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

*Profesores:*

*Mg. Ing. Ariel Alejandro Vega*

Trabajo Práctico/Actividad

N°1

Apellido y Nombre – LU /

Flores José Fernando

TUV000607

**FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS**

indice

[Ejercicio1 4](#_Toc164212421)

[Ejercicio 2 5](#_Toc164212422)

[Ejercicio 4 6](#_Toc164212423)

[Capturas Processing ejercicio4 7](#_Toc164212424)

[Ejercicio 5 7](#_Toc164212425)

[Ejercicio 6 9](#_Toc164212426)

[Ejercicio 8 10](#_Toc164212427)

[Ejercicio 10 11](#_Toc164212428)

[Ejercicio 12 12](#_Toc164212429)

[Ejercicio 13 13](#_Toc164212430)

[Captura Processing 14](#_Toc164212431)

[Ejercicio 14 15](#_Toc164212432)

[Ejercicio 15 17](#_Toc164212433)

[Captura Processing 18](#_Toc164212434)

[Ejercicio 16 19](#_Toc164212435)

[Ejercicio 17 20](#_Toc164212436)

[Ejercicio 18 22](#_Toc164212437)

[24](#_Toc164212438)

[Ejercicio 19 25](#_Toc164212439)

[Ejercicio 20 27](#_Toc164212440)

[**Captura de pantalla, prosessing** 28](#_Toc164212441)

[Ejercicio 21 29](#_Toc164212442)

[Ejercicio 22 31](#_Toc164212443)

[Conclusión 33](#_Toc164212444)

# Ejercicio1

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:

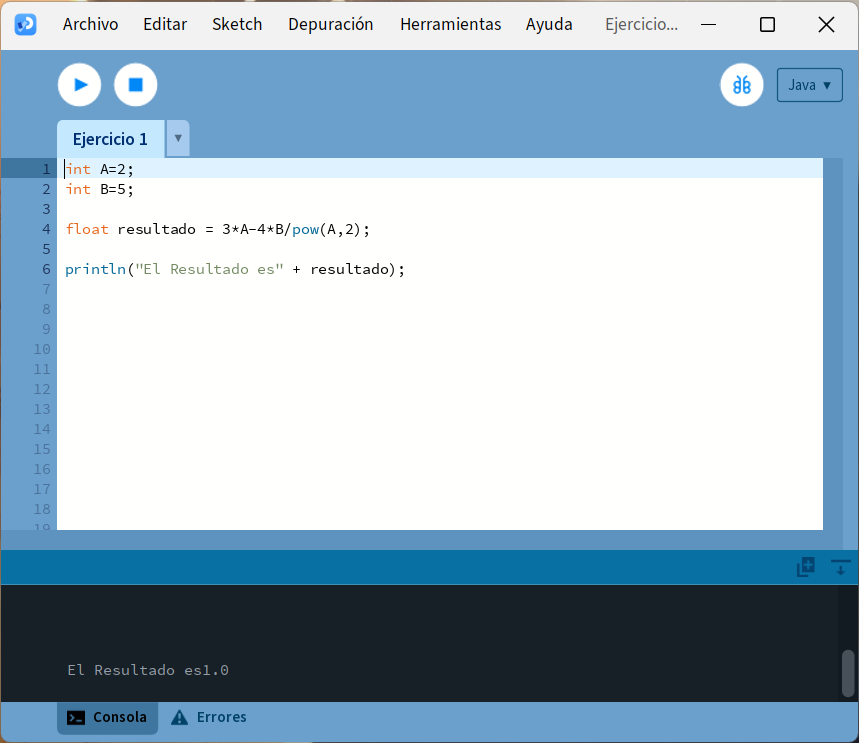
(3\*A)-(4\*B/(A^2))

6-(4\*B/4)

6-5

1

Captura de Processing



# Ejercicio 2

Evaluar la siguiente expresión 4 / 2 \* 3 / 6 + 6 / 2 / 1 / 5 ^ 2 / 4 \* 2

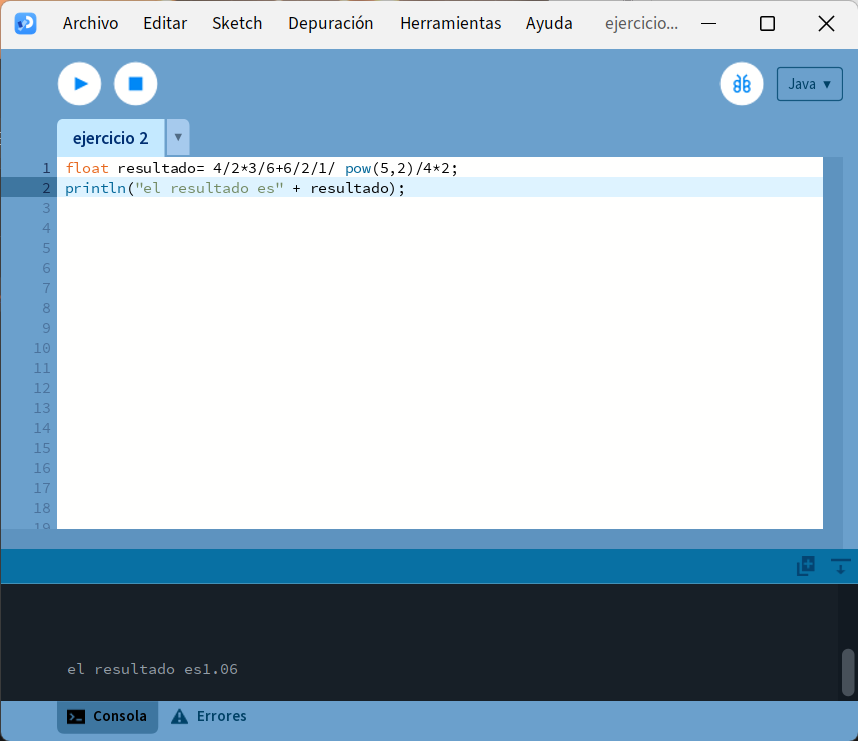
**4 / 2 \* 3 / 6 + 6 / 2 / 1 / 5 ^ 2 / 4 \* 2**

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

1.0 + 0.06

1.06

Captura de 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 a=2 b=4 c=1

|  |  |
| --- | --- |
| Aritmética | Algebraica |
| b ^ 2 – 4 \* a \* c  (4 ^ 2) - (4 \* 2 \* 1)  16 – 8  8 |  |

