**Desarrollo:**

**Ejercicio1: evaluar la siguiente expresión para a=2 y b=5;**

**3\*a-4\*b / a^2**

**Resolución**

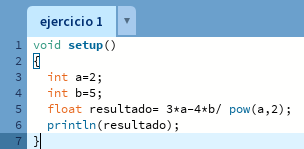
**(3\*a) - (4\*b / (a^2))**

**6-(4\*b/4)**

**6-5**

**1**

**Captura**



**Ejercicio2: evaluar la siguiente expresión**

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

**Resolución**

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

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

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

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

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

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

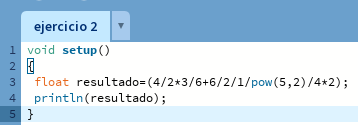
**1+(0.12/4)\*2**

**1+(0.03\*2)**

**(1+0.06)**

**1.06**

**Captura**



**Ejercicio4: evaluar las siguientes expresiones aritméticas, para lo cual indicar el valor de las variables. Luego escribirlas como expresiones algebraicas: para a=2 b=3 c=4 d=5 x=6 y=7**

**a) b^2-4\*a\*c algebraica:**

**(3^2)-(4\*2\*4)**

**9-32**

**-23**

**b) 3\*x^4-5\*x^3+x\*12-17 algebraica:**

**3\*(6^4)-5\*(6^3)+6\*12-17**

**3\*1296-5\*216+6\*12-17**

**(3\*1296)-5\*216+6\*12-17**

**3888-(5\*216)+6\*12-17**

**3888-1080+(6\*12)-17**

**(3888-1080)+(72-17)**

**2808+55**

**2863**

**c) (b+d)/(c+4) algebraica:**

**(3+5)/(4+4)**

**8/8**

**1**

**d) (x^2+y^2)^(1/2) algebraica:**

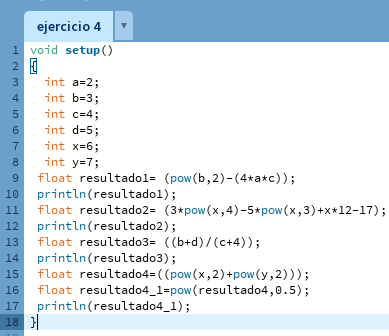
**(6^2+7^2)^(1/2)**

**(36+49)^0.5**

**85^0.5**

**9.21**

**Captura**



**Ejercicio5: si el valor de a=4, b=5 y c=1 evaluar las siguientes expresiones**

**a) b\*a-b^2/4\*c**

**5\*4-5^2/4\*1**

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

**(5\*4)-25/4\*1**

**20-(25/4)\*1**

**20-(6.25\*1)**

**20-6.25**

**13.75**

**b) (a\*b)/3^2**

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

**20/(3^2)**

**20/9**

**2.222…**

**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**

**((12+10)\*3\*5)-6**

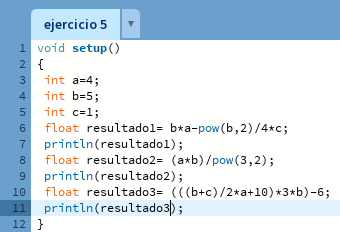
**((22\*3)\*5)-6**

**(66\*5)-6**

**330-6**

**324**

**Captura**



**Ejercicio6: para x=3, y=4, z=1**

**R1=y+z**

**=4+1**

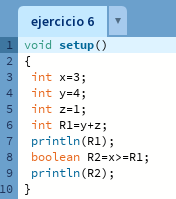
**=5**

**R2=x>=R1**

**=3>=5**

**=false**

**Captura**



**Ejercicio7: para contador1=3 , contador2=4**

**R1=++contador1**

**=1+3**

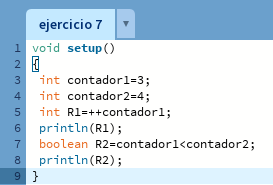
**=4**

**R2=contador1<contador2**

**=4<4**

**=false**

**Captura**



**Ejercicio8: para a=31, b=1, x=3, y=2**

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

