1 / 2)

X=1 y=1

|  |  |
| --- | --- |
| Aritmética | Algebraica |
| (x^2+y^2) ^(1/2)  (4^2 + 3^2) ^(1/2)  25^(1/2)  5 | (      =5 |

*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:

1. (B \* A) – (B ^ 2 / 4 \* C)

20 - 25/4 \*C

20 - 6,25 \*C

20 - 6,25

13,75

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

20 / 9

2,22



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

((6/2\*A+10) \*3\*B) -6

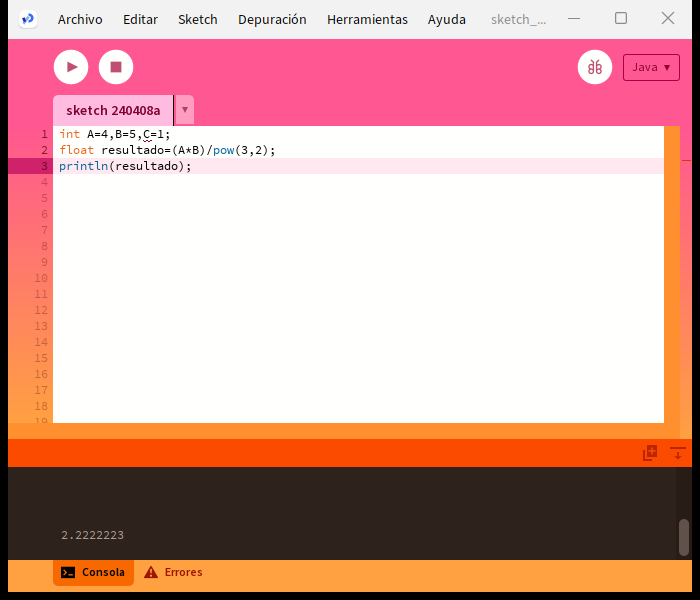
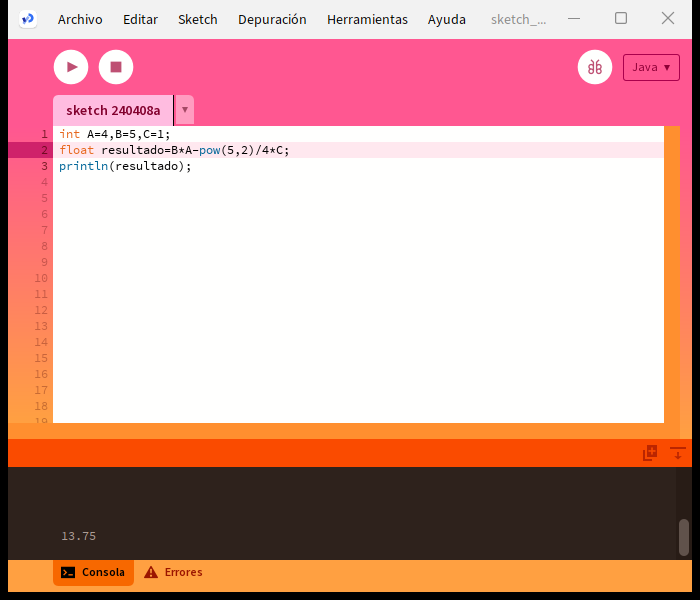
((3\*A+10) \*3\*B)-6

