**FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS**

Trabajo práctico/Actividad

N° 1

Apellido y Nombre – LU/

Ramos, Joel Axel/LU: TUV000670

Profesor:

Mg. Ing. Ariel Alejandro Vega

Año: 2024

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

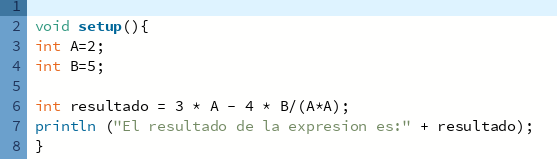
**3\* A - 4 \* B / A ^ 2**

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

6-(4\*B/4)

6-5

1



**Ejercicio 2:** Evaluar la siguiente expresión:

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

Aritmetica

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

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

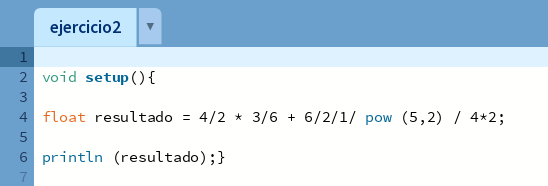
1. + (0,12 / 4 \* 2)

1 + (0,03 \* 2)

1. + (0,06)

1,06

Algrbraica

****

**Ejercicio 3**: Evaluar las siguientes expresiones aritméticas, para lo cual indicar en el caso de las variables, el valor indicado. Luego escribirlas como expresiones algebraicas. (Ignorar por el momento)

**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)

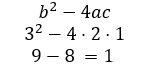
d) (x ^ 2 + y ^ 2) ^ (1 / 2)

Supongamos que:

b=3 a=2 y c=1 d=2

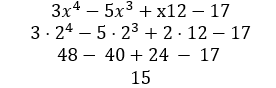
X=2 Y=4

Remplazamos los valores a la expresión y obtenemos:

1. b^2 – 4\*a\*c

(3^2) – (4\*2\*1)

9 – 8 = 1

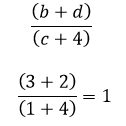


1. 3\*X^4 – 5 \*X ^ 3 + X\*12 – 17

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

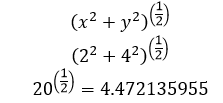
(3\*16) – (5 \*8) + 24 -17

48 – 40 + 7 = 15

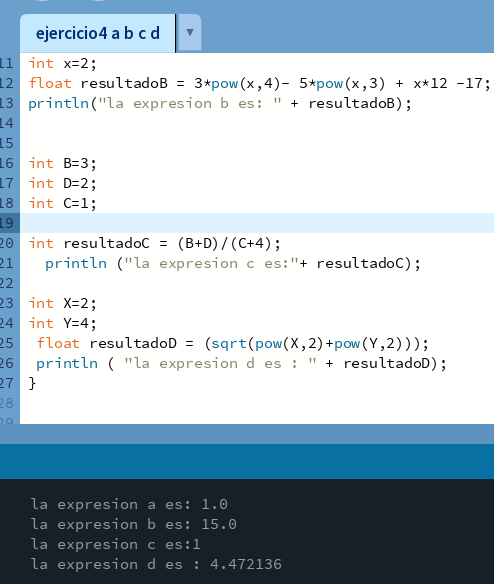
1. (b+d) / (c+4)

(3+2) / (1+4)

5/ 5 = 1

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

(2^ (2) + 4 ^ (2)) ^ (1/2) = 4.472135

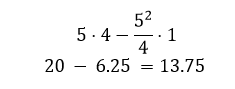
****

**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:

SI **A= 4** , **B=5** y **C=1**

**a)**

**B \* A – B ^ 2 / 4 \* C**

5 \* 4 – ((5 ^ 2) / 4) \* 1

20 - 6.25

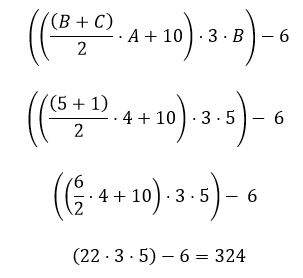
13.75

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

(4 \* 5) / 3 ^ 2 

2.2222 ...