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

|  |  |
| --- | --- |
| Aritmética | Algebraica |
| 3 \* X ^ 4 – 5 \* X ^ 3 + X 12 – 17  (3\*(2^4))-(5\*(2^3))+(2\*12)-17  48-40+24-17  15 | 484024  15 |

c) (b + d) / (c + 4) b=2 c=3 d=2

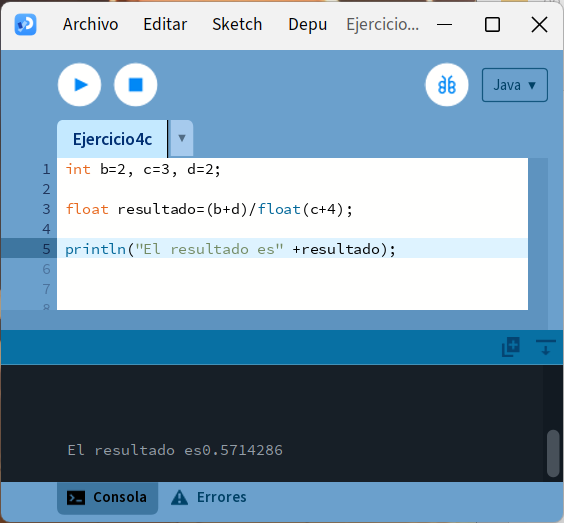
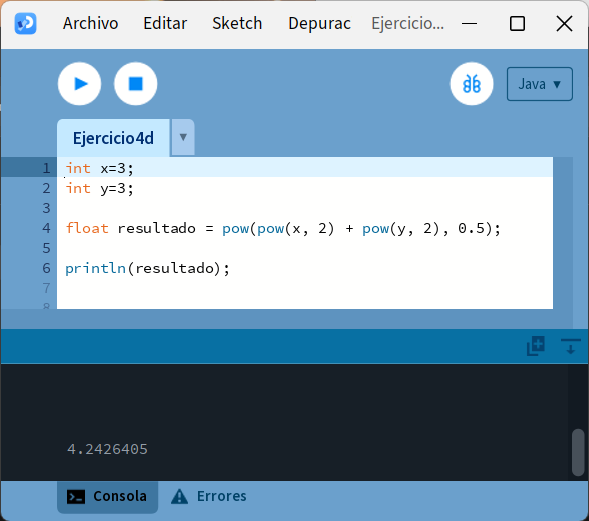
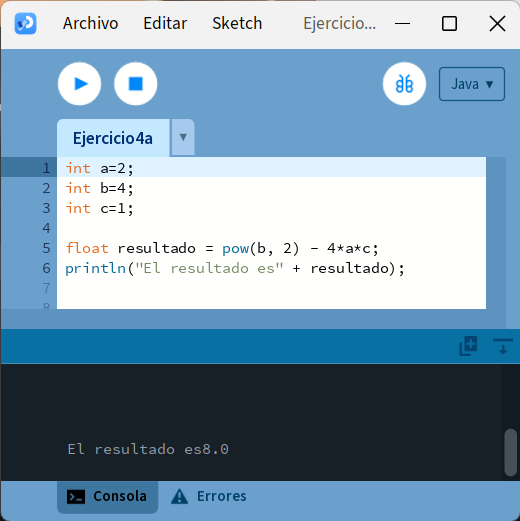
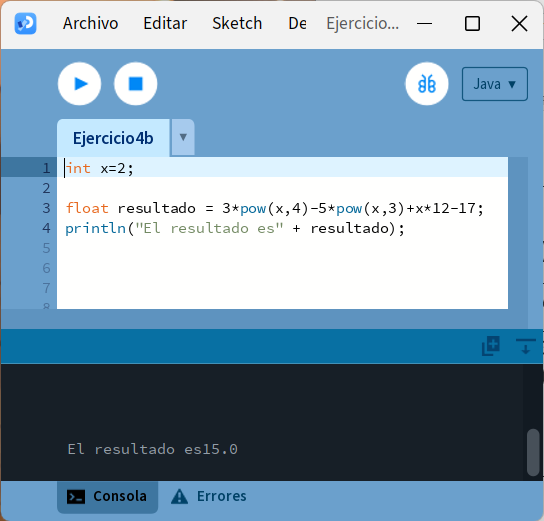
|  |  |
| --- | --- |
| Aritmética | Algebraica |
| (b+d)/(c+4)  (2+2)/(3+4)  0.5714 | =0.5714 |

d) (x ^ 2 + y ^ 2) ^ (1 / 2) x=3 y=3

|  |  |
| --- | --- |
| Aritmética | Algebraica |
| (x^2+y^2) ^(1/2)  (3^2 + 3^2) ^(1/2)  18^(1/2)  4,2426 | (    = 4,2426 |