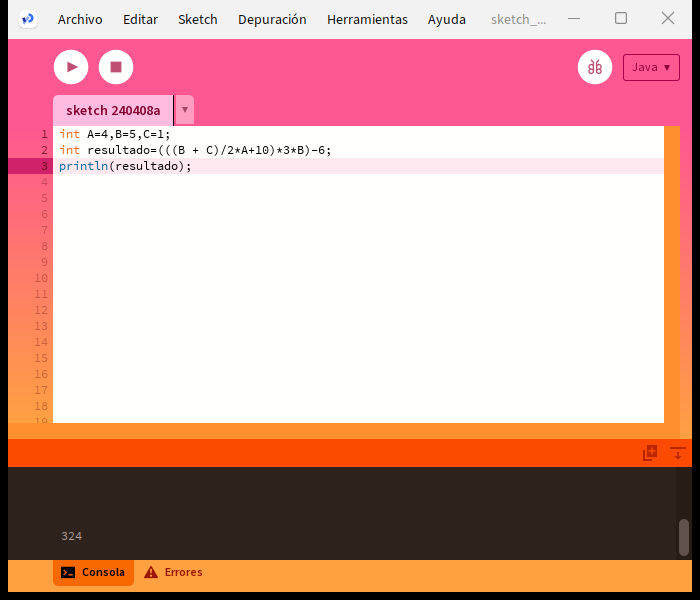
((12+10) \*3\*B) -6

(22\*3\*B)-6

(66\*B)-6

330-6=324

***Captura de Processing*

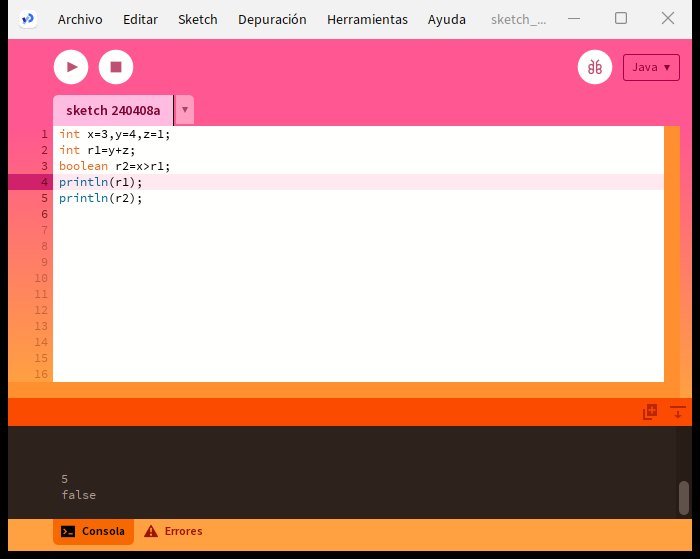


*Ejercicio 6:*

Para x=3, y=4; z=1, evaluar el resultado de R1 = y+z R2 = x >= R

R1=4+1=5

R2=3>5 =Falso

*Captura de Processing*

*Ejercicio 7*

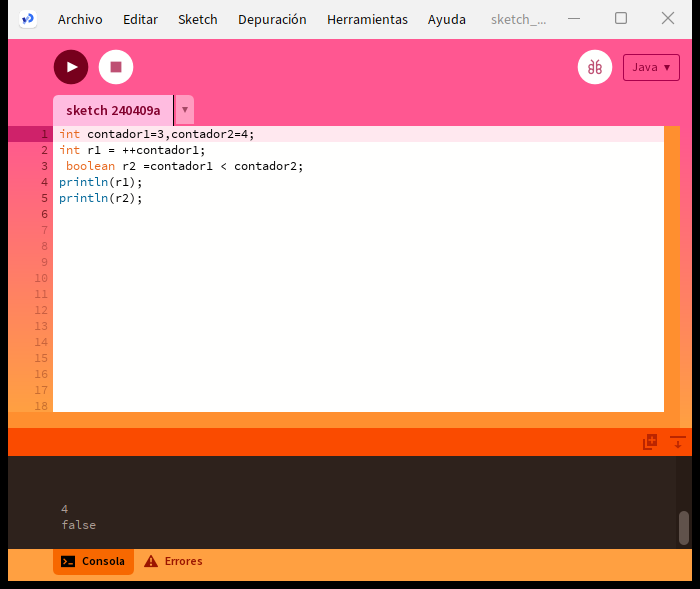
 Para contador1=3, contador2=4, evaluar el resultado de

R1 = ++contador1

R2 = contador1 < contador2

R1 = 1+3 = 4

R2 = 4 < 4 = False

*Captura de Processing*

Ejercicio 8

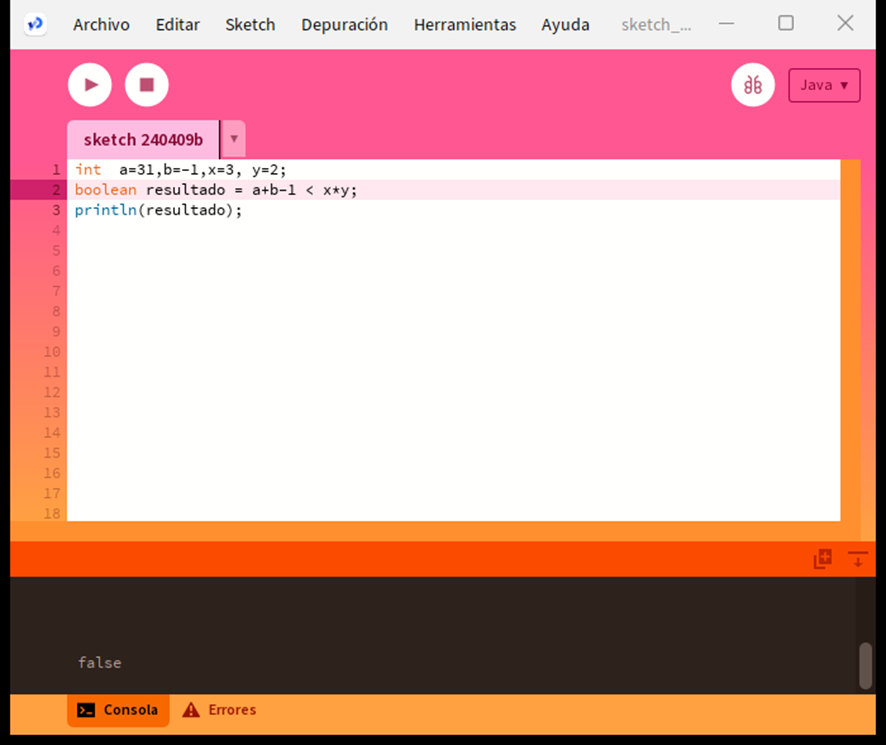
Para a=31, b=-1; x=3, y=2, evaluar el resultado de



a+b-1 < x\*y

30-1<6

29<6 = false

*Captura de Processing*

*Ejercicio 09*

Para , evaluar el resultado de



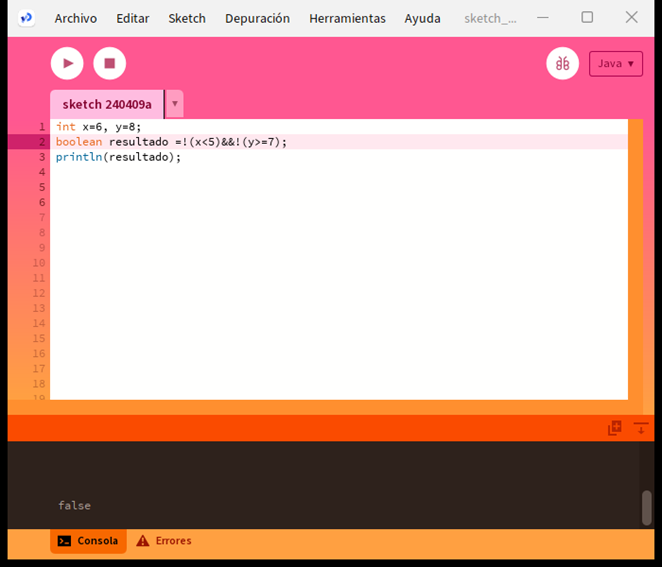
!(x<5)&&!(y>=7)

