|  |  |  |
| --- | --- | --- |
| 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*

*Año 2024*

Trabajo Práctico

N°1

Ibañez Mario Ezequiel

TUV000617

**FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS**

Indice

Ejercicio 1:

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

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

Desarrollo del Ejercicio

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

6-(4\*B/4)

6-5

1

Captura de pantalla



Ejercicio 2

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

Desarrollo del Ejercicio 2

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

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

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

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

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

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

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

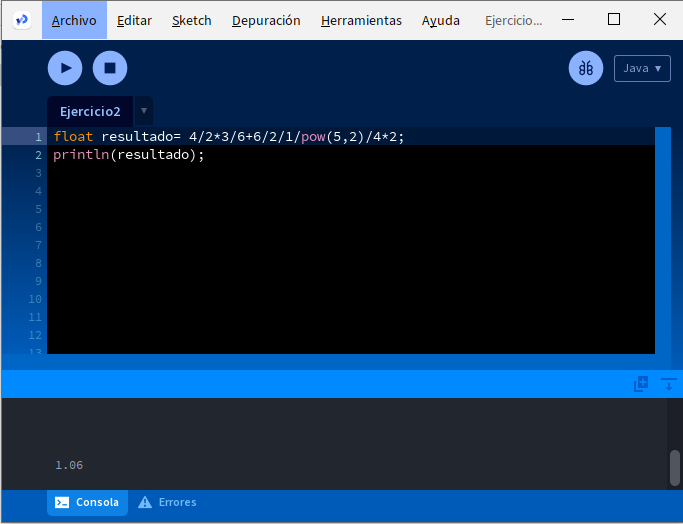
1+ (0.12 / 4) \* 2

1+ 0.03 \* 2

1+ 0.06

1.06

Captura de pantalla



Ejercicio 3

Desarrollo del Ejercicio 3

Ejercicio 4

Desarrollo del Ejercicio 4

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

Desarrollo del Ejercicio 5

a) (5\*4) -(5^2) / (4\*1)

(20) -(25/4)