# Capturas Processing ejercicio4

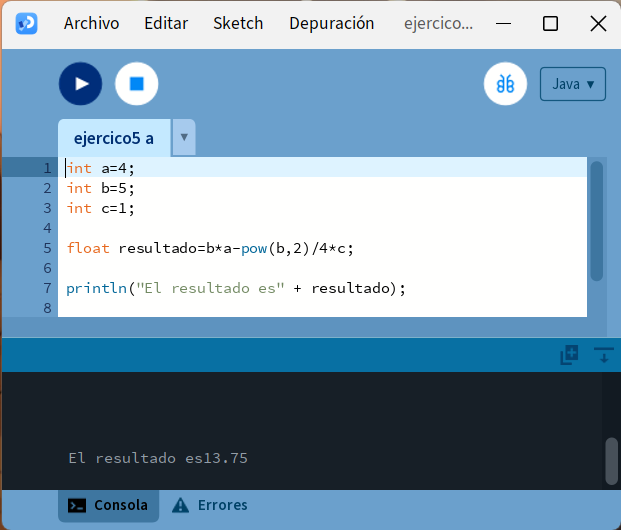
# Ejercicio 5



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 Captura Processing**

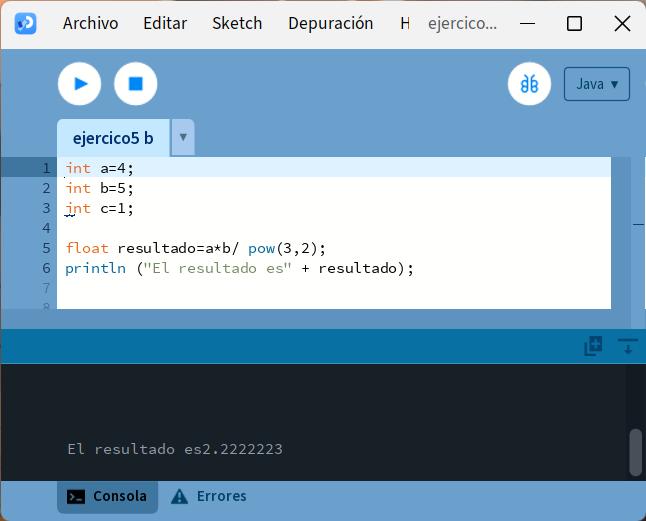
5 \* 4 – ((5 ^ 2) / 4) \* 1



20 - 6.25

13.75

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

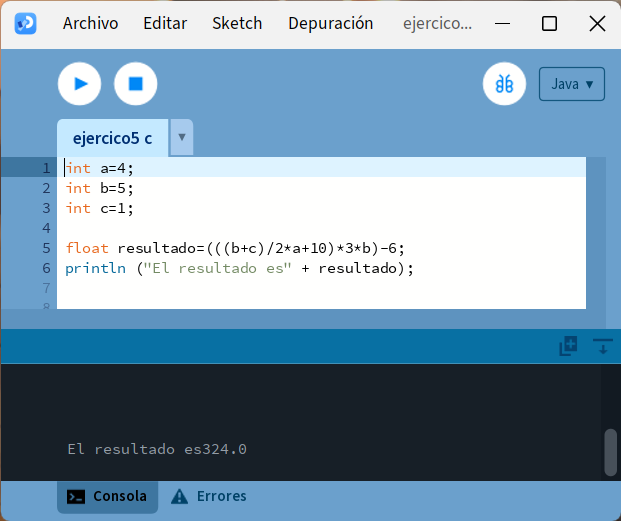


(4 \* 5) / 3 ^ 2

2.22

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

(((5 + 1) / 2 \* 4 + 10) \* 3 \* 5) - 6



((6 / 2 \* 4 + 10) \* 3 \* 5) - 6

((3 \* 4 + 10) \* 3 \* 5) - 6

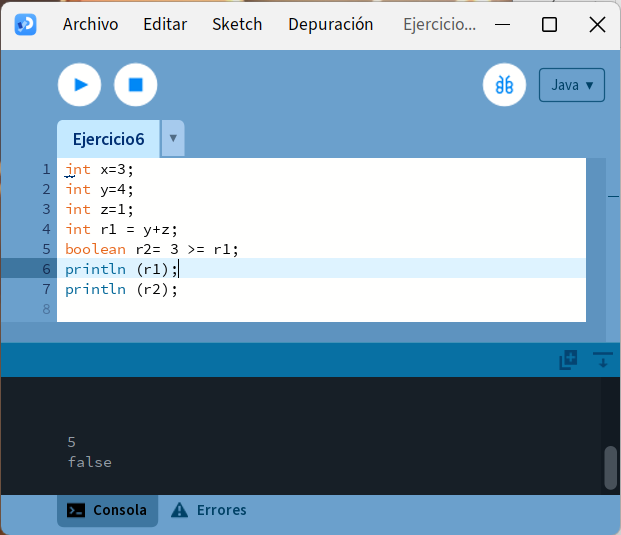
(22 \* 3 \* 5) - 6

330 – 6

324

# Ejercicio 6

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

R1=4+1 Captura Processing

R1=5

R2 =3>=R1

R2=3>=5

R2=false

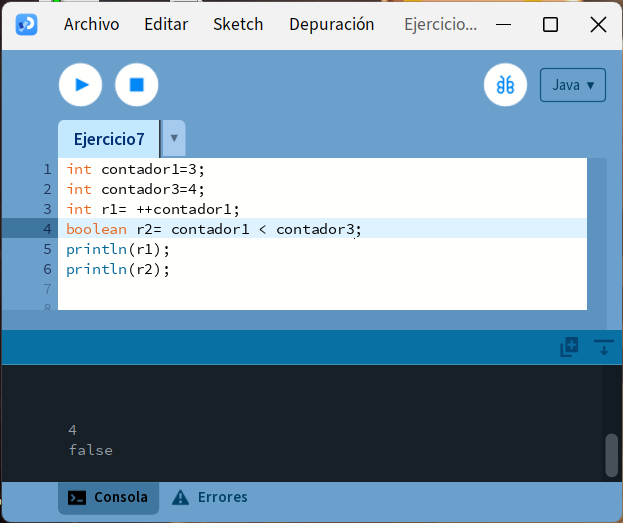
**Ejercicio 7**

Para contador 1=3, contador3=4

evaluar el resultado de

R1 = ++contador1 R2 = contador1 < contador2

R1=++contador1 Captura processing

R1=1+3

R1=4

R2=contador<contador3

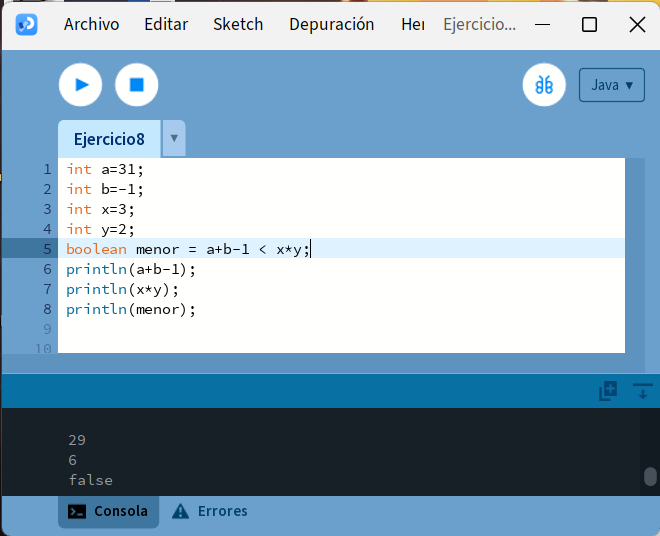
R2=4<4

R2=false

# Ejercicio 8

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

a+b-1 < x\*y Captura Processing

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

31-2 < 6

29 < 6

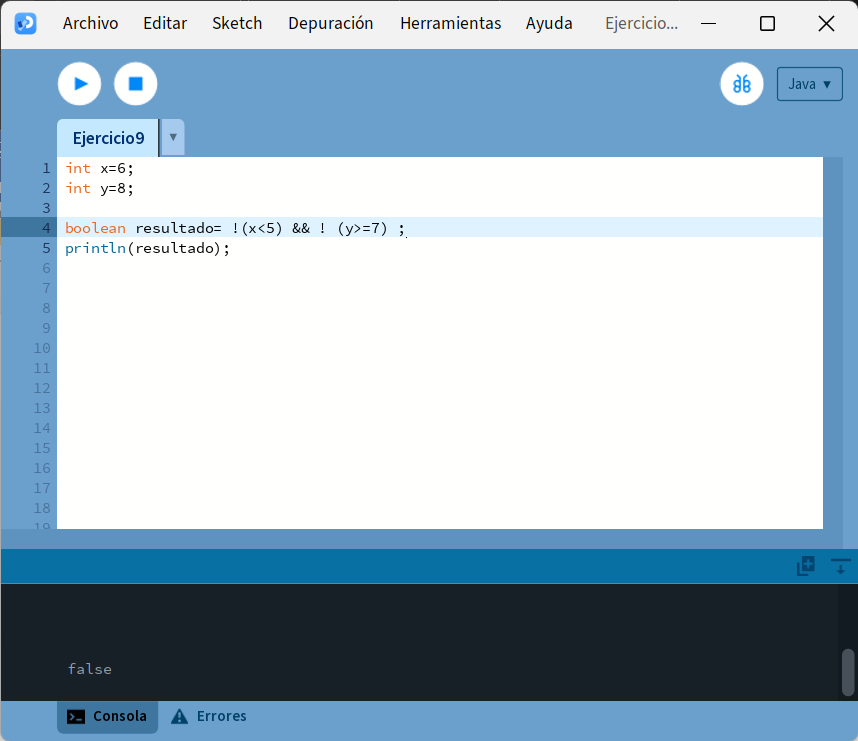
False

Ejercicio 9

Para x=6, y=8, evaluar el resultado de

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

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

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

!(verdadero) && !(verdadero)