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

!(Falso)&&!(verdadero)

Verdadero && Falso=Falso

*Captura de Processing*



*Ejercicio 10*

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

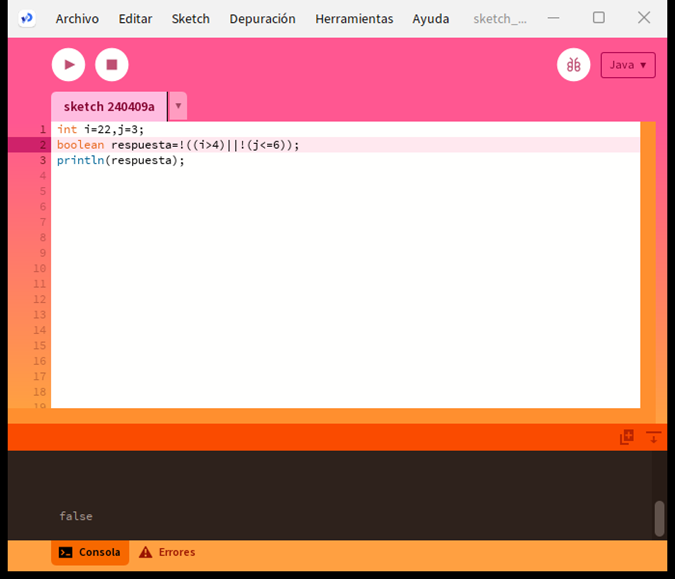
! ((i>4) ||! (j<=6))

! ((22>4)||!(3<=6))

! (Verdadero)||!(Verdadero)

Falso

*Captura de Processing*



*Ejercicio 11*

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

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

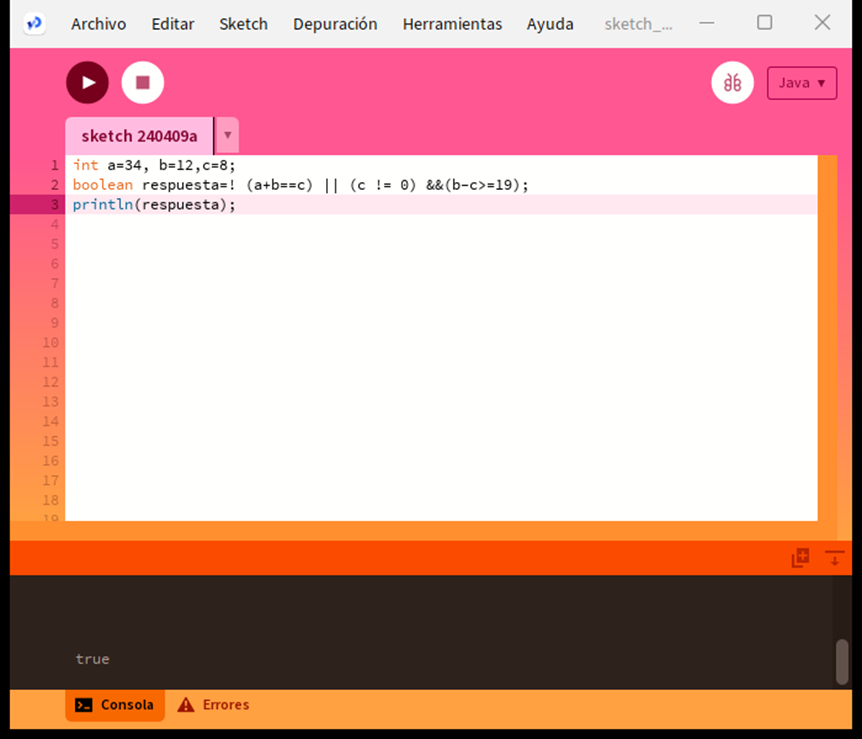
! (34+12==8)||(8!=0)&&(12-8>=19)

!(46==8)||Verdadero&&(4>=19)

!(Falso)||Verdadero&& Falso

Verdadero||Falso=Verdadero

*Captura 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.

**PLANTILLA EJERCICIOS**

**Definición del Problema: Ingresar un nombre y un saludo**

**Análisis:**

* **Datos de Entrada:** Nombre de la persona
* **Datos de Salida:** “Bienvenido.” Nombre de la persona
* **Proceso:**

¿Quién debe realizar el proceso?: El usuario

¿Cuál es el proceso que realiza …? Unir el mensaje con el

nombre de la persona

**Diseño:**

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA: **El usuario** |
| **VARIABLES**  NombreUsuario:String // almacenan el nombre del usuario que se agrega  mensajeBienvenida:String// almacena el mensaje de mensaje final |
| NOMBRE ALGORITMO:saludar\_usuario  PROCESO DEL ALGORITMO   * + - 1. Inicio       2. *Mostrar* “Ingrese su nombre: ”       3. *Leer* NombreUsuario       4. saludoBienvenida← “Bienvenido” + Nombreusuario       5. Mostrar mensajeBienvenida}       6. fin |