20-6.25

13.75

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

20/9

2.2222223

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

(((6/2) \*4+10) \*15)-6

((3\*4+10) \*15)-6

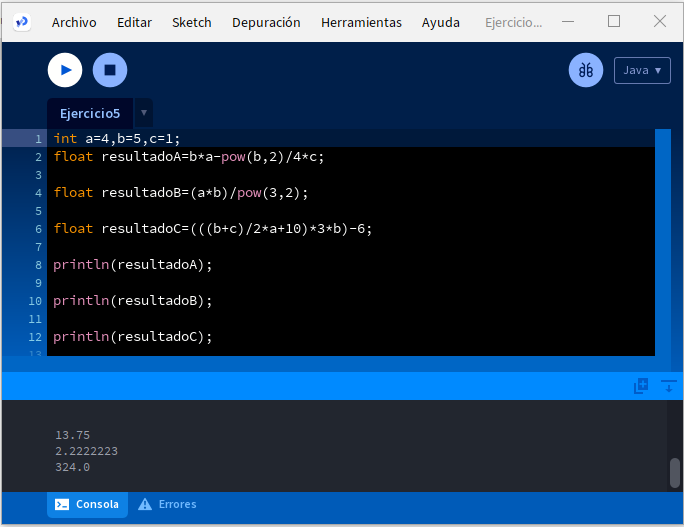
((12+10) \*15)-6

(22\*15)-6

330-6

324

Captura de pantalla



Ejercicio 6

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

Desarrollo del ejercicio 6

R1=4+1

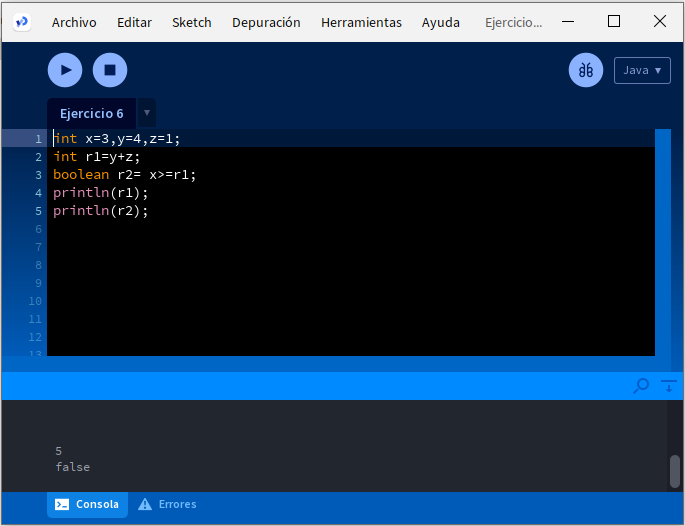
R1=5

R2 =3>=R1

R2=3>=5

R2=false

Captura de pantalla



Ejercicio 7

Para contador1=3, contador3=4, evaluar el resultado de

R1 = ++contador1

R2 = contador1 < contador3

Desarrollo del ejercicio 7

R1=++contador1

R1=1+3

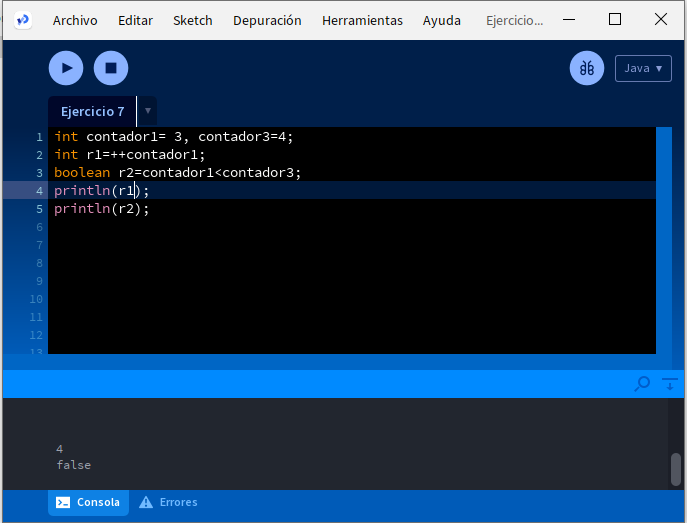
R1=4

R2=contador<contador3

R2=4<4

R2=false

Captura de pantalla



Ejercicio 8

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

a+b-1 < x\*y

Desarrollo del ejercicio 8

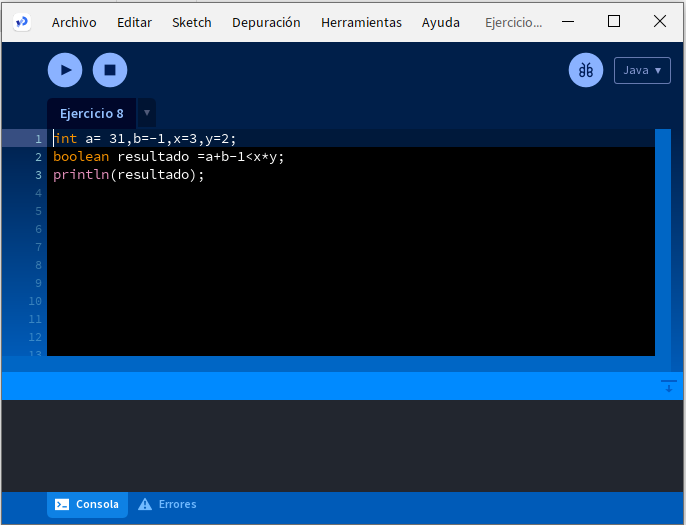
a+b-1 < x\*y

31+1-1 <3\*2

31 < 6

False

Captura de pantalla



Ejercicio 9

Para x=6, y=8, evaluar elresultado de

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

Desarrollo del ejercicio 9

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

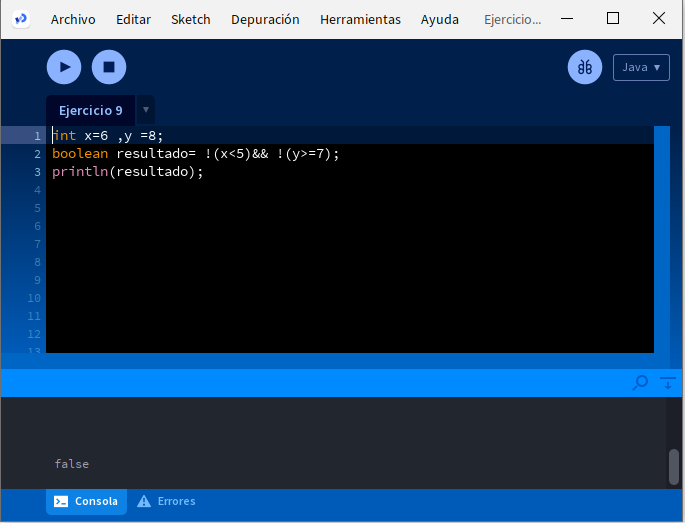
!(6<5)&&!(8>=7)

!(verdadero) && !(verdadero)

Falso && falso

falso

Captura de pantalla



Ejercicio 10

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

Desarrollo del Ejercicio 10

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

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

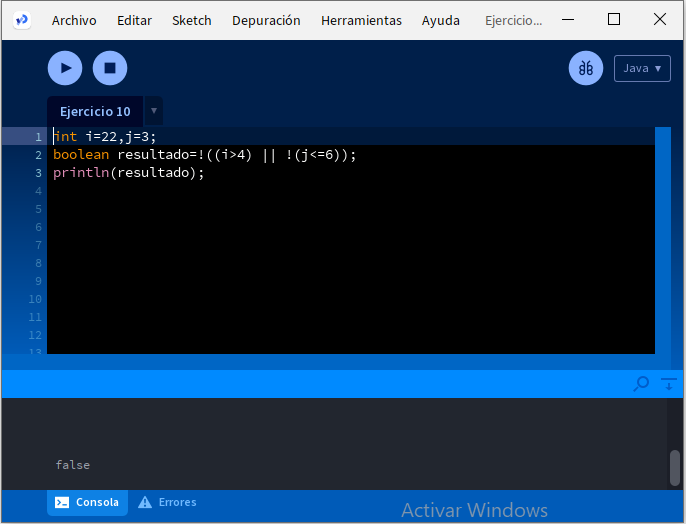
!((verdadero) || !(verdadero))

!((verdadero) || falso)

!(verdadero)

Falso

Captura de pantalla



Ejercicio 11

Para a=34, b=12, c=8, evaluar el resultado de !(a+b==c) || (c!=0)CC(b-c>=19)

Desarrollo del ejercicio 11

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

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

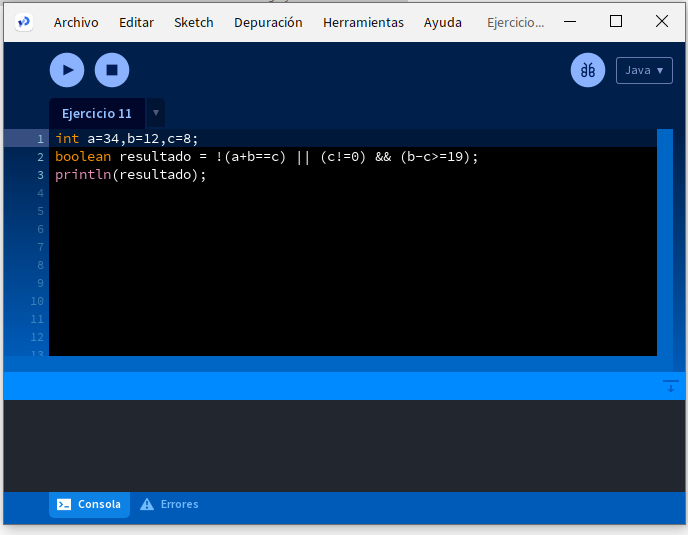
!(falso)||(verdadero)&&(falso)

!(falso )||falso)

!(falso)

verdadero

Captura de pantalla



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:

menjaseBienvenida:

* Proceso:

¿Quién debe realizar el proceso?: progrmador

¿Cuál es el proceso que realiza …?