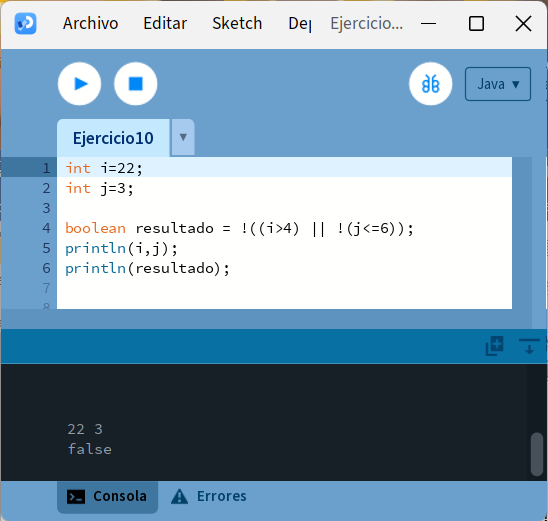
Falso && falso

falso

# Ejercicio 10

Para i=22,j=3, evaluar el resultado de !((i>4) || !(j<=6))

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

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

!((verdadero) || !(verdadero))

!((verdadero) || falso)

!(verdadero)

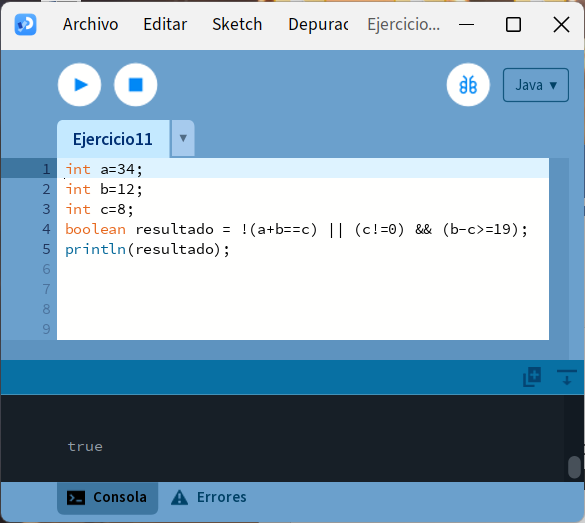
Falso

Ejercicio 11

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

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

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

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

!(falso)||(verdadero)&&(falso)

!(falso )||falso)

!(falso)

verdadero

# Ejercicio 12

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

**Definición del problema:** 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:**

SaludoBienvenido

* **Proceso:**

¿Quién debe realizar el proceso?: El estudiante

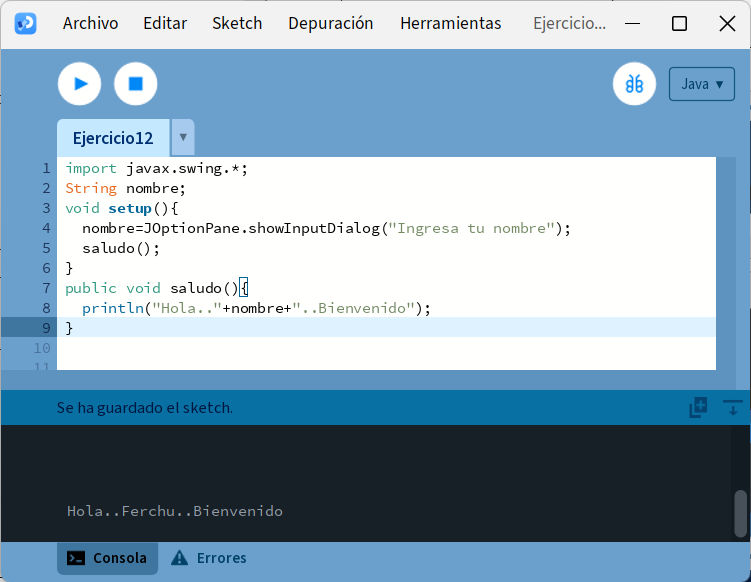
¿Cuál es el proceso que realiza?

Se debe solicitar el nombre de una persona para posteriormente

saludarlo mediante un mensaje.

Diseño:

|  |
| --- |
| Entidad que resuelve el problema: El estudiante |
| Variables  Nombre=String//almacena el nombre |
| NOMBRE ALGORITMO: bienvenido  PROCESO DEL ALGORITMO   * + - 1. **inicio**       2. Mostrar “Ingresa tu nombre”       3. Leer nombre       4. saludo ← "Hola.."+nombre+"..Bienvenido"       5. mostrar saludo       6. fin |



**Captura Processing**

# 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**

Altura: Real

Base: Real

* **Datos de Salida:**

Perímetro

Área

* **Proceso:**

¿Quién debe realizar el proceso?: El estudiante

¿Cuál es el proceso que realiza …?

Calcular el perímetro y el área con las siguientes formulas:

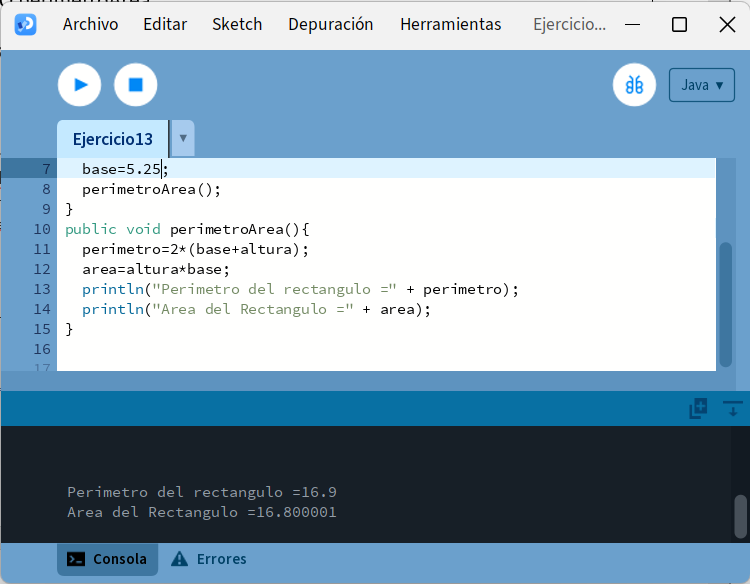
Perímetro= 2 \* (base+altura)