*Interfaz de usuario gráfica, Aplicación, PowerPoint

Descripción generada automáticamenteCaptura de Processing*

*Ejercicio 13:*

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

*PLANTILLA EJERCICIOS*

**Definición del Problema: Calcular el perímetro y área de un rectángulo**

**Análisis:**

* **Datos de Entrada:**

baseRectangulo:Entero

alturaRectangulo:Entero

* **Datos de Salida:**

perimetroRectangulo:Entero

areaRectangulo:Entero

* **Proceso:**

¿Quién debe realizar el proceso?: El alumno

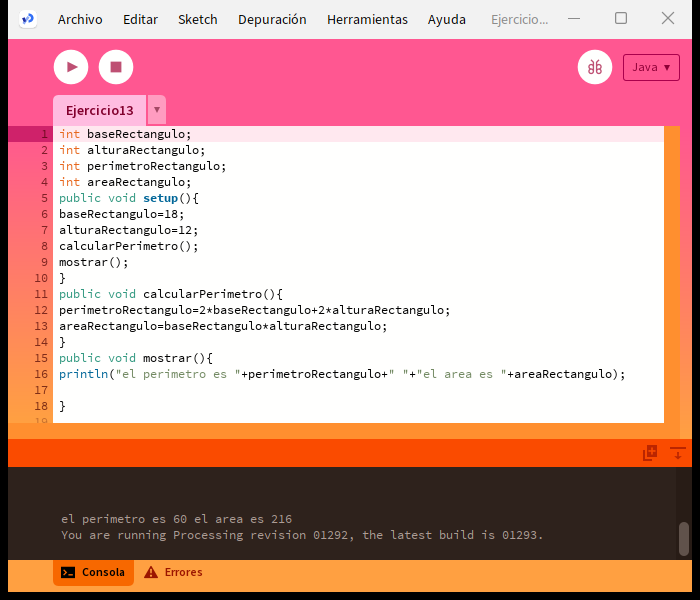
¿Cuál es el proceso que realiza …?

perimetroRectangulo🡨 2\*baseRectangulo+ 2\*alturaRectangulo

areaRectangulo🡨baseRectangulo\*alturaRectangulo

**Diseño:**

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA: **El alumno** |
| **VARIABLES**  baseRectangulo:Entero// almacenan la base del rectángulo  alturaRectangulo:Entero// almacena el área del rectángulo  perimetroRectangulo:Entero// almacena el resultado de 2\*BaseRect+ 2\*AlturaRect  areaRectangulo:Entero// almacena el resultado BaseRect\* AlturaRect |
| NOMBRE ALGORITMO: Calcular­­­­­\_perimetro­­ \_area  PROCESO DEL ALGORITMO   1. inicio 2. *Leer* baseRectangulo 3. *Leer* alturaRectangulo 4. PerimetroRect 🡨 2\*BaseRect+ 2\*AlturaRect 5. Árearect🡨 BaseRect\* AlturaRect 6. *Mostrar* PerimetroRect 7. *Mostrar* Árearect 8. *fin* |

*Captura de Processing*

*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.

Imagen que contiene Diagrama

Descripción generada automáticamente

PLANTILLA EJERCICIOS

**Definición del Problema**: **Obtener la hipotenusa**

**Análisis:**

**•Datos de Entrada:**

catetoBase:Entero

catetoAltura:Entero

**•Datos de Salida:**

hipotenusaTriangulo:Real

**•Proceso:**

¿Quién debe realizar el proceso?: matemático

¿Cuál es el proceso que realiza …?

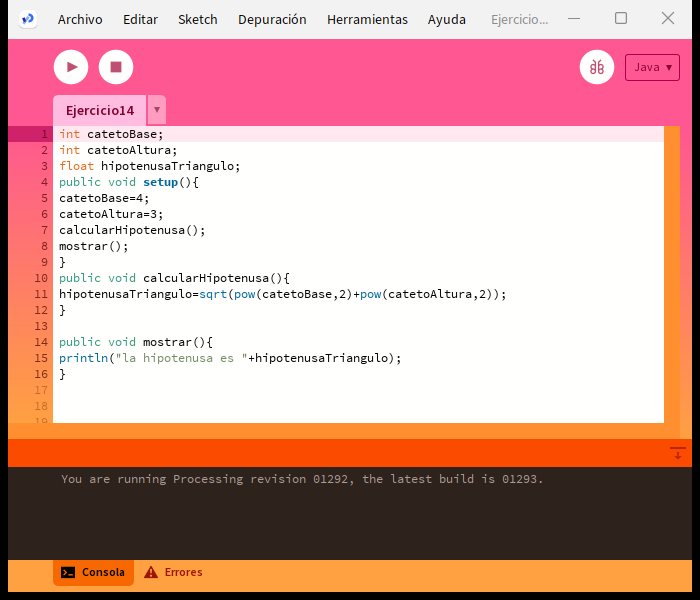
Teorema de Pitágoras🡨

Una caricatura de una persona

Descripción generada automáticamente con confianza baja**Diseño:**

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA: **Matemático** |
| **VARIABLES**  catetoBase:Entero// almacena la base del triangulo  catetoAltura:Entero// almacena la altura del triangulo  hipotenusaTriangulo:Real // almacena la hipotenusa del triangulo |
| NOMBRE ALGORITMO: calcular\_hipotenusa  PROCESO DEL ALGORITMO   1. inicio 2. catetoBase🡨4 3. catetoAltura🡨3 4. hipotenusaTriangulo 🡨 5. *Mostrar* hipotenusaTriangulo 6. *fin* |

*Captura de Processing*

**

*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.

PLANTILLA EJERCICIOS

**Definición del Problema: dos números calcular la suma, resta, multiplicación y división**

**Análisis:**

**•Datos de Entrada:**

primerNumero:Entero

segundoNumero:Entero

**•Datos de Salida:**

resultadoSuma:Entero

resultadoRestar:Entero

resultadoMultiplicar:Entero

resultadoDividir:Entero

**•Proceso:**

¿Quién debe realizar el proceso?: Profesor

¿Cuál es el proceso que realiza …?