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

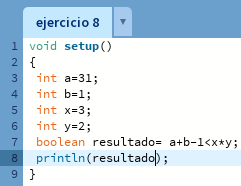
**31+1-1<6**

**31+(1-1)<6**

**31+0<6**

**False**

**Captura**



**Ejercicio9: para x=6, y=8**

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

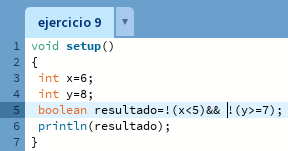
**!(6<5)&&!(8>=7)**

**(6>5)&&(8<7)**

**(true) && (false)**

**False**

**Captura**



**Ejercicio10: para i=22, j=3**

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

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

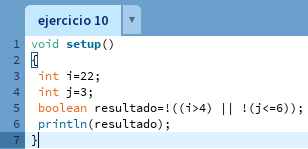
**!((22>4) || (3>6))**

**!((true) || (false))**

**!(true)**

**False**

**Captura**



**Ejercicio11: para a=34, b=12 c=8**

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

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

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

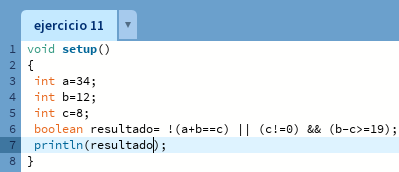
**(46!=8) || (8!=0) && (4>=19)**

**(true) || ((true) && (false))**

**(true) || (false)**

**true**

**Captura**



**Ejercicio12:**

**Análisis**

**Especificación del problema: se debe ingresar un nombre para posteriormente usarlo junto a un saludo**

**Datos de entrada**

**Nombre: String**

**Datos de salida**

**Saludo, nombre: String**

**Proceso**

**Pedido del nombre // se busca que el usuario ingrese su nombre**

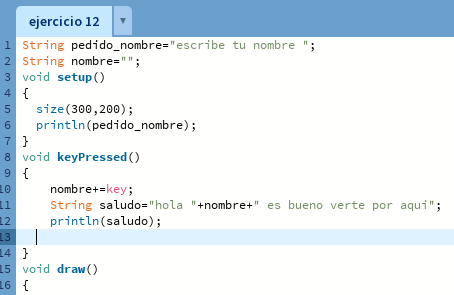
**Nombre ingresado // el nombre suministrado por el usuario**

**Saludo // se presenta un saludo acompañado del nombre**

**Diseño**

|  |
| --- |
| **Entidad que resuelve el problema: saludo\_al\_usuario** |
| **Variables**  **pedido\_nombre, nombre, saludo: String** |
| **Nombre del algoritmo: saluda**  **Proceso del algoritmo**  **1. pedido\_nombre ← “escribe tu nombre”**  **2. nombre ← string**  **3. saludo ←”hola ”+nombre+”es bueno verte por aqui” // se define la variable agregándole el nombre**  **4. nombre+=key //se pide que guarde las ultimas teclas que presiono el usuario**  **5. mostrar saludo** |

**Captura**



**Ejercicio13:**

**Análisis**

**Especificación del problema: se busca calcular el área y perímetro de un rectángulo dada su base y altura**

**Datos de entrada:**

**Base, altura: float**

**Datos de salida:**

**Área, perímetro: float**

**Proceso**

**Base y altura // se le asigna un valor a las variables**

**Área= base\*altura // se hace uso de la formula correspondiente**

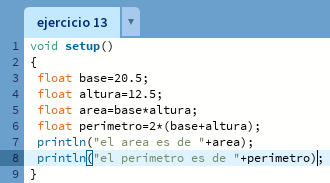
**Perímetro=2\*(base+altura) // se hace uso de la formula correspondiente**