Área= base\*altura

Diseño:

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA: El estudiante |
| VARIABLES  Altura: Real  Base: Real  Perímetro: Real  Área: Real |
| NOMBRE ALGORITMO perimetroArea  PROCESO DEL ALGORITMO   * + - 1. ***inicio***       2. leer altura       3. leer base       4. perímetro ← 2\*(base+altura)       5. mostrar “El Perimetro de un rectángulo es” + perímetro       6. área ← altura\*base       7. mostrar “ El Area de un rectángulo es ”+área       8. **fin** |

# Captura Processing



# Ejercicio 14

Obtenga la hipotenusa de un triángulo rectángulo conociendo sus catetos

Definicion del problema: Obtenga la hipotenusa

Analisis:

* **Datos de Entrada:**

Base: Entero

Altura: Entero

* **Datos de salida:**

Hipotenusa: Real

* **Proceso:**

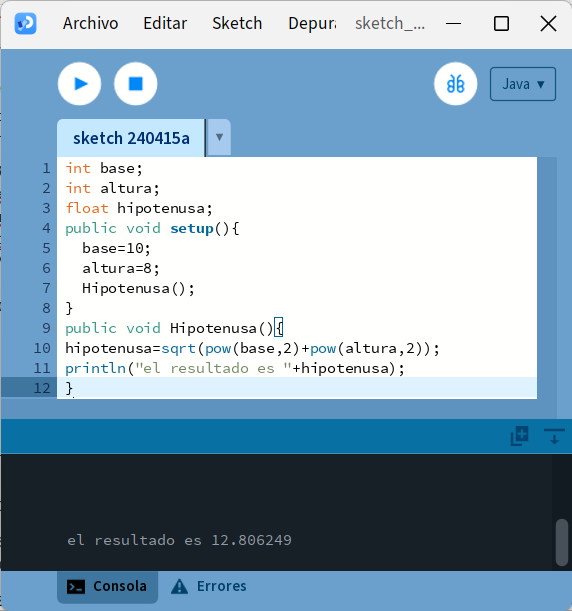
¿Quién debe realizar el proceso? Calculadora

¿Cuál es el proceso que realiza?

Obtener la hipotenusa de un triángulo rectángulo aplicando el teorema de Pitágoras y utilizando la siguiente formula

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA: CALCULADORA |
| Variables  Base: Entero  Altura: Entero  Hipotenusa: Real |
| NOMBRE ALGORITMO: hipotenusa  PROCESO DEL ALGORITMO   1. inicio 2. Base← 10 3. Altura← 8 4. hipotenusa ← 5. *Mostrar* hipotenusa 6. ***Fin*** |
|  |

Captura Processing



# Ejercicio 15

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:** Num1, Num2

* **Datos de salida:**

Suma: Entero

Resta: Entero

Multiplicación: Entero

División: Entero

* **Proceso:**

¿Quién debe realizar el proceso? Calculadora

¿Cuál es el proceso que realiza?

Dados 2 números, debe realizar las operaciones básicas

Suma, resta, multiplicación, división. También mostrar el resultado.

**Diseño:**

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA: CALCULADORA |
| Variables:  num1: Entero  num2: Entero  suma: Entero  resta: Entero  multiplicación: Entero  división: Entero |
| Nombre del Algoritmo: opBasicas |
| **Proceso del algoritmo:**   1. inicio 2. Leer num1 3. Leer num2 4. suma ← num1 + num2 5. mostrar ← “el resultado de la suma es: “ + suma 6. resta ← num1 – num2 7. mostrar ← “el resultado de la resta es: “ + resta 8. multiplicación ← num1 \* num2 9. mostrar ← “el resultado de la multiplicación es: “ + multiplicación 10. división ← num1 / num2 11. mostrar ← “el resultado de la división es: “ + división 12. *Fin* |

# Captura 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



Definición del problema: convertir una temperatura Fahrenheit a grados Celsius

**Análisis:**

* Datos de entrada:

Fahrenheit: Real

* Datos de salida:

Celsius

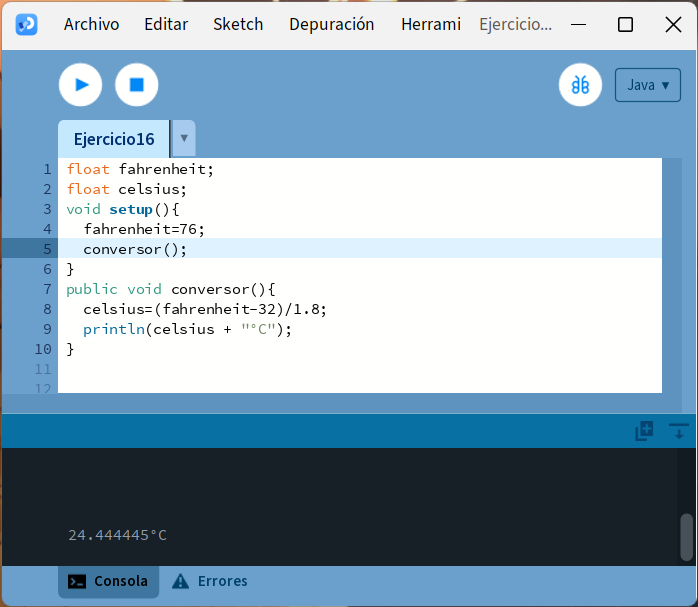
* Proceso:

¿Quién debe realizar el proceso?: Estudiante

¿Cuál es el proceso que realiza?

Pasar de grados Fahrenheit a grados Celsius, utilizando la siguiente

Formula: temperaturaCelsius = temperaturaFahrenheit-32/1.8

Diseño:

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA: estudiante |
| Variables  Fahrenheit: Real  Celsius: Real |
| NOMBRE DEL ALGORITMO: conversor |
| PROCESO DEL ALGORITMO  *Leer* Fahrenheit  Celsius ← (Fahrenheit-32)/1.8;  *Mostrar* Fahrenheit |