resultadoSuma 🡨primerNumero+ segundoNumero

resultadoRestar 🡨primerNumero- segundoNumero

resultadoMultiplicar 🡨primerNumero\* segundoNumero

resultadoDividir🡨primerNumero/ segundoNumero

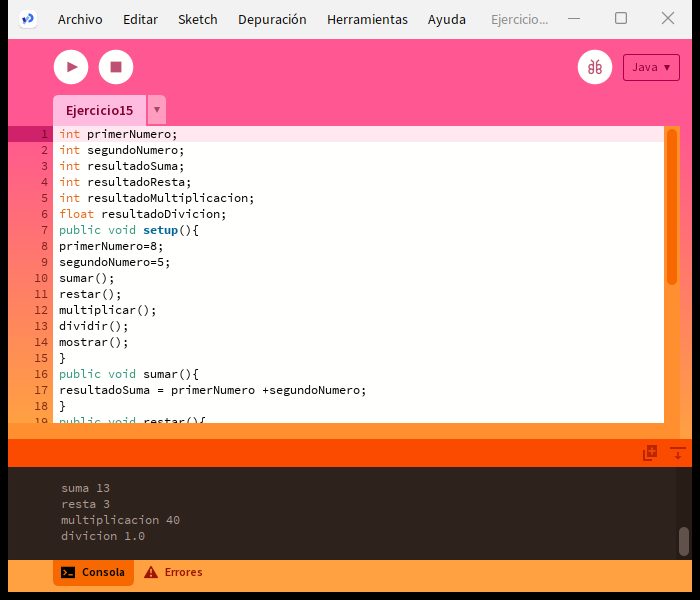
**Una caricatura de una persona

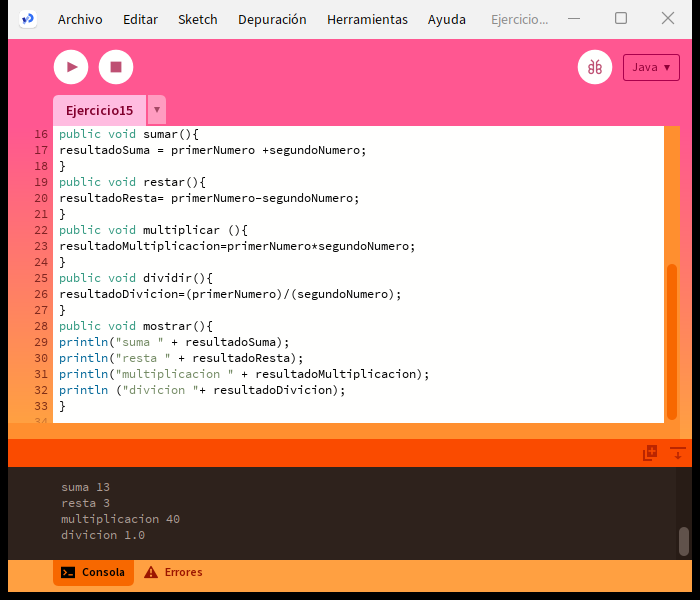
Descripción generada automáticamente con confianza baja**

**Diseño:**

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA : **Profesor** |
| **VARIABLES**  primerNumero, segundoNumero: Entero // almacenan el valor del primer y segundo numero  resultadoSuma: Entero // almacena el resultado de la suma  resultadoRestar:Enterol//almacena el resultado de la resta  resultadoMultiplicar:Entero//almacena el resultado de multiplicar  resultadoDividir:Entero//almacena el resultado de dividir |
| NOMBRE ALGORITMO: sumar  PROCESO DEL ALGORITMO   1. inicio 2. primerNumero🡨8 3. segundoNumero🡨5 4. resultadoSuma 🡨 primerNumero + segundoNumero 5. *Mostrar* resultadoSuma 6. *Fin* |
| PROCESO DEL ALGORITMO restar   1. Inicio 2. primerNumero🡨8 3. segundoNumero🡨5 4. resultadoRestar 🡨 primerNumero - segundoNumero 5. *Mostrar* resultadoRestar 6. Fin |
| NOMBRE ALGORITMO: multiplicar  PROCESO DEL ALGORITMO   1. inicio 2. primerNumero🡨8 3. segundoNumero🡨5 4. resultadoMultiplicar 🡨 primerNumero \* segundoNumero 5. *Mostra*r resultadoMultiplicar 6. Fin |
| PROCESO DEL ALGORITMO dividir   1. inicio 2. primerNumero🡨8 3. segundoNumero🡨5 4. resultadoDividir🡨 primerNumero / segundoNumero 5. *Mostrar* resultadoDividir 6. *Fin* |

*Captura de Processing*



**

*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



*PLANTILLA EJERCICIOS*

**Definición del Problema: convertir una** **temperatura Fahrenheit en grados Celsius.**

**Análisis:**

**•Datos de Entrada:**

temperaturaFahrenheit :Real

**•Datos de Salida:**

resultadoCelsius: Real

**•Proceso:**

Una caricatura de una persona

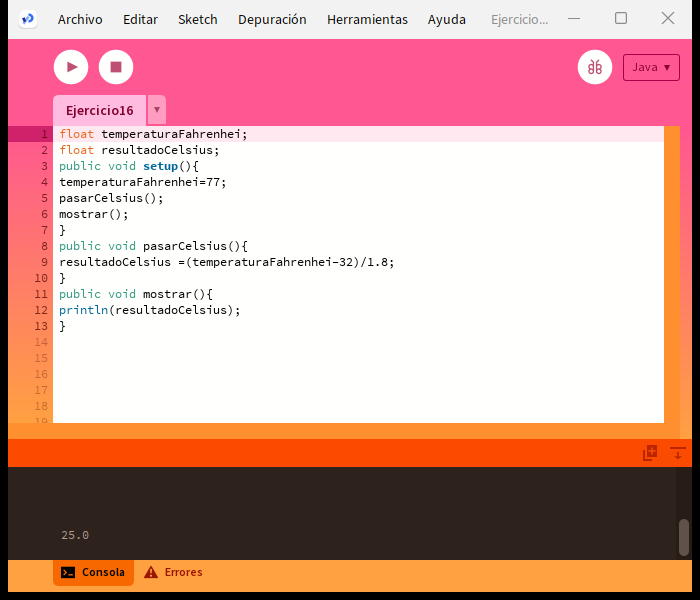
Descripción generada automáticamente con confianza baja¿Quién debe realizar el proceso?: La calculadora