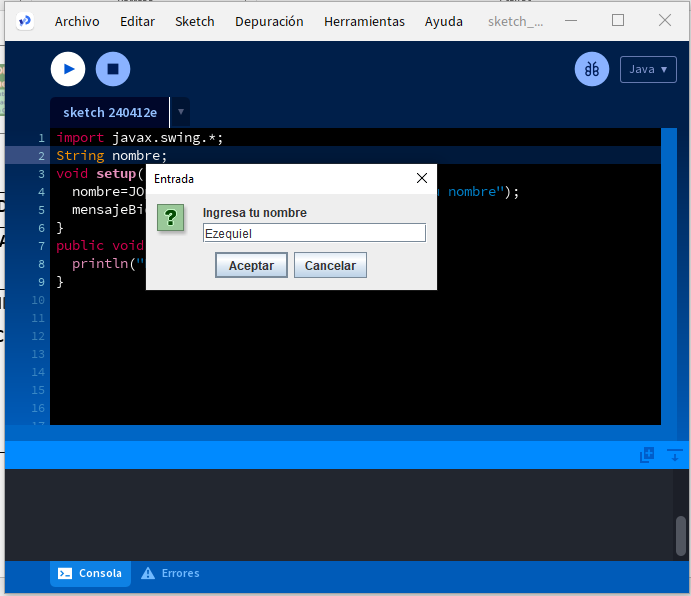
En una ventana se debe pedir el nombre de la persona para después

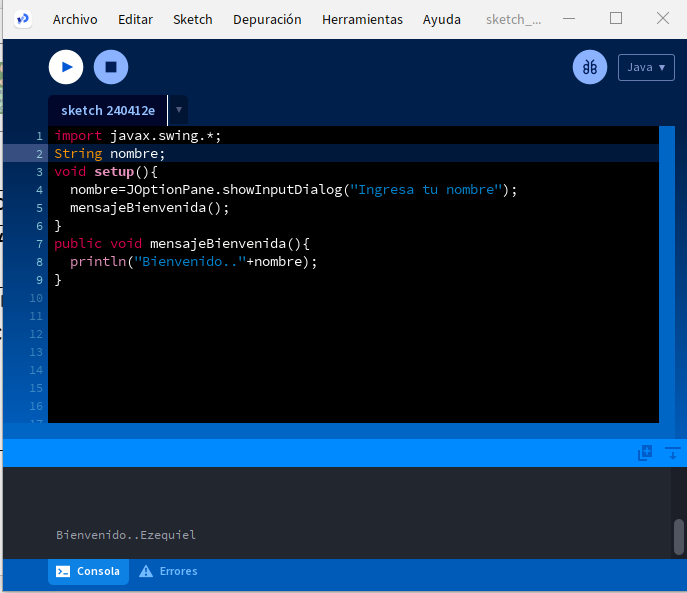
saludarlo con un mensaje de Bienvenida

“Bienvenido..”+nombre

**Diseño**:

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL PROBLEMA**: Programador |
| **VARIABLES**  Nombre=String://se almacena el nombre de la persona |
| **NOMBRE ALGORITMO**: mensajeBienvenida  **PROCESO DEL ALGORITMO**   * + - 1. **inicio**       2. Mostrar “Ingresa tu nombre”       3. Leer nombre       4. mensajeBienvenida ← “bienvenido..”+nombre       5. mostrar mensajeBienvenida       6. **fin** |

**Captura de pantalla**



Ejercicio 13

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

**Definición del Problema**: Calcular el perímetro y área de un rectángulo dada su base y su altura

**Análisis**:

* Datos de Entrada:

Altura: Real

Base : Real

* Datos de Salida:

Perimetro

Area

* Proceso:

¿Quién debe realizar el proceso?: progrmador

¿Cuál es el proceso que realiza …?

Para el perímetro usamos la siguiente formula

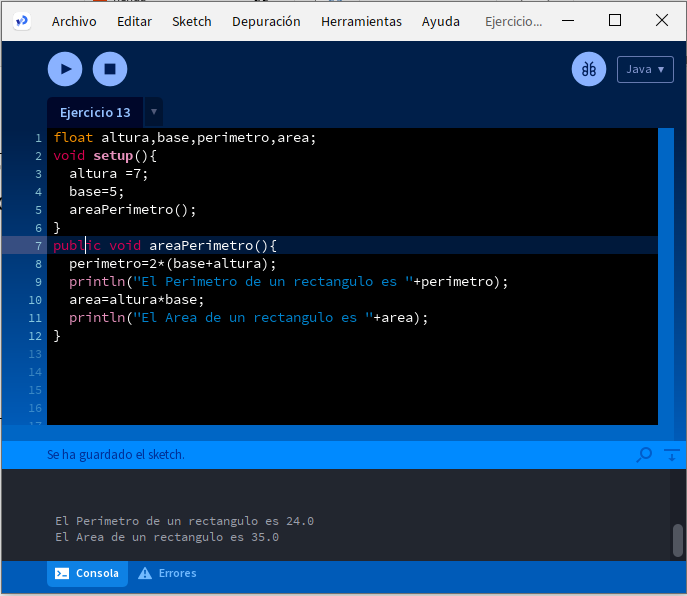
Perimetro= 2 \*(base+altura)