Captura Processing

# 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

**Definición del Problema:** determinar la distancia entre link y un tesoro

**Análisis:**

**•Datos de Entrada:**

posicionLink:real(PVector)

posicionTesoro: real(PVector)

lados:entero

**•Datos de Salida:**

distancia

**•Proceso:**

¿Quién debe realizar el proceso?: programador

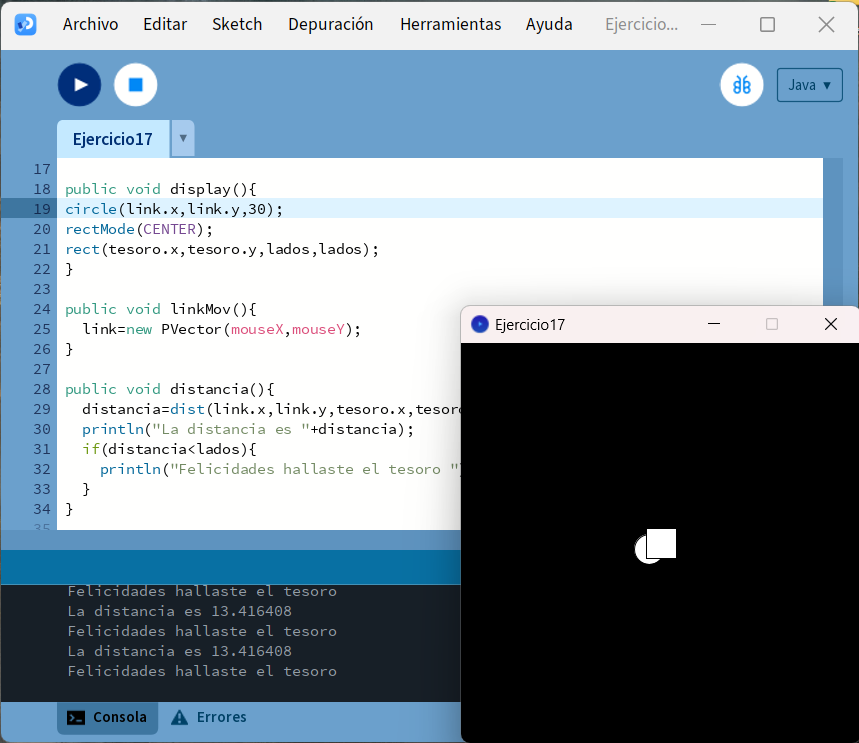
¿Cuál es el proceso que realiza …?

Con la posición x e y del jugador(Link) determinar la distancia entre la posición x e y del tesoro

**Diseño:**

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA : **programador** |
| **VARIABLES**  **lInk.x : real**  **link.y : real**  **tesoro.y : real**  **distancia : real**  **tesoro.x:Real**  **lados : entero** |
| Metodo setup  PROCESO DEL ALGORITMO   1. Inicio 2. Lienzo(400,400) 3. lados30 4. tesoro.xwidth/2 5. tesoro.yheight/2 6. *Fin* |
| Metodo draw  PROCESO DEL ALGORITMO   1. Incio 2. linkMov 3. display 4. distancia |
| NOMBRE ALGORITMO: display  PROCESO DEL ALGORITMO   1. inicio 2. circle posicionLink.x,posicionLink.y,30 3. rectMode CENTER 4. rect posicionTesoro.x,posicionTesoro.y,lados,lados 5. Fin |
| NOMBRE ALGORITMO: linkMov  PROCESO DEL ALGORITMO   1. inicio 2. lInk.xmouseX 3. link.ymouseY 4. *Fin* |
| NOMBRE ALGORITMO: distancia  PROCESO DEL ALGORITMO   1. inicio 2. distanciadist (link.x,link.y,tesoro.x,tesoro.y) 3. mostrar "La distancia es "+distancia 4. **si** distancia<lados **entonces** 5. mostrar "Felicidades hallaste el tesoro " 6. **fin\_si** 7. **fin** |

Captura 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.

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

**Análisis:**

**•Datos de Entrada:**

A,b,c,d:Entero

discriminante:Real

x1,x2:Real

**•Datos de Salida:**

resultado

**•Proceso:**

¿Quién debe realizar el proceso?: programador

¿Cuál es el proceso que realiza …?

**Diseño:**

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA: **programador** |
| **VARIABLES**  a: Entero  b: Entero  c: Entero  d: Entero  discriminante:Real;  x1,x2:Real; |
| NOMBRE ALGORITMO: discriminante  PROCESO DEL ALGORITMO   1. Inicio 2. a1 3. b-2 4. c1 5. discriminante(b^2)-((4\*a)\*c) 6. **si** discriminante <0 **entonces** 7. mostrar “no se puede” 8. **si\_no** 9. **si** discriminante >0 entonces 10. x1(-b-discriminante)/(2\*a) 11. mostrar x1 12. x(-b+discriminante)/(2\*a) 13. mostrar x2 14. si\_no 15. si discriminante = 0 entonces 16. x1(-b)/(2\*a) 17. mostrar x1 18. **fin\_si** 19. *Fin* |

# 

# 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 bac kground(). 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

**Definición del Problema**: dibujar una línea con una ellipse que se extienda a lo largo del lienzo. cuando la línea supere la posición de la altura del lienso debe invertir su sentido ,es decir dirigirse hacia arriba arrastrando la elipse

**Análisis:**

**• Datos de Entrada:**

pocicionY:entero

colicion: entero

tamaño:entero

reversa:entero:

**• Datos de Salida:**

Dibujo de colisión

**• Proceso:**

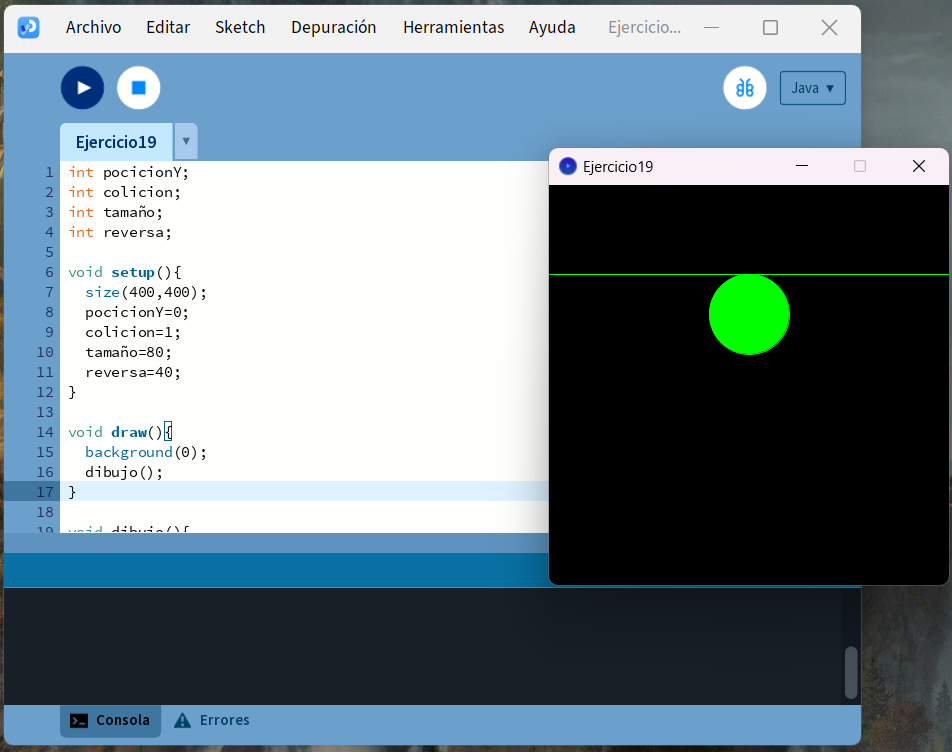
¿Quién debe realizar el proceso?: programador

¿Cuál es el proceso que realiza …?