¿Cuál es el proceso que realiza …?

resultadoCelsius 🡨 (temperaturaFahrenhei-32)/1.8

**Diseño:**

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA: **La calculadora** |
| **VARIABLES**  temperaturaFahrenheit :Real// almacenan la temperatura fahrenheit  resultadoCelsius: Real // almacena los grados Celsius. |
| NOMBRE ALGORITMO:Pasar\_a\_Celsius  PROCESO DEL ALGORITMO   1. Inicio 2. temperaturaFahrenheit 3. resultadoCelsius 🡨(temperaturaFahrenhei-32)/1.8; 4. *Mostrar* resultadoCelsius 5. Fin |

*Captura de Processing*

*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.

*PLANTILLA EJERCICIOS*

**Definición del Problema: obtener las raíces de una ecuación de segundo grado.**

**Análisis:**

**•Datos de Entrada:**

a,b,c,d:Entero

discriminante:Real

**•Datos de Salida:**

float x1,x2:Real

**•Proceso:**

Una caricatura de una persona

Descripción generada automáticamente con confianza baja¿Quién debe realizar el proceso?: Matematico

¿Cuál es el proceso que realiza …?

**Diseño:**

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA: **Matematico** |
| **VARIABLES**  a,b,c,d:Entero  discriminante:Real  float x1,x2:Real |
| NOMBRE ALGORITMO: resolver  PROCESO DEL ALGORITMO   1. Inicio 2. A🡨1 3. b🡨-5 4. C🡨6 5. discriminante🡨(b^2)-((4\*a)\*c) 6. **SI** (discriminante<0) 7. Mostrar("error"); 8. **Si\_no Si** (discriminante>0) 9. x1🡨(-b-discriminante)/(2\*a) 10. **Mostrar** “primer resultado” + x1 11. x2🡨(-b+discriminante)/(2\*a) 12. *Mostrar* “segundo resultado” +x2 13. **Si\_no Si**(discriminante==0) 14. x1=(-b)/(2\*a) 15. *Mostrar* x1 16. **fin\_si** 17. fin |

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media*Captura de Processing*

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.

*PLANTILLA EJERCICIOS*

**Definición del Problema:** **dibujar una línea que se dibuja por todo el ancho.Sobre el punto medio de la línea dibuje una elipse**

**Análisis:**

**•Datos de Entrada:**

posY:Entero

incremento:Entero

distanciaRadio:Entero

**•Datos de Salida:**

**dibujar**

**•Proceso:**

¿Quién debe realizar el proceso?: Programador

¿Cuál es el proceso que realiza …?

Una caricatura de una persona

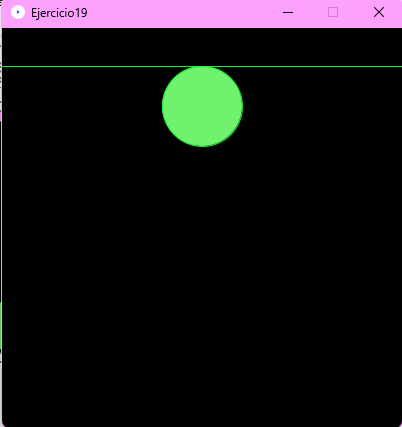
Descripción generada automáticamente con confianza bajaUsar la Estructura Selectiva Simple SI (if) para dibujar una línea que se dibuja por todo el ancho.Sobre el punto medio de la línea dibuje una elipse.

**Diseño:**

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA: **Programador** |
| **VARIABLES**  posY:Entero  incremento:Entero  distanciaRadio:Entero |
| NOMBRE ALGORITMO:dibujar  PROCESO DEL ALGORITMO   1. inicio 2. posY🡨0 3. incremento🡨1 4. distanciaRadio🡨40 5. posY+=incremento 6. **SI**(posY>=height || posY<=0) **entonces** 7. incremento\*=-1 8. distanciaRadio\*=-1 9. *Mostrar* 10. **fin\_si** 11. fin |

*Captura de Processing*

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamenteInterfaz de usuario gráfica

Descripción generada automáticamente

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza mediaInterfaz de usuario gráfica

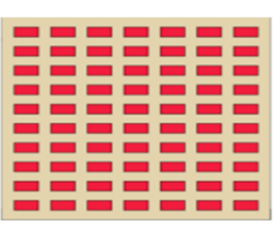
Descripción generada automáticamente con confianza mediaInterfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

EJERCICIO 20

Forma, Cuadrado

Descripción generada automáticamenteDibuje 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:



**PLANTILLA EJERCICIOS**

**Definición del Problema: Dibujar en toda la extensión del lienzo rectángulos de idénticas medidas**

**Análisis:**

* **Datos de Entrada:**

Entero: anchoRectangulo

Entero: altoRectangulo

Entero: distanciaEntreRectangulo;