Y para el área usamos la siguiente formula

Área=base\*altura

**Diseño**:

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL PROBLEMA**: Programador |
| **VARIABLES**  Altura:Real//almacena número real  Base: Real //almacena los numero real  Perímetro : Real // almacena los resultados de perímetro=2\*(base+altura)  Área : real // almacena los resultados del área=base\*altura |
| **NOMBRE ALGORITMO**: areaPerimetro  **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 de pantalla**

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

**Definición del Problema**: Obtener la hipotenusa

**Análisis:**

**•Datos de Entrada:**

Base:Entero

Altura:Entero

**•Datos de Salida:**

hipotenusa:Real

**•Proceso:**

¿Quién debe realizar el proceso?: usuario

¿Cuál es el proceso que realiza …?

Para sacar la hipotenusa se debe sacar la raíz cuadrada de base^2 y altura ^2

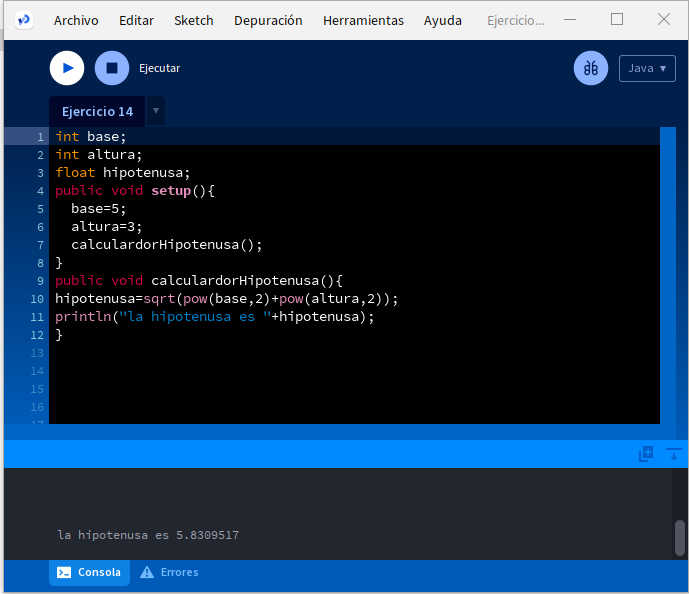
Haci aplicado el teorema de Pitágoras

Teorema de Pitágoras←

**Diseño:**

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA: Programador |
| **VARIABLES**  Base:Entero// almacena la base del triangulo  Altura:Entero// almacena la altura del triangulo  hipotenusaTriangulo:Real // almacena la hipotenusa del triangulo |
| NOMBRE ALGORITMO: calculardorHipotenusa  PROCESO DEL ALGORITMO   1. inicio 2. Base← 4 3. Altura← 3 4. hipotenusa ← 5. *Mostrar* hipotenusa 6. *Fin* |

Captura de pantalla



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:**

numeroA:Entero

numeroB: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 NumeroA+ NumeroB

resultadoRestar NumeroA- NumeroB

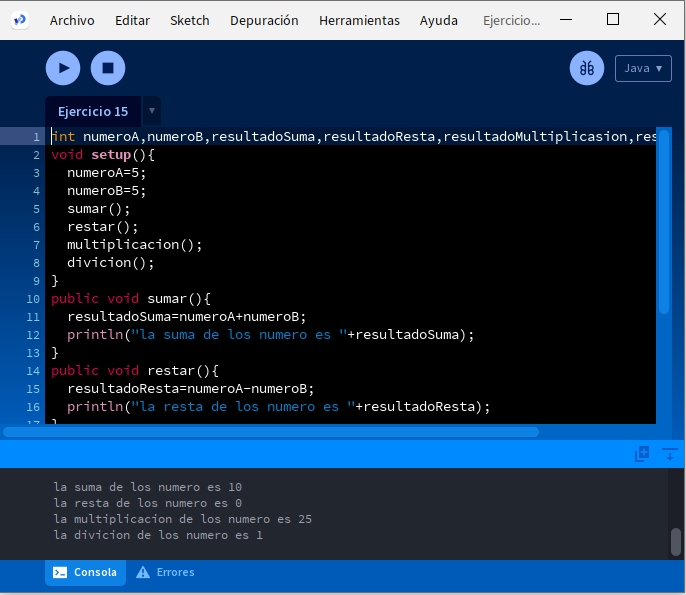
resultadoMultiplicar NumeroA\* NumeroB

resultadoDividir NumeroA/ NumeroB

**Diseño:**

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA : **programa** |
| **VARIABLES**  NumeroA, NumeroB: 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. NumeroA5 3. NumeroB5 4. resultadoSuma NumeroA + NumeroB 5. *Mostrar* resultadoSuma 6. *Fin* |
| PROCESO DEL ALGORITMO restar   1. Inicio 2. NumeroA5 3. NumeroB5 4. resultadoRestar NumeroA- NumeroB 5. *Mostrar* resultadoRestar 6. Fin |
| NOMBRE ALGORITMO: multiplicar  PROCESO DEL ALGORITMO   1. inicio 2. NumeroA5 3. NumeroB5 4. resultadoMultiplicar NumeroA \* NumeroB 5. *Mostra*r resultadoMultiplicar 6. Fin |
| PROCESO DEL ALGORITMO dividir   1. inicio 2. NumeroA5 3. NumeroB5 4. resultadoDividir NumeroA / NumeroB 5. *Mostrar* resultadoDividir 6. *Fin* |

Captura de pantalla



Ejercicios 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 en grados Celsius

**Análisis**:

* Datos de Entrada:

TemperaturaFahrenheight:real

* Datos de Salida:

TemperaturaCelcius

* Proceso:

¿Quién debe realizar el proceso?: usuario

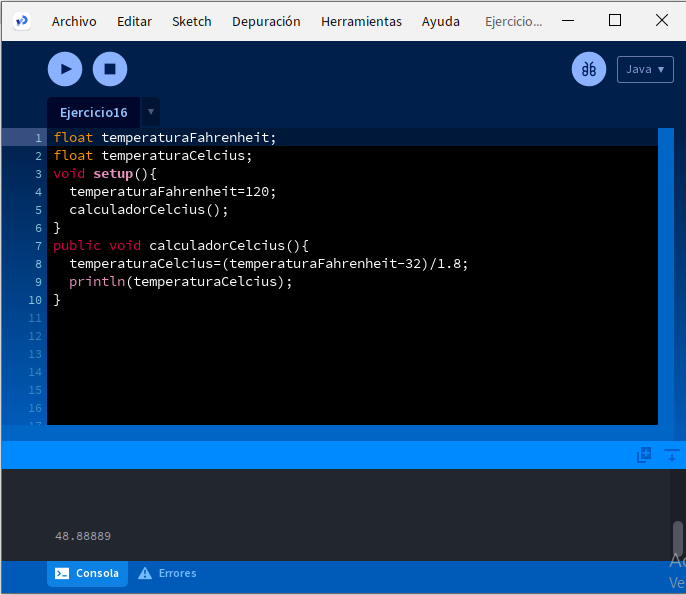
¿Cuál es el proceso que realiza …?

Ala temperatura Fahrenheit le restamos 32 y después lo dividimos en 1.8 que dará la temperatura Celsius

temperaturaCelcius=(temperaturaFahrenheit-32)/1.8;

**Diseño**:

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL PROBLEMA**: |
| **VARIABLES**  temperaturaFahrenheit: Real //almacena el número de la temperatura  temperaturaCelcius: Real // almacena el re resultado en grados celcius |
| **NOMBRE ALGORITMO**: calculadorCelcius  **PROCESO DEL ALGORITMO**   * + - 1. *Leer* temperaturaFahrenheit       2. temperaturaCelcius ← (temperaturaFahrenheit-32)/1.8;       3. *Mostrar* temperaturaFahrenheit |

Captura de pantalla

**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

**PLANTILLA EJERCICIOS**

**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?: juego

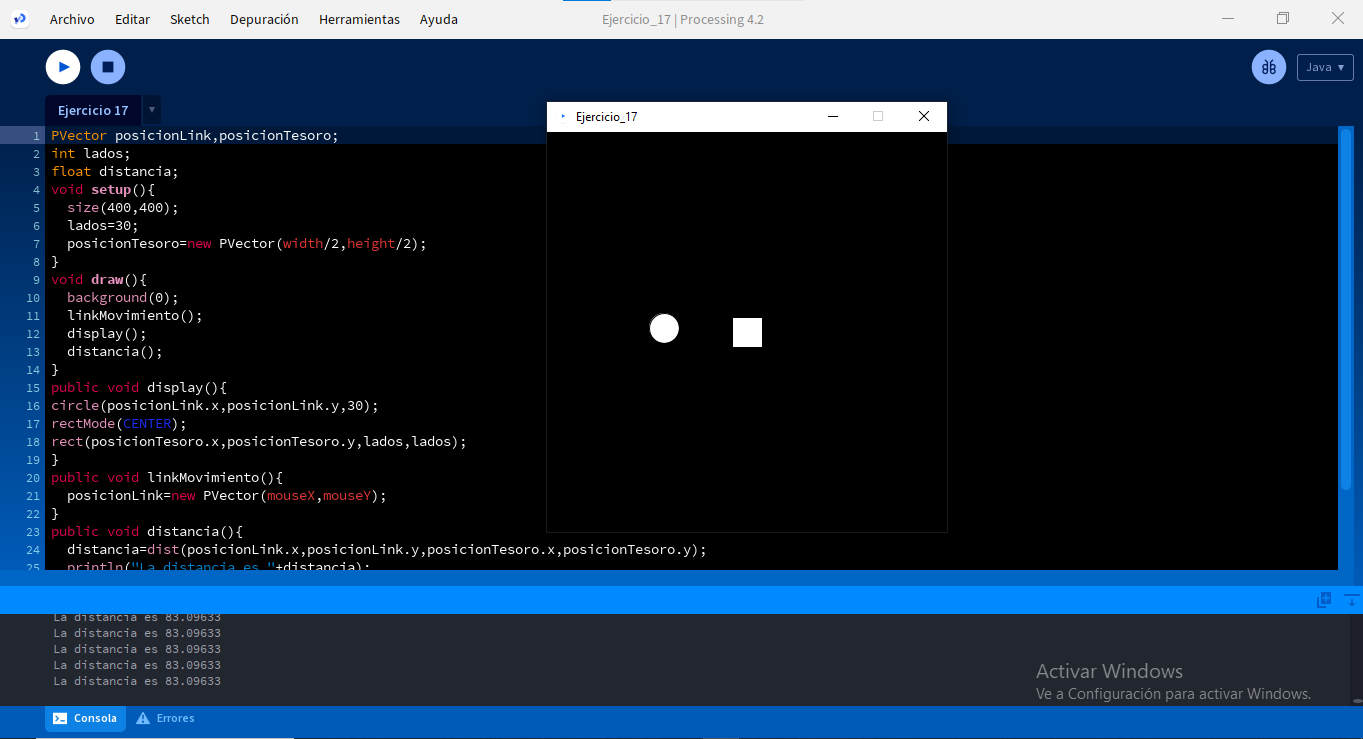
¿Cuál es el proceso que realiza …?