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

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

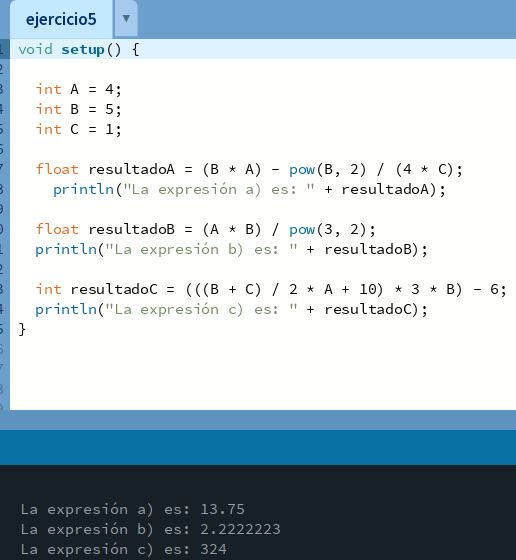
((6 / 2 \* 4 + 10) \* 3 \* 5) – 6

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

(22 \* 3 \* 5) - 6

330 – 6

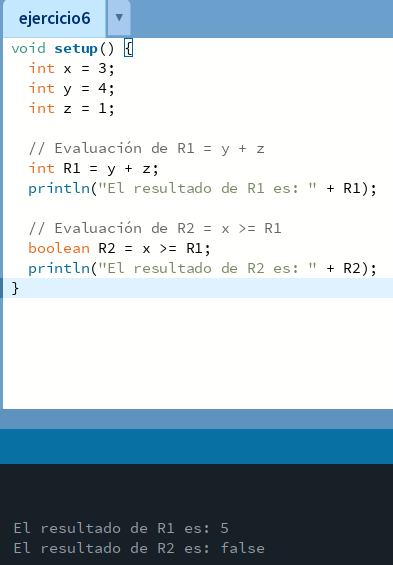
324

****

**Ejercicio 6**: Para x=3, y=4; z=1, evaluar elresultado de R1 = y+z R2 = x >= R1

R1= 4+1 = 5

R2= 3 >= R1 Falso



**Ejercicio 7**: Para contador1=3, contador3=4, evaluar el resultado de

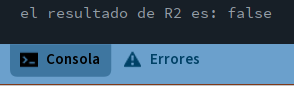
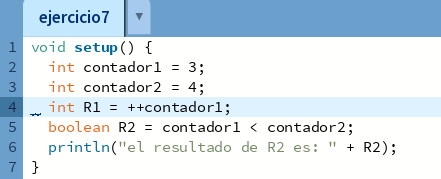
R1 = ++contador1

R2 = contador1 < contador2

Como :

R2= 4 < 4

R2= falso es falso

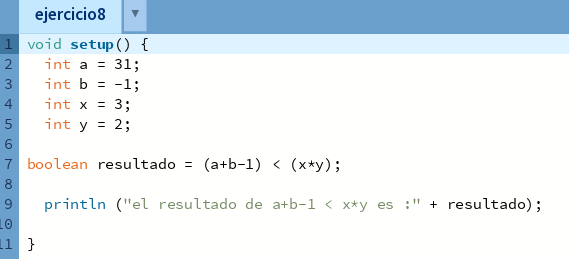
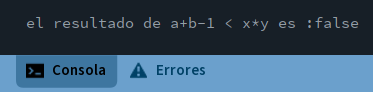
 

**Ejercicio 8:** Para a=31, b=-1; x=3, y=2, evaluar el resultado de a+b-1 < x\*y

a+b-1 < x\*y

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

29 < 12 es falso

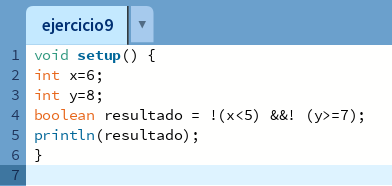
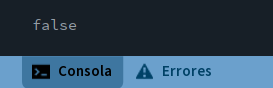


**Ejercicio 9**: Para x=6, y=8, evaluar el resultado de !(x<5)CC !(y>=7)

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

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

falso && falso (entonces es falso)



**Ejercicio 10:** Para i=22,j=3, evaluar el resultado de !((i>4) || !(j<=6))

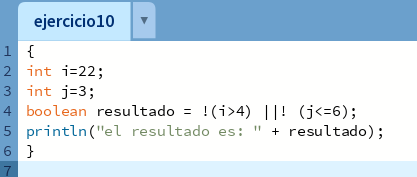
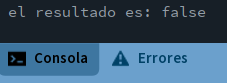
!((i>4) || !(j<=6))

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

!(verdadero || falso)

!(verdadero) “pero entra en negación es falso”

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) && (b-c>=19)

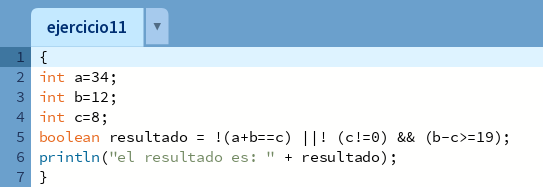
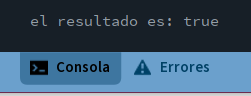
!(34+12==8) || (8!=0)&&(12-8>=19)

!(46==8) || (8!=0)&&(4>=19)

verdadero || verdadero && falso (la compueta && tiene prioridad)

verdadero|| falso

verdadero (es verdadero)



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

\_Para cada ejercicio, en el archivo Word agregar las secciones de análisis y diseño, mientras que, para la codificación, crear el archivo 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.

**Definición de problema:** Evaluar la expresión aritmética

**Análisis:**

•**Datos de Entrada:** nombre- ingresado

•**Datos de Salida:** mensaje- saludo

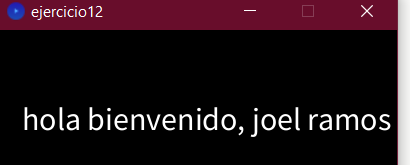
•***PROCESO:***

¿Quien debe realizar el proceso?: computadora

¿Cuál es el proceso que se realiza?: resolvemos lo siguiente: Ingresar un nombre que devolverá la creación de un saludo personalizado con el nombre proporcionado y su presentación en pantalla.