Cuando la línea llegue a la altura máxima o mínima del lienzo se debe actualizar la dirección arrastrando la elipse

**Diseño:**

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA: **programador** |
| **VARIABLES**  pocicionY : Entero  colicion:Entero  tamaño:Entero  reversa:Entero |
| NOMBRE ALGORITMO: colisión    PROCESO DEL ALGORITMO   1. Inicio 2. pocicionY0 3. colicion1 4. tamaño80 5. reversa40 6. pocicionY+colicion 7. line(width,pocicionY,0,pocicionY) 8. ellipse(width/2,pocicionY+reversa,tamaño,tamaño); 9. **si** pocicionY<0 || pocicionY>height **entonces** 10. colicion\*-1 11. reversa\*-1 12. **fin\_si** 13. ***Fin*** |

**Captura Processing**

# 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.

**Definición del problema:** Dibujar rectángulos de idénticas medidas que mantengan una distancia de 20 pixeles entre ellos tanto horizontal como vertical

**Análisis:**

* **Datos de entrada:**

ancho: Entero

alto: Entero

espacioRectangulo: Entero

posicionRectangulo: PVector

* **Datos de Salida:**

dibujarRectángulo

* **Proceso:**

¿Quién debe realizar el proceso?: Estudiante

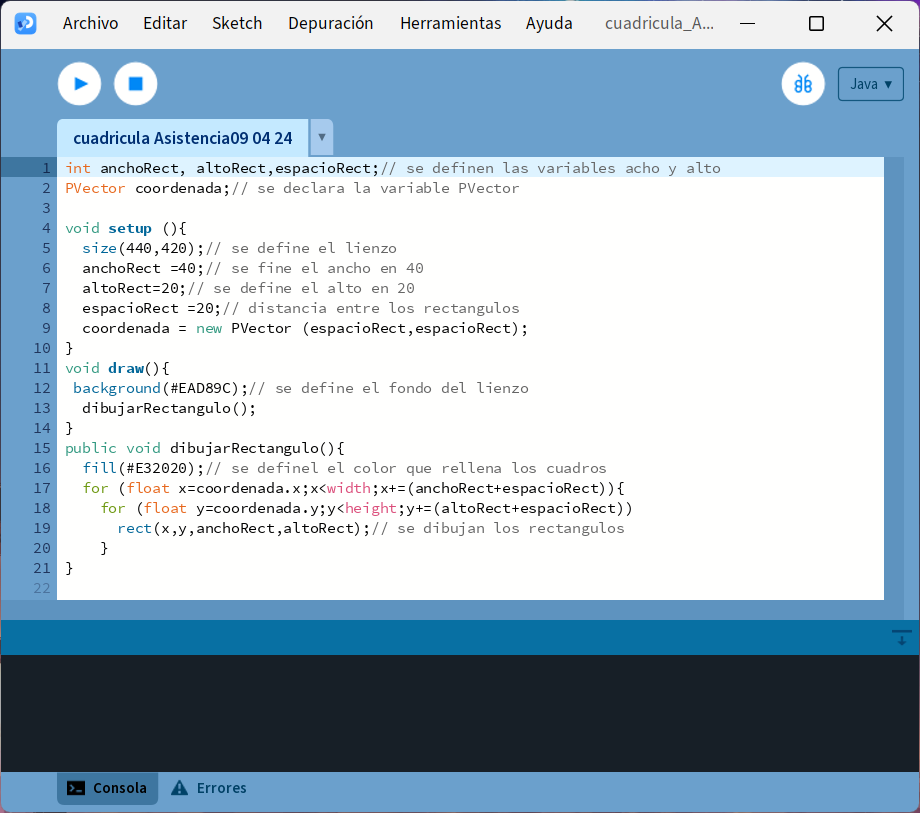
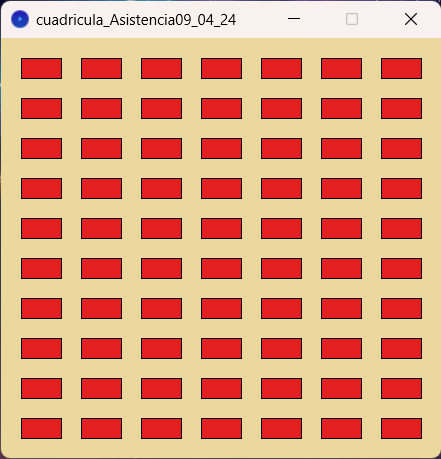
¿Cuál es el proceso que realiza…?

Dibuja una cuadrícula de rectángulos, donde cada rectángulo tiene dimensiones definidas y un espacio entre ellos, utilizando la estructura interactiva (For)

**Diseño:**

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL PROBLEMA**: Estudiante |
| **VARIABLES**  Ancho :Entero//almacena el ancho de los rectángulos  alto:Entero//almacena el alto de los rectángulos  espacioRectangulo:Entero//almacena la distancia entre rectángulo  pocicionRectangulo:PVector//almacena la posición x ,y |
| **NOMBRE ALGORITMO**: cuadricula  **PROCESO DEL ALGORITMO**   * + - 1. *inicio*       2. ancho←40       3. alto←20       4. distanciaRectangulo←20       5. posicionRectangulo← espacioRectangulo , espacioRectangulo       6. dibujarRectangulo ←**para** x ←posicionRectangulo.x **hasta** x<width **incremento** x+=(ancho+ espacioRectangulo) **hacer**       7. **para** y ←posicionRectangulo.y **hasta** y<height **incremento** y+=(alto+ espacioRectangulo) **hacer**       8. rect(x,y,ancho,alto)       9. **fin\_para**       10. **fin\_para**       11. *Mostrar*dibujarRectangulo       12. **fin** |