PVector posición

* **Datos de Salida:**

dibujarRectangulo

* **Proceso:**

¿Quién debe realizar el proceso?: Programador

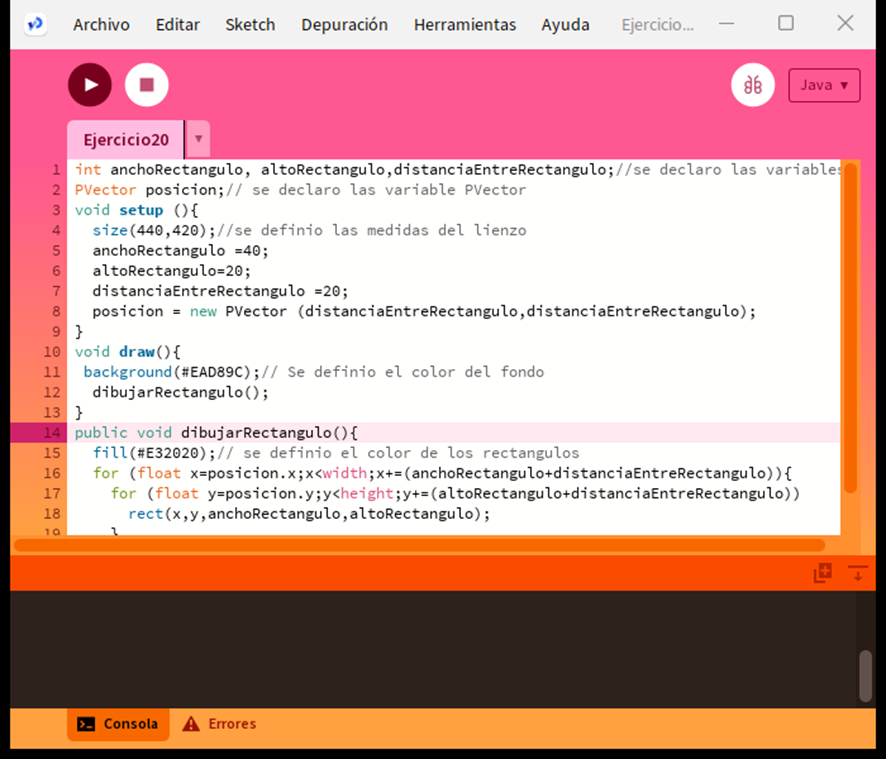
¿Cuál es el proceso que realiza …?

Dibujar rectángulos de idénticas medidas usando la ESTRUCTURA ITERATIVA PARA (For)

**Diseño:**

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA :**Programador** |
| **VARIABLES**  anchoRectangulo:Entero//almacena el ancho del rectangulo  altoRectangulo:Entero// almacena el alto del rectangulo  distanciaEntreRectangulo: Entero//almacena la distancia entre los rectangolos  posición : PVector // almacena la posición x e y del rectángulo |
| NOMBRE ALGORITMO: dibujar\_Rectangulo  PROCESO DEL ALGORITMO   1. Inicio 2. anchoRectangulo 🡨 40 3. altoRectangulo🡨 20 4. distanciaEntreRectangulo 🡨 20 5. Posición 🡨distanciaEntreRectangulo, distanciaEntreRectangulo 6. Para x 🡨 posicion.x hasta x<width incremento x+🡨(anchoRectangulo+distanciaEntreRectangulo)) 7. Para y 🡨posicion.y hasta y<heigth incremento y+🡨(altoRectangulo+distanciaEntreRectangulo)) 8. rect(x,y,anchoRectangulo,altoRectangulo) 9. fin\_para 10. fin para 11. fin |

*Captura de Processing*

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

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.

PLANTILLA EJERCICIOS

**Definición del Problema**: **dibujar líneas que forman escalones y sobre cada borde de escalon se dibuje un punto de color rojo**

**Análisis:**

**•Datos de Entrada:**

x,y=Entero

anchoEscalon=entero

altoEscalon=entero

**•Datos de Salida:**

dibujarEscalon

**•Proceso:**

¿Quién debe realizar el proceso?: programador

¿Cuál es el proceso que realiza …?

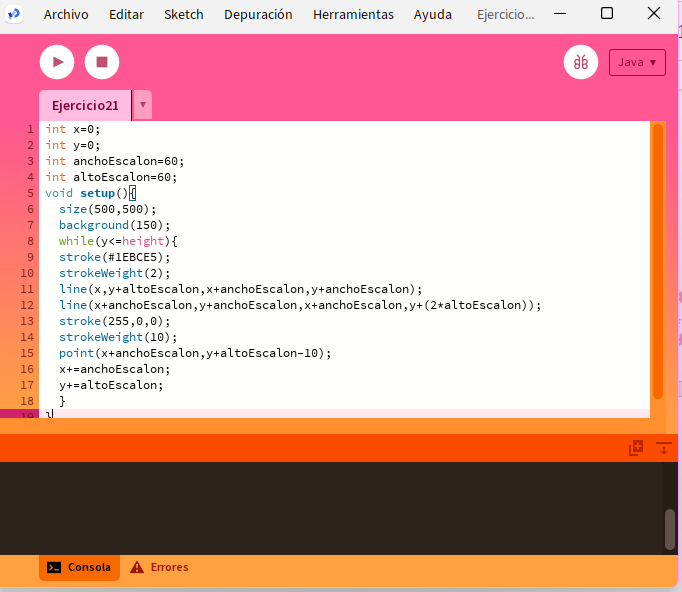
Usando la ESTRUCTURA ITERATIVA mientras(while) se dibujará unas líneas que forman una escalera con unos puntos sobre ella en las esquinas

Una caricatura de una persona

Descripción generada automáticamente con confianza baja

**Diseño:**

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA: **Programador** |
| **VARIABLES**  x:Entero  y:Entero  anchoEscalon=Entero  altoEscalon=Entero |
| NOMBRE ALGORITMO: dibujar\_Escalon  PROCESO DEL ALGORITMO   1. Inicio 2. X🡨0 3. Y🡨0 4. anchoEscalon🡨60 5. altoEscalon🡨60 6. **mientras** (y<height) hacer 7. line(x,y+altoEscalon,x+anchoEscalon,y+anchoEscalon) 8. line(x+anchoEscalon,y+anchoEscalon,x+anchoEscalon,y+(2\*altoEscalon)) 9. point(x+anchoEscalon,y+altoEscalon-10) 10. x+=anchoEscalon 11. y+=altoEscalon 12. **Fin\_mientras** 13. *Fin* |

*Captura de Processing*

*Gráfico, Gráfico de cajas y bigotes

Descripción generada automáticamente*

*Ejercicio 22:*

Forma, Cuadrado

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

Diagrama

Descripción generada automáticamente

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.