Diseño:

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL POBLEMA** : Algoritmo |
| **VARIABLES:**  nombre\_ingresado: **string** // almacena el nombre  mensaje\_saludo: **string** // almacenara una cadena de caracteres |
| **NOMBRE DEL ALGORITMO:** salude\_nombre  **PROCESO DEL ALGORITMO:**   1. inicio 2. Leer nombre\_ingresado 3. mensaje\_saludo ← “Hola bienvenido, ” + nombre\_ingresado 4. Mostrar saludo 5. fin |



**Ejercicio 13**: Será común resolver problemas utilizando variables. Calcule el perímetro y

área de un rectángulo dada su base y su altura.

**Definición de problema:** Calcular el perímetro y área de un rectángulo dadas sus dimensiones (base y altura).

**Análisis:**

•**Datos de Entrada:** base=15, altura=12

•**Datos de Salida:** perimetro, area

•***PROCESO:***

¿Quien debe realizar el proceso?: El usuario o calculadora

¿Cuál es el proceso que se realiza?: Se obtiene el calculo del perímetro(P) y del área (A) de un rectángulo utilizando las fórmulas :

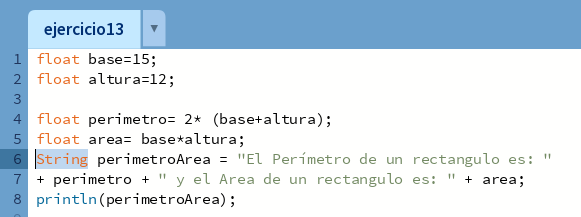
**P = 2 \* (b + h)** y **A = b \* h**

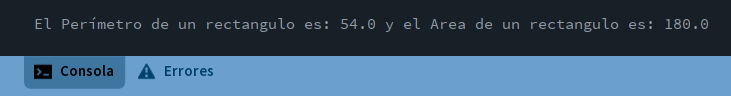
P= 2\*(15+12) A= 15\*12

P= 54 A=180

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL POBLEMA** : persona |
| **VARIABLES:**  base: funcion float  area: funcion float  perimetro: funcion float  area: función float  perimetroArea: función String |
| **NOMBRE DEL ALGORITMO:** CalcularPerimetroyAreaRectangulo  **PROCESO DEL ALGORITMO:**   1. inicio 2. Leer base 3. Leer area 4. perimetro ← 2\*(base + altura) 5. area ← base \* altura 6. perimetroArea ← “el perimetro de un rectángulo es: ” + perimetro + “ y la area de un rectángulo es: ” + area 7. Mostrar perimetroArea 8. fin |

Diseño:





**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 de problema:** Obtener la hipotenusa de un triángulo rectángulo conociendo la longitud de sus dos catetos.

**Análisis:**

•**Datos de Entrada:** catetoa, catetob

•**Datos de Salida:** hipotenusa

•***PROCESO:***

¿Quien debe realizar el proceso?: persona o calculadora

¿Cuál es el proceso que se realiza?: El teorema de Pitágoras establece que en un triángulo rectángulo, en donde la hipotenusa (el lado opuesto al ángulo recto) es igual a la raíz cuadrada de la suma de los cuadrados de los dos catetos (los lados que forman el ángulo recto). La fórmula matemática para calcular la hipotenusa es: : **h^2=a^2+b^2**

h = √(a² + b²) en donde las variables

h = √(3² + 4²)

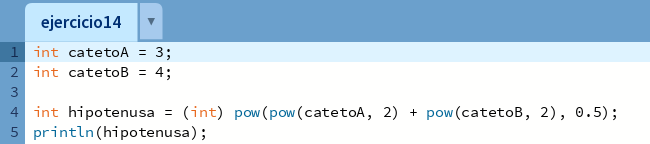
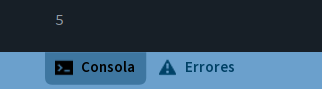
h = √(9 + 16) a= cateto a tiene 3

h= √25 b= cateto b tiene 4

h = 5 h= hipotenusa (resultado)

Diseño:

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL POBLEMA** : programador |
| **VARIABLES:**  catetoA: int  catetoB: int ( almacena un valor decimal)  hipotenusa: int (almacena un valor de cálculos) |
| **NOMBRE DEL ALGORITMO:** CalcularHipotenusaTrianguloRectangulo  **PROCESO DEL ALGORITMO:**   1. Inicio 2. Leer catetoA 3. Leer catetoB 4. hipotenusa ← (a^2 + b^2 ) ^(0.5) 5. mostrar hipotenusa 6. Fin |



**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.

Implementar una calculadora básica que permita realizar las operaciones de suma, resta, multiplicación y división entre dos números.