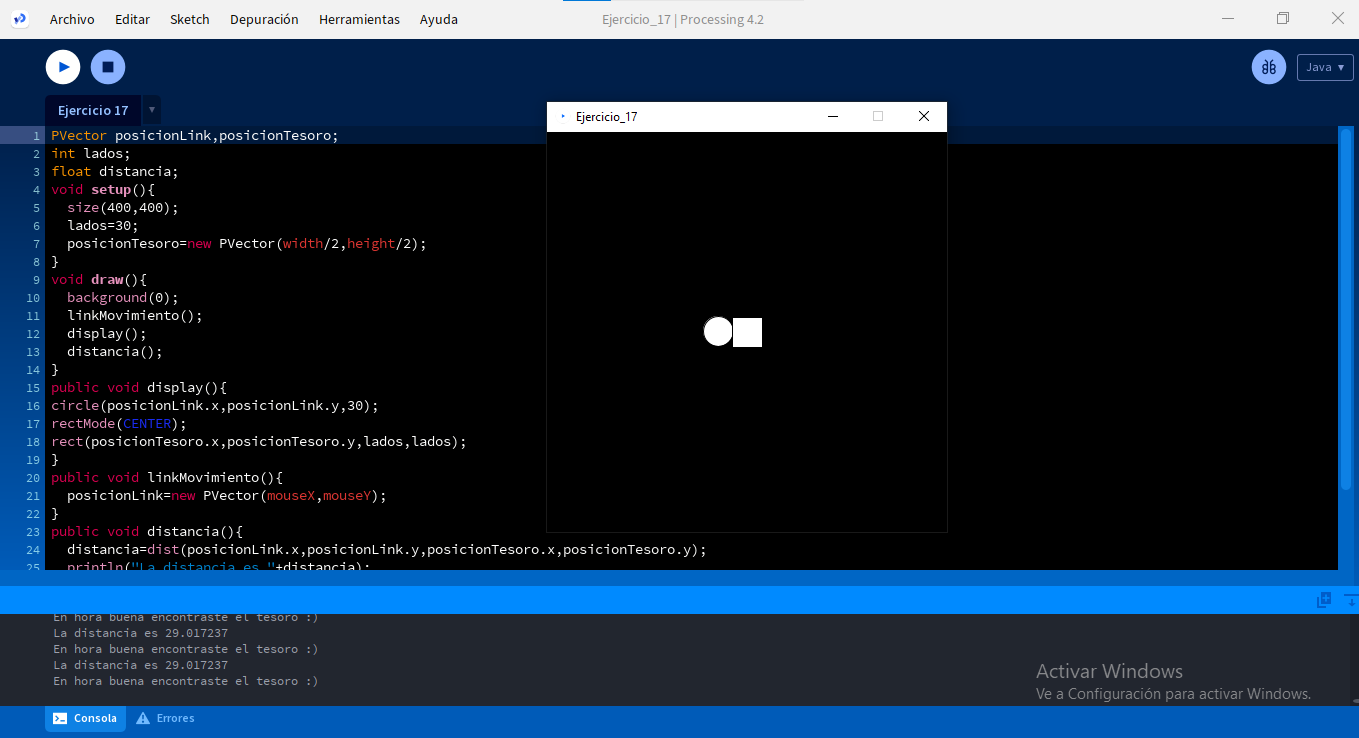
Con la posición x e y del jugador(Link) determinar la distancia entre la posición x e y del tesoro

**Diseño:**

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA : **programa** |
| **VARIABLES**  posicionLInk.x : real//almacena la posición x del jugador (Link)  posicionLink.y : real//almacena la posición y de jugador(Link)  posicionTesoro.x : real//almacena la posición x del tesoro  posicionTesoro.y : real //almacena la posición y del tesoro  distancia : real // almacena la distancia entre el jugador (LInk)y tesoro  lados : entero // almacena los lados del rectángulo |
| Metodo setup  PROCESO DEL ALGORITMO   1. Inicio 2. Lienzo(400,400) 3. lados30 4. posicionTesoro.xwidth/2 5. posicionTesoro.yheight/2 6. *Fin* |
| Metodo draw  PROCESO DEL ALGORITMO   1. Incio 2. linkMovimiento 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: linkMovimiento  PROCESO DEL ALGORITMO   1. inicio 2. pocicionLInk.xmouseX 3. posicionLink.ymouseY 4. *Fin* |
| NOMBRE ALGORITMO: distancia  PROCESO DEL ALGORITMO   1. inicio 2. distanciadist (posicionLink.x,posicionLink.y,posicionTesoro.x,posicionTesoro.y) 3. mostrar "La distancia es "+distancia 4. **si** distancia<lados **entonces** 5. mostrar "En hora buena encontraste el tesoro :) " 6. **fin\_si** 7. **fin** |

Captura de pantalla



****

**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

x1,x2:Real

**•Datos de Salida:**

resultado

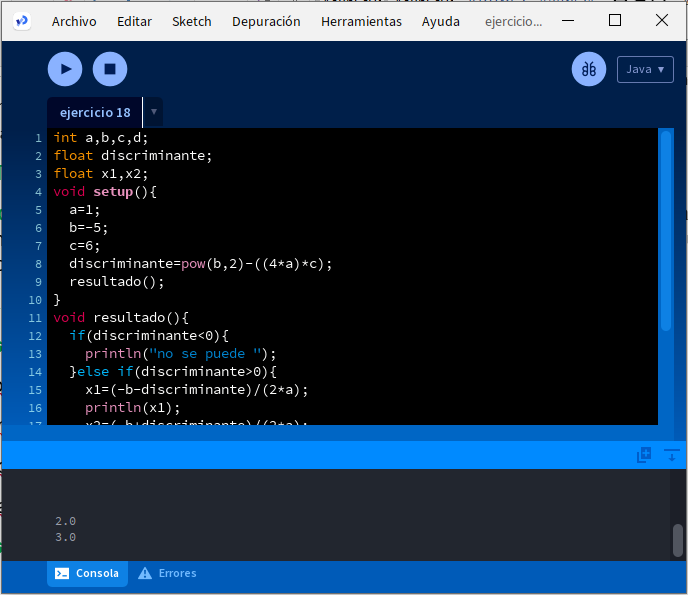
**•Proceso:**

¿Quién debe realizar el proceso?: programador

¿Cuál es el proceso que realiza …?