PLANTILLA EJERCICIOS

**Definición del Problema**: **dibujar líneas de igual medida donde se dibujarán círculos de por medio**

**Análisis:**

**•Datos de Entrada:**

y:Entero

x :Entero

distanciaLineas:Entero

distanciaCirculos:Entero

yc :Entero

xc:Entero

**•Datos de Salida:**

dibujar

**•Proceso:**

¿Quién debe realizar el proceso?: Programador

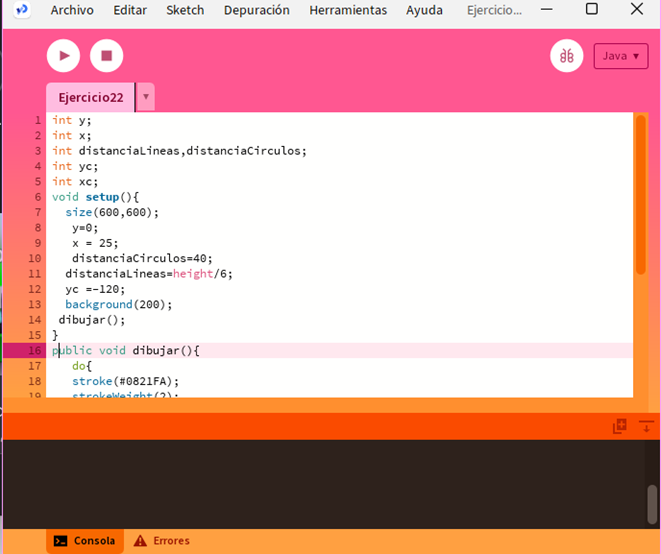
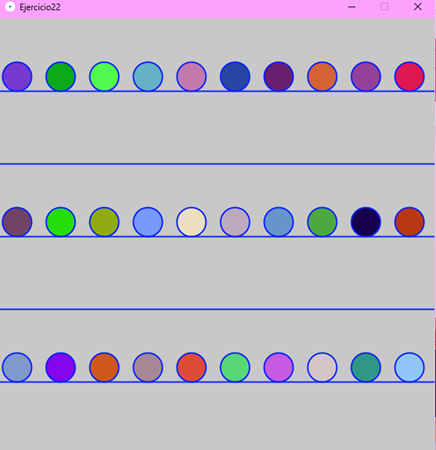
Una caricatura de una persona

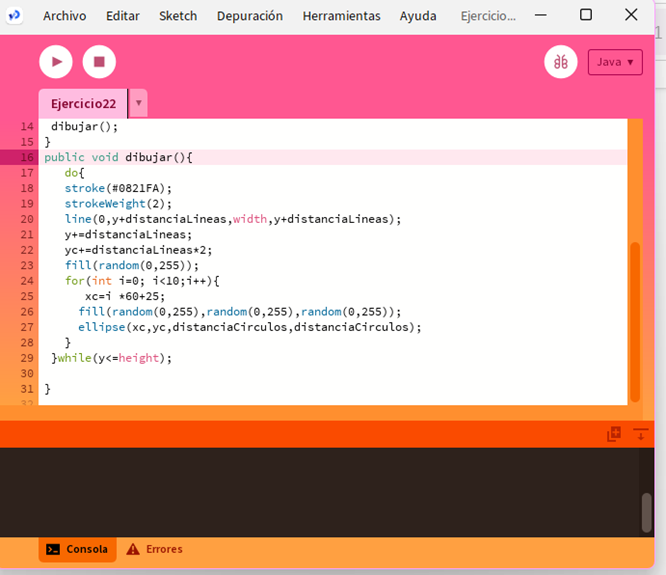
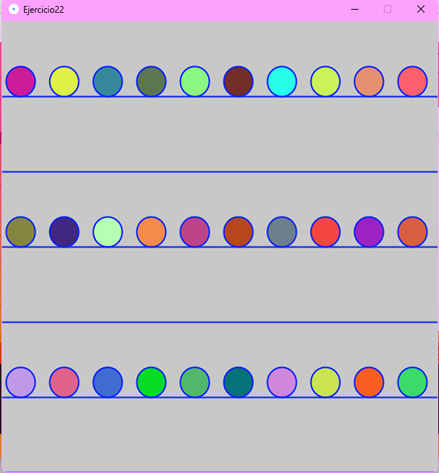
Descripción generada automáticamente con confianza baja¿Cuál es el proceso que realiza …?

Usar la ESTRUCTURA ITERATIVA HACER-MIENTRAS(do-While)

**Diseño:**

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA: **Programador** |
| **VARIABLES**  y:Entero  x :Entero  distanciaLineas:Entero  distanciaCirculos:Entero  yc :Entero  xc:Entero |
| NOMBRE ALGORITMO: dibujar  PROCESO DEL ALGORITMO   1. Inicio 2. x🡨0 3. y🡨25 4. distanciaLineas 5. distanciaCirculos🡨40 6. **Hacer** 7. line(0,y+distanciaLineas,width,y+distanciaLineas) 8. y+=distanciaLineas 9. yc+=distanciaLineas\*2 10. fill(random(0,255)) 11. **Para** i🡨0;**hasta** i<10 **Incremento** i++ **hacer** 12. xc=i \*60+25 13. fill(random(0,255),random(0,255),random(0,255)) 14. ellipse(xc,yc,distanciaCirculos,distanciaCirculos) 15. **Fin\_Para** 16. **Mientras**(y<=height) 17. *Fin* |

*Captura de Processing*



**conclusión**