**Análisis:**

•**Datos de Entrada:**

num1 (valor 110) y num2 (valor 12)

•**Datos de Salida:**

suma: suma = num1 + num2

resta: resta = num1 - num2

Multiplicación: multiplicacion = num1 \* num2

división: division = num1 / num2

•***PROCESO:***

¿Quien debe realizar el proceso?: persona o calculadora

¿Cuál es el proceso que se realiza?:

Le asignanamos valores enteros a las variables num1 y num2

Ejemplos **Num1=27** y **num2=8**

Se realiza la operación suma = num1 + num2 y se almacena el resultado en la variable suma ;luego se imprime el resultado.

**Suma= 35**

Se realiza la operación resta = num1 - num2 y se almacena el resultado en la variable resta; luego se imprime el resultado.

**Resta=19**

Se realiza la operación multiplicacion = num1 \* num2 y se almacena el resultado en la variable multiplicación; y se imprime el resultado

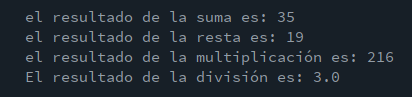
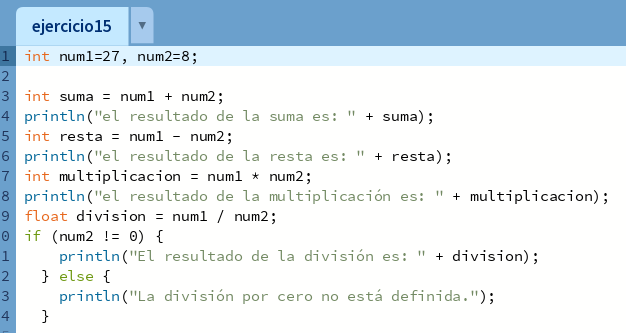
**Multiplicación=216**

Cálculo de la división: Se realiza la operación division = num1 / num2 y se almacena el resultado en la variable division. Se utiliza un bloque condicional if para verificar si num2 es distinto de cero.

**División= 3.375**

Diseño:

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL POBLEMA** : persona |
| **VARIABLES:**  -num1: int (almacena un valor entero)  -num2: int (almacena un valor entero)  -suma: int (almacena un valor de una suma)  -resta: int (almacena un valor de una resta)  -multiplicacion: int (almacena un valor de una multiplicación)  -division: int (almacena un valor de una división) |
| **NOMBRE DEL ALGORITMO:** CalculadoraBásica  **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. multiplicacion ← num1 \* num2 9. mostrar ← “el resultado de la multiplicación es: “ + multiplicacion 10. division ← num1 / num2 11. Si (num2 !=0) entonces 12. mostrar ← “el resultado de la división es: “ + division 13. si\_no 14. mostrar ← “la division por cero no está definida.” 15. Fin |



**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 de problema:** Convertir de grados Fahrenheit en grados Celsius.

**Análisis:**

•**Datos de Entrada:** Temperatura Fahrenheit (un valor a convertir)

•**Datos de Salida:** Temperatura Celsius (resultado de la conversión)

•***PROCESO:***

¿Quien debe realizar el proceso?: persona o calculadora

¿Cuál es el proceso que se realiza?:

Fórmula: valor\_**celsius = (valor\_fahrenheit - 32) \* 5 / 9**

Usuario: Ingresa 68°F.

Programa:

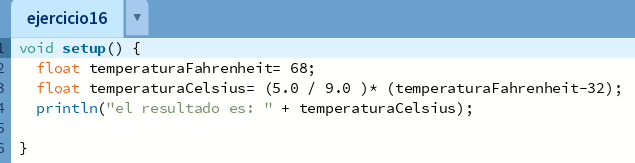
\* valor\_fahrenheit = 68

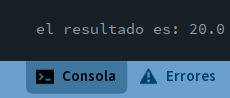
\* valor\_celsius = (68 - 32) \* 5.0 / 9.0 = 20°C

Programa: Muestra "20°C".

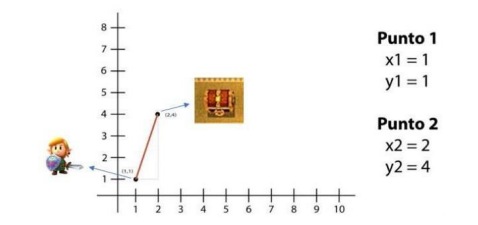
Diseño:

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL POBLEMA** : persona |
| **VARIABLES:**  temperaturaFahrenheit : float (almacena valor decimal)  temperaturaCelsius: float |
| **NOMBRE DEL ALGORITMO:** Convertidor\_temperaturas  **PROCESO DEL ALGORITMO:**   1. Inicio 2. Leer temperaturaFahrenheit 3. temperaturaCelsius ← (5.0/ 9.0) \* (temperaturaFahrenheit – 32) 4. mostrar temperaturaCelsius 5. Fin |





**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.

**Definición de problema:** determino la distancia entre Link y el tesoro para que, al llegar a una distancia específica (x unidades), el tesoro desaparezca.