## **Captura de pantalla, processing**



# 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

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=entero

y = entero

ancho=entero

alto=entero

**•Datos de Salida:**

DibujoEscalera

**•Proceso:**

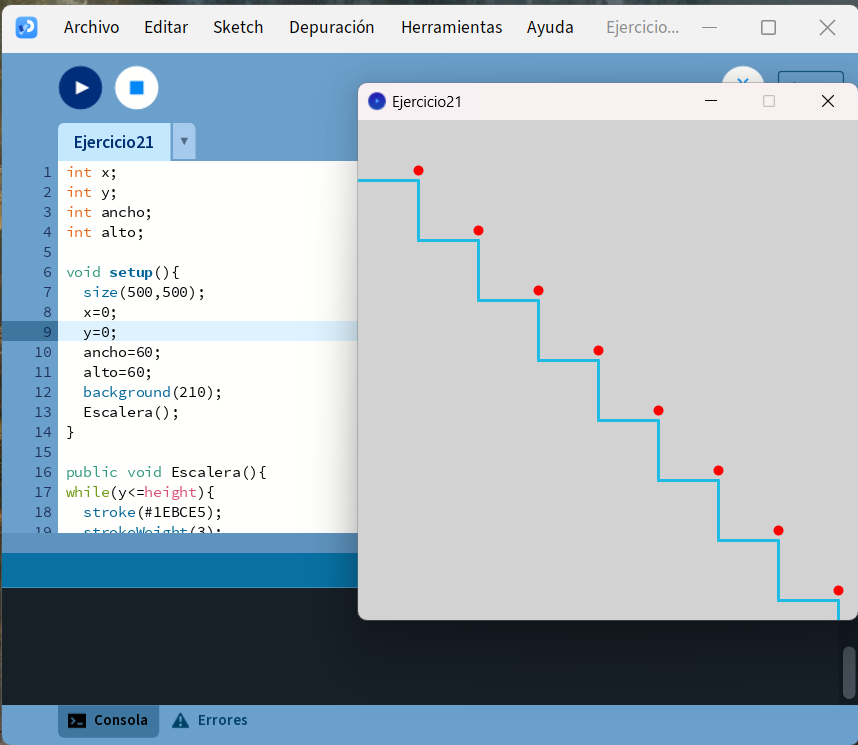
¿Quién debe realizar el proceso?: programador

¿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

**Diseño:**

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA: **programador** |
| **VARIABLES**  x:Entero  y:Entero  ancho=entero  alto=entero |
| NOMBRE ALGORITMO: escalera  PROCESO DEL ALGORITMO   1. Inicio 2. x0 3. y0 4. ancho60 5. alto60 6. mientras (y<height) hacer 7. line(x,y+alto,x+ancho,y+ancho) 8. line(x+ancho,y+ancho,x+ancho,y+(2\*alto)) 9. point(x+ancho,y+alto-10) 10. x+=ancho 11. y+=alto 12. Fin\_mientras 13. *Fin* |

Captura Processing

# 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

**Definición del Problema:** dibujar los círculos sobre cada línea de por medio con círculos con distanciamiento

**Análisis:**

**•Datos de Entrada:**

distanciaLlinia :Entero

y:Entero

espacioCirculo:Entero

**•Datos de Salida:**

Circulos

**•Proceso:**

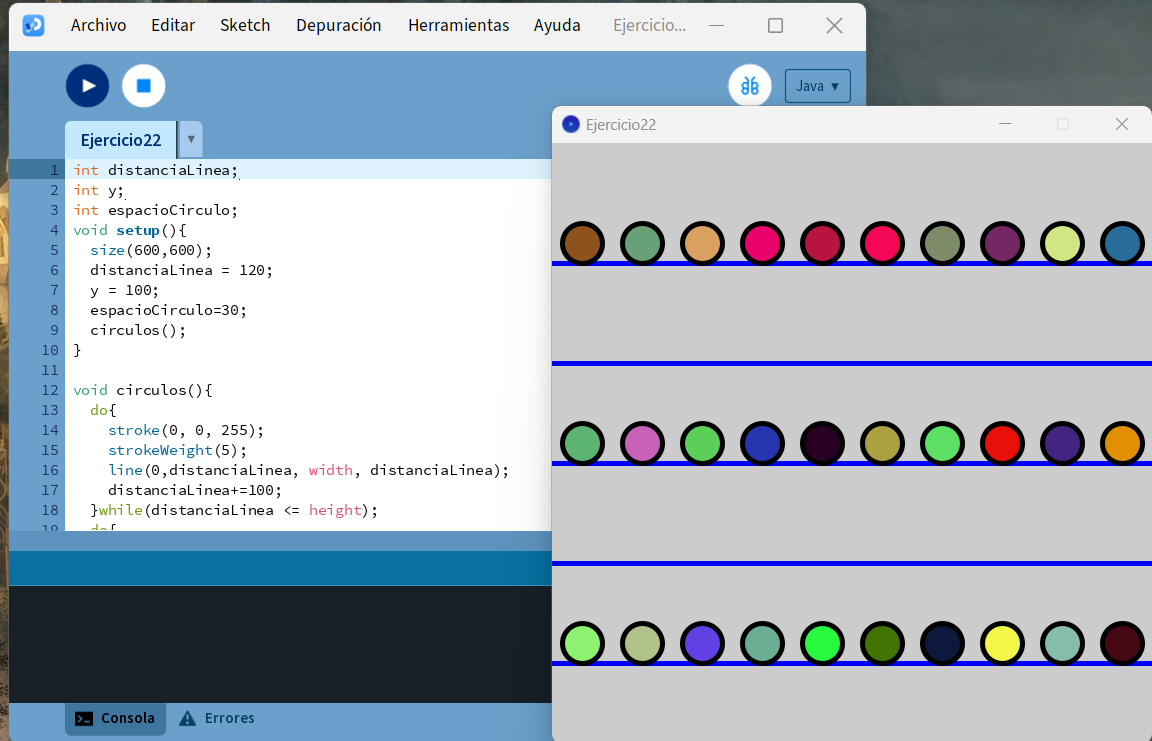
¿Quién debe realizar el proceso?: programador

¿Cuál es el proceso que realiza …?

**Diseño:**

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA: **programador** |
| **VARIABLES**  distancialinea :Entero  y:Entero  espacioCirculo:Entero |
| NOMBRE ALGORITMO: Escalera  PROCESO DEL ALGORITMO   1. Inicio 2. distancialinea 120 3. Y0 4. espacioCirculo30 5. **hacer** 6. line(0,espaciolinea, width, distanciaLinea) 7. distanciaLinea+100 8. **Mientras** (distanciaLinea <= height) 9. **Hacer** 10. **Para** xespacioCirculo **hasta** x<widht **incremento** x+=60 **hacer** 11. ellipse(x, posicionY, 40, 40) 12. **fin\_para** 13. Y+200 14. **Mientra** (Y<=height) 15. *Fin* |

Captura Processing



# Conclusión

Trabajo practico extenso, que no logre terminar a tiempo, el análisis y diseño me costó bastante. Pero quedé satisfecho con lo que hice.

**Fuentes bibliográficas**

Semana 3 y 4 del aula virtual

**20 Estructura for en Processing**

**21 Ejemplo de for anidado**

22 Video El manejo de las imágenes en Processing

25 Manejo de la posición Mouse en Processing

26 Manejo del mouse en Processing