**Diseño:**

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA: **programador** |
| **VARIABLES**  a,b,c,d:Entero;  discriminante:Real;  x1,x2:Real; |
| NOMBRE ALGORITMO: resultado  PROCESO DEL ALGORITMO   1. Inicio 2. a1 3. b-5 4. c6 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* |

Captura de pantalla

**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

PLANTILLA EJERCICIOS

**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

**•Proceso:**

¿Quién debe realizar el proceso?: programador

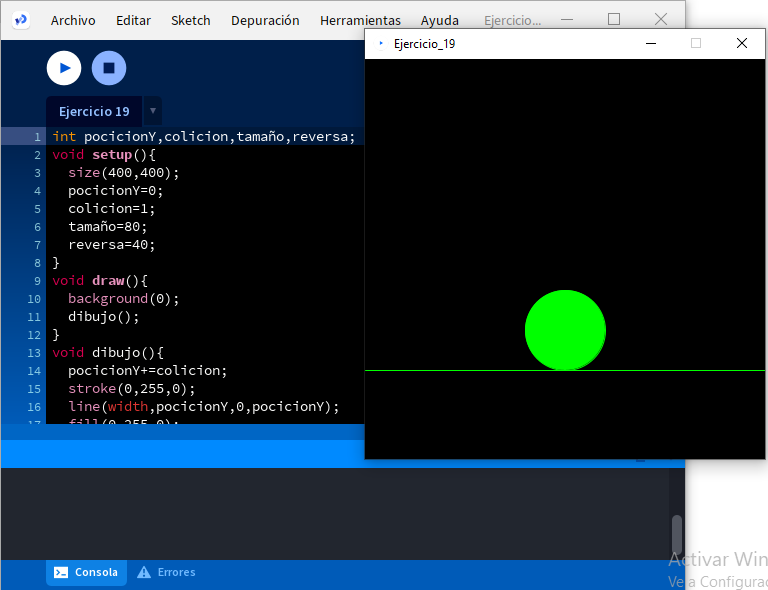
¿Cuál es el proceso que realiza …?

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: dibujo  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 de pantalla



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

**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;

distanciaRectangulo : Entero;

pocicionRectangulo : PVector;

* Datos de Salida:

DibujarRectangulo;

* Proceso:

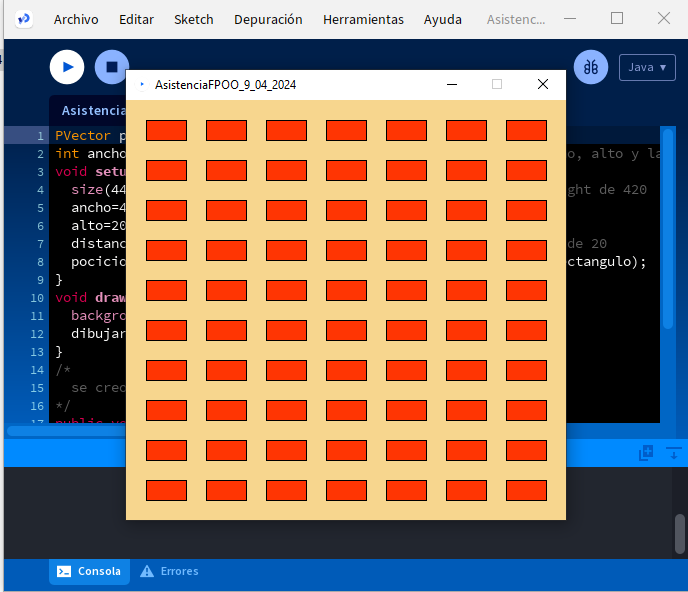
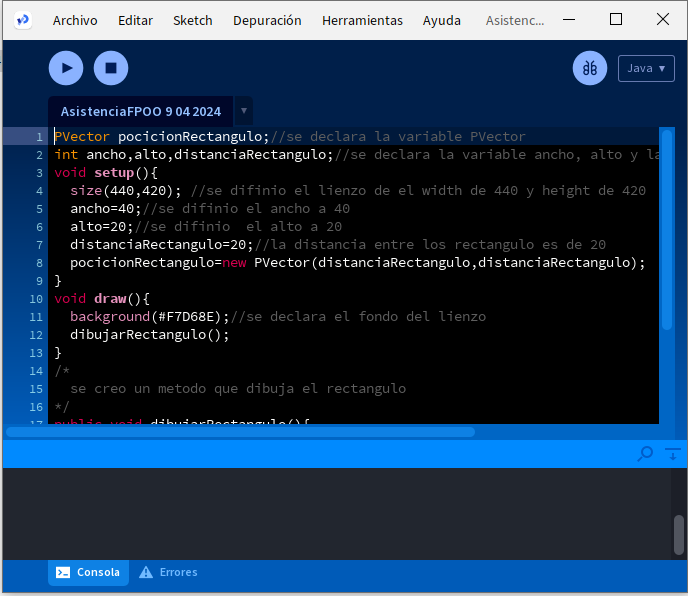
¿Quién debe realizar el proceso?: progrmador

¿Cuál es el proceso que realiza …?