**Análisis:**

•**Datos de Entrada:**

**Link**: coordenadas (x1, y1)

**tesoro**: coordenadas (x2, y2)

Distancia de desaparición (x): La distancia a la que el tesoro desaparece

•**Datos de Salida:**

Distancia entre Link y el tesoro:

•***PROCESO:***

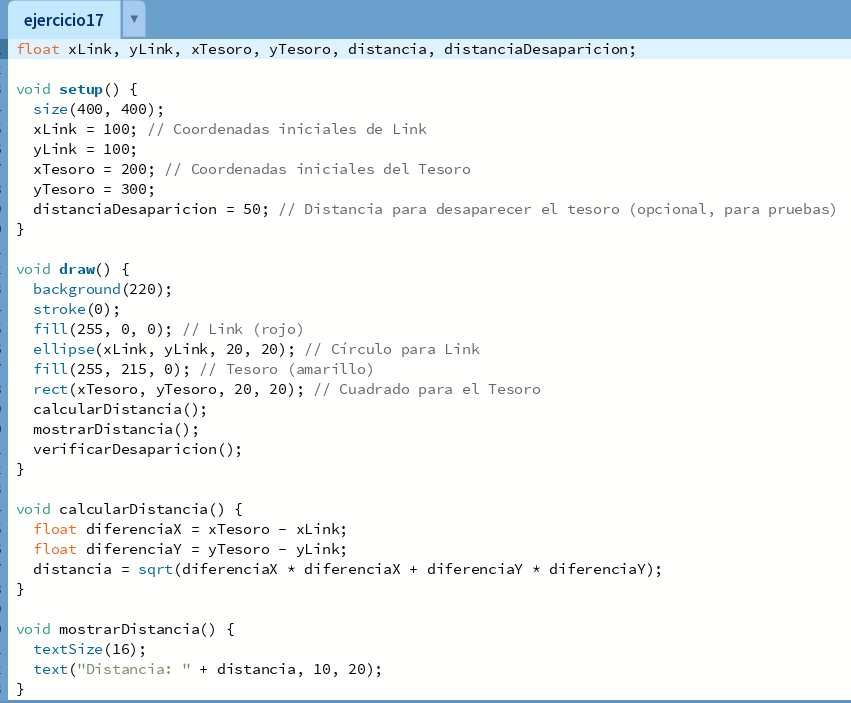
¿Quien debe realizar el proceso?: un programa o calculadora que pueda realizar cálculos matemáticos.

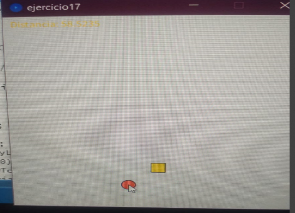
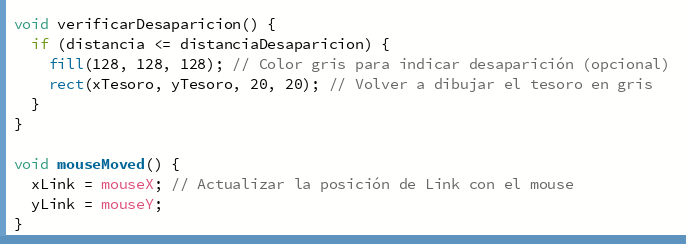
¿Cuál es el proceso que se realiza?:

Se dibujará en el lienzo un circulo que representa a link y un cuadrado representando al tesoro. Se moverá a link con el mouse, por medio de la consola se verá mostrando la distancia que separa a link del tesoro.

Diseño:

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL POBLEMA** : persona |
| **VARIABLES:**  - x1: float  - y1: float (almacena valor decimal)  - x2: float  - y2: float  - coordenadaX: float (almacena el resultado de un calculo  - coordenadaY: float  - distancia: float  - distanciaTesoro: float ( almacena un valor) |
| **NOMBRE DEL ALGORITMO:**  **PROCESO DEL ALGORITMO:**   1. inicio 2. Leer x1 3. Leer y1 4. Leer x2 5. Leer y2 6. distanciaTesoro ← 50 7. coordenadaX ← x2 - x1 8. coordenadaY ← y2 – y1 9. distancia ← ((coordenadaX)^2 + (coordenadaY)^2)^2 10. mostrar “la distancia es de: ” + distancia 11. si (distancia = distanciaTesoro) entonces 12. mostrar “¡PowerUp activado!” 13. fin\_si 14. fin |





**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 de problema:** Obtener las raíces de una ecuación de segundo grado.

**Análisis:**

•**Datos de Entrada:** Coeficientes de la ecuación cuadrática: a, b y c.

•**Datos de Salida:** Raíces de la ecuación cuadrática.

•***PROCESO:***

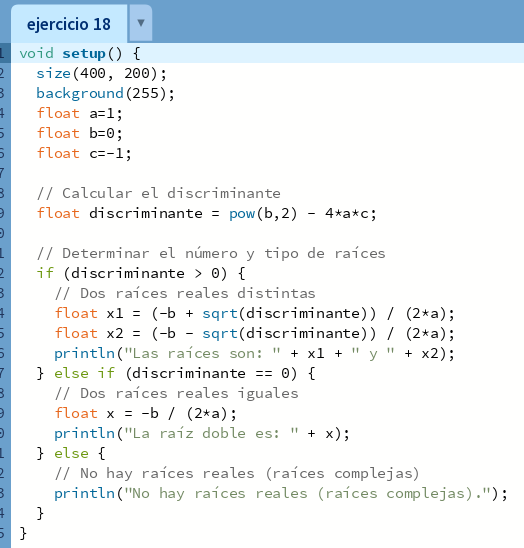
¿Quien debe realizar el proceso?: un programa o calculadora que pueda realizar cálculos matemáticos.

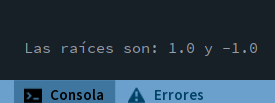
¿Cuál es el proceso que se realiza?:

Calcular el discriminante de la ecuación cuadrática utilizando la fórmula

Diseño:

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL POBLEMA** : persona |
| **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 calculos |
| **NOMBRE DEL ALGORITMO:** Encontrar\_raices  **PROCESO DEL ALGORITMO:**   1. inicio 2. Leer a 3. Leer b 4. Leer c 5. discriminante ← b^2 – 4\*a\*c 6. **si** (discriminante > 0) entonces 7. raiz1 ← (-b + (discriminante))^0.5 /(2\*a) 8. raiz2 ← (-b - (discriminante))^0.5 /(2\*a) 9. mostrar “las raíces son: ” + raiz1 + “ y ” + raiz2 10. **si\_no si** (discriminante == 0) entonces 11. raiz ← -b / (2\*a) 12. mostrar “la raíz doble es: “ + raiz 13. **si\_no** 14. mostrar “no hay raíces reales” 15. 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

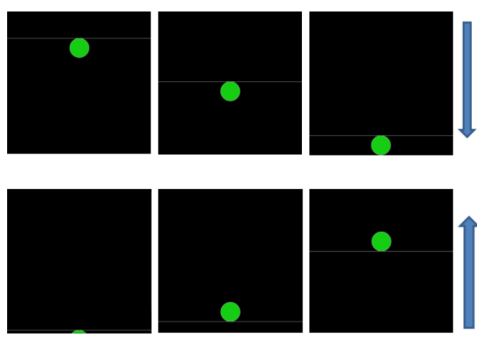
la elipse. Mantenga en cero el valor para background(). Cuando la línea supere la posición

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



**Definición de problema:** Dibujar una línea que se extiende por todo el ancho del lienzo y una elipse que se mueve a lo largo de la línea, cambiando de dirección cuando llega al borde superior o inferior del lienzo.

**Análisis:**

•**Datos de Entrada:**

línea, dire

•**Datos de Salida:**

Bucleo de la línea y circulo

•***PROCESO:***

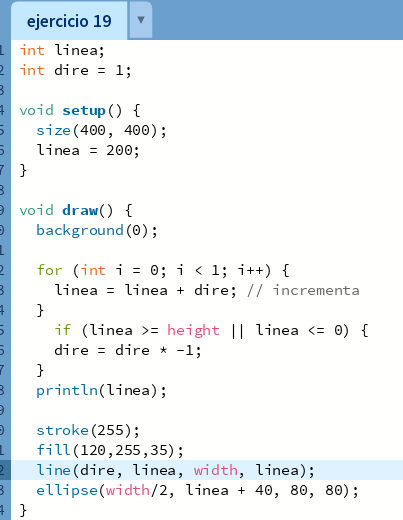
¿Quien debe realizar el proceso?: computadora

¿Cuál es el proceso que se realiza?:

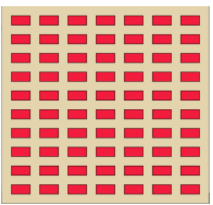
Se dibujara una línea que ocupara toda la extencion del lienzo, en su punto medio en donde se dibujara un circulo de color verde,el cual seguirá la trayectoria que se hizo.

Diseño:

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL POBLEMA** : lienzo |
| **VARIABLES:**  - linea: entero (almacena un valor entero)  - dire : entero |
| **NOMBRE DEL ALGORITMO:** dibujarLineaCirculo  **PROCESO DEL ALGORITMO:**   1. inicio 2. Leer linea 3. Leer dire 4. anchoLienzo ← 400 5. altoLienzo ← 400 6. **para** i ← 0 hasta **alto** incremento 1 **hacer** 7. linea ← linea + dire 8. ***fin\_para*** 9. si ((linea >= anchoLienzo) O (linea <= 0)) ***entonces*** 10. dire ← dire \* (-1) 11. ***fin\_si*** 12. mostrar linea 13. dibujar linea en (dire, linea, altoLienzo, linea) 14. dibujar circulo en (altoLienzo/2, linea + 40, 80, 80) 15. fin |



**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 de problema:** Dibujar una cuadrilla de rectángulos en todo el lienzo.