Dibuja rectángulos con distancia de 20 de separación tanto como vertical como horizontal en todo el lienzo usando la estructura interactiva para(for)

**Diseño**:

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

Captura de pantalla

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

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=entero

y = entero

anchoEscalera=entero

altoEscalera=entero

**•Datos de Salida:**

DibujoEscalera

**•Proceso:**

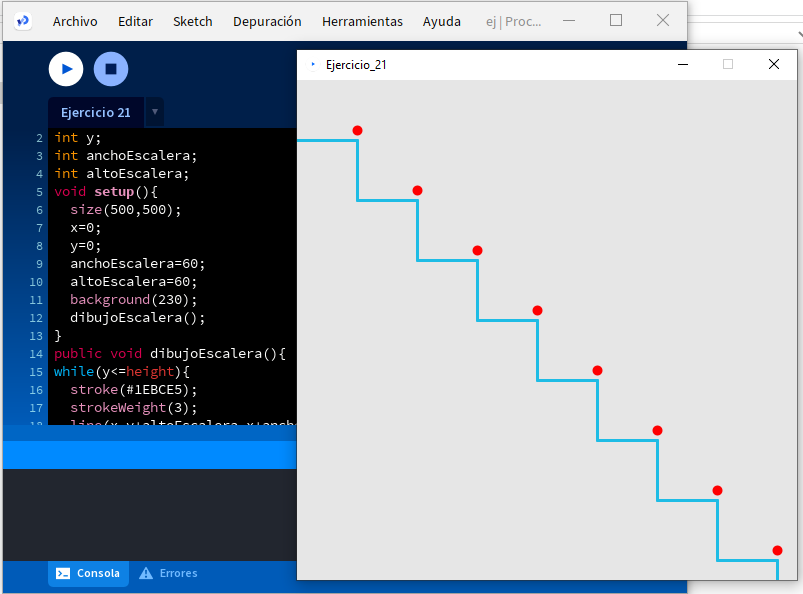
¿Quién debe realizar el proceso?: programador

¿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// almacena la base del triangulo  y:Entero// almacena la altura del triangulo  anchoEscalera=entero  altoEscalera=entero |
| NOMBRE ALGORITMO: dibujoEscalera  PROCESO DEL ALGORITMO   1. Inicio 2. x0 3. y0 4. anchoEscalera60 5. altoEscalera60 6. mientras (y<height) hacer 7. line(x,y+altoEscalera,x+anchoEscalera,y+anchoEscalera) 8. line(x+anchoEscalera,y+anchoEscalera,x+anchoEscalera,y+(2\*altoEscalera)) 9. point(x+anchoEscalera,y+altoEscalera-10) 10. x+=anchoEscalera 11. y+=altoEscalera 12. Fin\_mientras 13. *Fin* |

Captura de pantalla

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

**PLANTILLA EJERCICIOS**

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

**Análisis:**

**•Datos de Entrada:**

EspacioLlinia :Entero

posicionY:Entero

distanciaCirculo:Entero

**•Datos de Salida:**

DibujoCirculo

**•Proceso:**

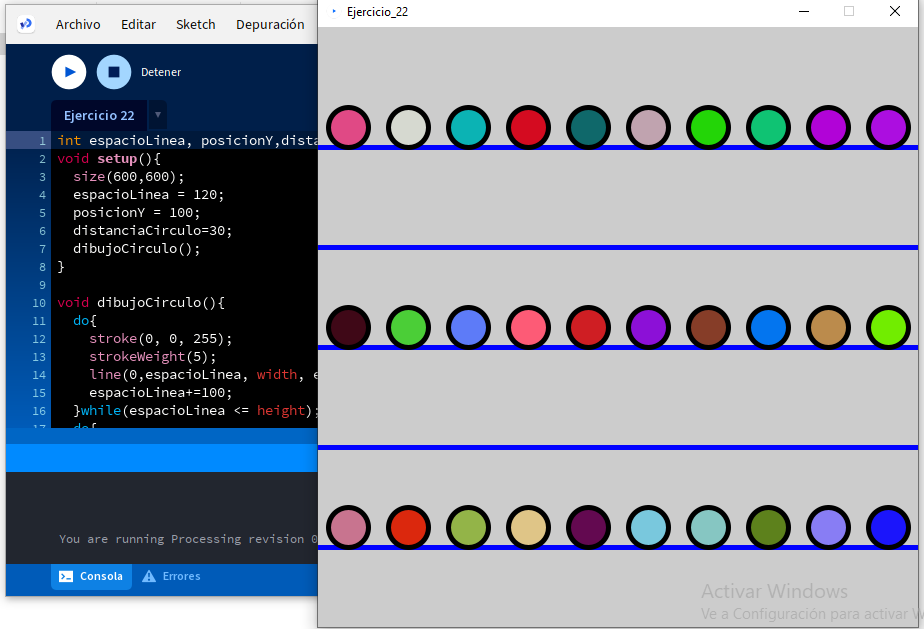
¿Quién debe realizar el proceso?: programador

¿Cuál es el proceso que realiza …?

**Diseño:**

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA: **programador** |
| **VARIABLES**  espacioLlinia :Entero  posicionY:Entero  distanciaCirculo:Entero |
| NOMBRE ALGORITMO: dibujoEscalera  PROCESO DEL ALGORITMO   1. Inicio 2. espacioLinia 120 3. posicionY0 4. distanciaCirculo30 5. **hacer** 6. line(0,espaciolinea, width, espaciolinea) 7. espaciolinea+100 8. **Mientras** (espacioLinea <= height) 9. **Hacer** 10. **Para** xdistanciaCirculo **hasta** x<widht **incremento** x+=60 **hacer** 11. ellipse(x, posicionY, 40, 40) 12. **fin\_para** 13. posicionY+200 14. **Mientra** (posicionY<=height) 15. *Fin* |

Captura pantalla



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)