**Análisis:**

•**Datos de Entrada:**

Lienzo: 440x420

ancho Rectángulo: 40

altoRectangulo: 20

distancias v y h: 20

•**Datos de Salida:**

Múltiples rectángulos de color rojo dibujados en el lienzo, manteniendo una distancia especifica.

•***PROCESO:***

¿Quien debe realizar el proceso?: computadora

¿Cuál es el proceso que se realiza?:

Establezco el tamaño del lienzo a 440x440 pixeles y establezco el color de fondo a gris claro.

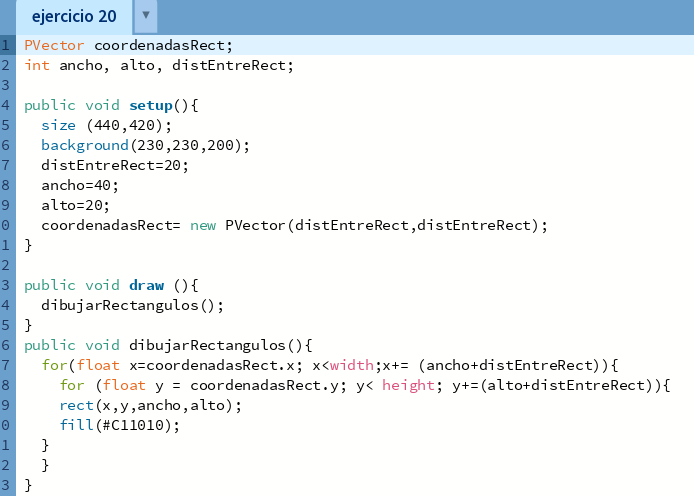
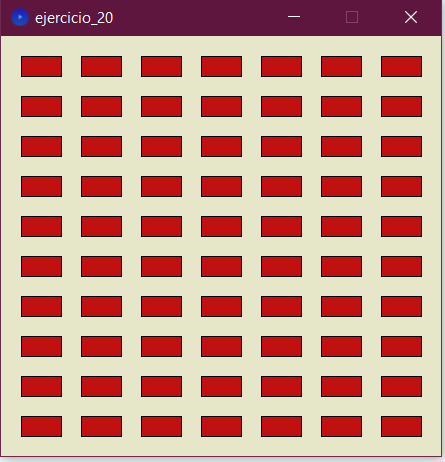
Defino las variables para almacenar el ancho y alto de los rectángulos, asi como la distancia horizontal y vertical entre ellos.

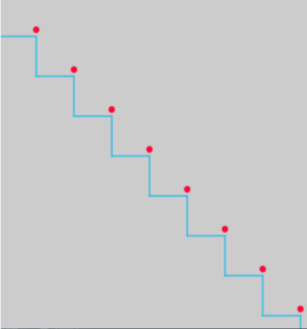
Calculo el número de filas y columnas de rectángulos que caben en el lienzo, teniendo en cuenta el ancho de los rectángulos y la distancia horizontal entre ellos.

Se utilizan dos bucles For para recorrer las filas y columnas del lienzo, estableciendo el color de relleno del rectángulo a rojo por el cual se procede a dibujar cada rectángulo en la posición calculada con las dimensiones especificadas.

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL POBLEMA** : El Lienzo |
| **VARIABLES:**  coordenadasRect, coordenadas cartesianas  ancho, alto, distanciaEntreRect:entero  anchoLienxo, altoLienzo: entero |
| **NOMBRE DEL ALGORITMO:** rectángulos repetidos  **PROCESO DEL ALGORITMO:**   1. Inicio 2. ancholienzo ← 440 3. altolienzo ← 420 4. distanciaEntreRect ←20 5. ancho ← 40 6. alto ← 20 7. **para** **x**= ← coordenadasRect.x **hasta**¨anchoLienzo¨¨**con¨¨paso**(ancho+distEnteRect). 8. hacer 9. **Para y**= ← coordenadasRect.y **hasta**¨altoLienzo¨¨**con¨¨paso**(alto+distEnteRect). 10. hacer 11. dibujar rectángulo en (x,y ancho,alto) 12. fin para 13. fin |

Diseño:



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.

**Definición de problema:** Dibujar una cuadrilla de rectángulos en todo el lienzo.

**Análisis:**

•**Datos de Entrada:** puntoA, puntoB, puntoC, puntoD, distancia

•**Datos de Salida:** una imagen que consiste en escalones con puntos de color rojo en los bordes.

Múltiples rectángulos de color rojo dibujados en el lienzo, manteniendo una distancia especifica.

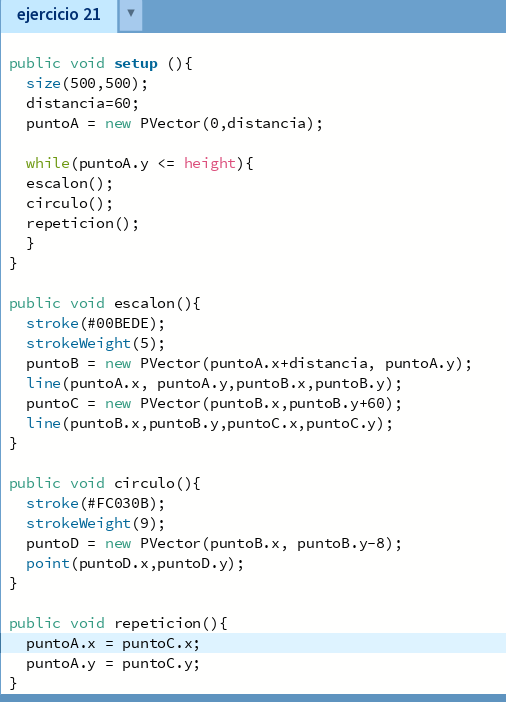
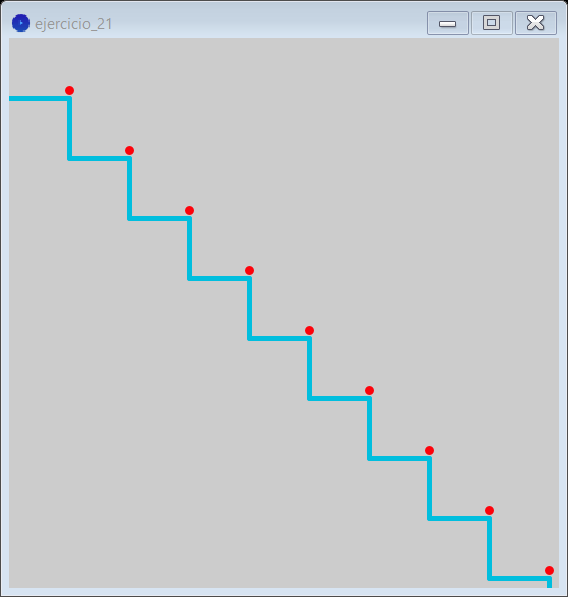
•***PROCESO:***

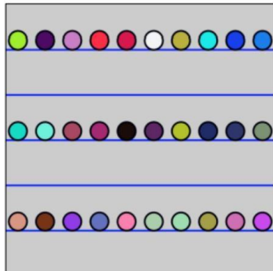
¿Quien debe realizar el proceso?: un programa

Una vez definido tanto el alto como el ancho del escalón, con uso del while/mientras, se procederá a dibujar en el lienzo los escalones, que irán intercalándose a medida que vallan sumando sus valores. Luego de los escalones están dibujados, se procederá a dibujar los pequeños círculos en cada escalón ,para ello, le restamos 10 unidades al valor que se contenga y en ese momento.

Diseño:

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL POBLEMA** : El Lienzo |
| **VARIABLES:**  - puntoA,puntoB,puntoC,puntoD: **int**  (almacena un vector)  - distancia : **int** (almacena un valor entero) |
| **NOMBRE DEL ALGORITMO:** rectángulos repetidos  **PROCESO DEL ALGORITMO:** escalas\_puntos   1. Inicio 2. anchoLienzo ← 500 3. altoLienzo ← 500 4. distancia ← 60 5. **mientras** (puntoA.y sea menor o igual que anchoLienzo) 6. Hacer 7. **dibujar** línea horizontal en (puntoA.x, puntoA.y, puntoB.x, puntoB.y) 8. **dibujar** línea vertical en (puntoB.x, puntoB.y, puntoC.x, puntoC.y) 9. **dibujar** circulo en (puntoD.x, puntoD.y) 10. puntoA.x ← puntoC.x 11. puntoA.y ← puntoC.y 12. **fin\_mientras** 13. Fin |



**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 de problema:** Dibujar líneas en el lienzo, dibujando a la vez círculos de varios colores por línea de por medio.

**Análisis:**

•**Datos de Entrada:** líneas, y (enteros)

•**Datos de Salida:** círculos de varios colores sobre líneas con un color azul con distanciamiento por medio en pantalla.

•***PROCESO:***

¿Quien debe realizar el proceso?: un programa

¿Cuál es el proceso que se realiza?:

Se dibujaran líneas con una separación entre ellas en que deben dibujar los círculos manteniendo una distancia entre ellos. En el cual se dibujaran los círculos en la primera línea, y en la segunda línea no se dibujara, y se volverá a dibujar en la tercera, sucesivamente.

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL POBLEMA** : Programador |
| **VARIABLES:**  - línea: enteros  - cordenadas y: enteros |
| **NOMBRE DEL ALGORITMO:** dibujar\_lineas\_circulos  **PROCESO DEL ALGORITMO:**   1. **Inicio** 2. anchoLienzo ← 600 3. altoLienzo ← 600 4. linea ← 120 5. **hacer** 6. line(0,línea,ancho,línea) 7. línea+=100 8. **mientras** (línea<=altoLienzo) 9. **hacer** 10. **Para** **x** ←30 **hasta** anchoLienzo incremento 60 **hacer** 11. Elipse(x,y,40,40) 12. Y+=200 13. **Mientras** ( y<=altoLienzo) 14. Fin |