**// se muestra el resultado para ambas operaciones**

**Diseño**

|  |
| --- |
| **Entidad que resuelve el problema:formula\_rectangulo** |
| **Variables:**  **Base, altura : float** |
| **Nombre del algoritmo: calculo\_area\_perimetro**  **Proceso del algoritmo:**  **1. leer base**  **2. leer altura**  **3. area ← base\*altura**  **4. perímetro ←2\*(base+altura)**  **5. mostrar área**  **6. mostrar perímetro** |

**Captura**



**Ejercicio14:**

**Análisis**

**Especificación del problema: se necesita obtener la hipotenusa de un triangulo rectángulo dados sus catetos**

**Datos de entrada:**

**Cateto1, cateto2:float**

**Datos de salida:**

**Hipotenusa:float**

**Proceso:**

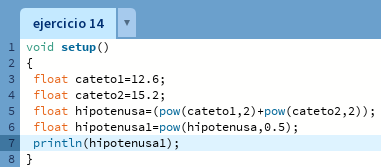
**Cateto1, cateto2 // se asigna una valor para cada variable**

**Hipotenusa=(cateto1^2+cateto2^2)^(1/2) //se utiliza pitagoras para calcular la hipotenusa, además de utilizarse la reciproca de la raíz ,es decir, a^(1/2)**

**Diseño**

|  |
| --- |
| **Entidad que resuelve el problema: pitagoras** |
| **Variables:**  **cateto1,cateto2: float** |
| **Nombre del algoritmo: calcular\_hipotenusa**  **Proceso del algoritmo:**  **1.leer cateto1**  **2.leer cateto2**  **3.hipotenusa ← (cateto1^2+cateto2^2)^(1/2)**  **4.mostrar hipotenusa** |

**Captura**



**Ejercicio15:**

**Ejercicio16:**

**Análisis**

**Especificación del problema: se necesita convertir grados Fahrenheit a grados Celsius**

**Datos de entrada:**

**Grados Fahrenheit: float**

**Datos de salida:**

**Grados Celsius**

**Proceso:**

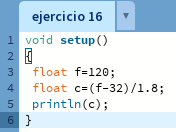
**Fahrenheit // se ingresan los gados en esta unidad de temperatura**

**Celsius=(Fahrenheit-32)/1.8 // se hace uso de la fórmula de conversión**

**Diseño**

|  |
| --- |
| **Entidad que resuelve el problema:formula\_convercion** |
| **Variables**  **F, c:float** |
| **Nombre del algoritmo: conversión\_celcius**  **Proceso del algoritmo:**  **1. leer f**  **2.c ← (f-32)/1.8**  **3 mostrar c** |

**Captura**



**Ejercicio18:**

**Análisis**

**Especificación del problema: se desea encontrar las raices de una funcion cuadrática**

**Datos de entrada:**

**Pendiente, valor lineal, termino independiente: float**

**Datos de salida:**

**Raiz1, raiz2: float**

**Proceso:**

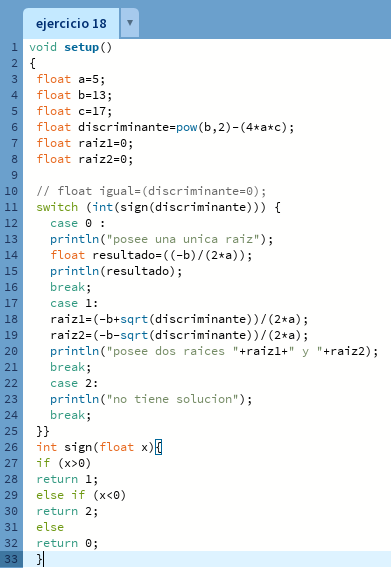
**Pendiente, valor lineal, termino independiente // se asignan los valores correspondientes para cada variable**

**// se hace uso de la formula cuadrática para encontrar las raíces x1 y x2**

**Diseño**

|  |
| --- |
| **Entidad que resuelve el problema: formula\_cuadratica** |
| **Variables:**  **a,b,c:float** |
| **Nombre del algoritmo: encontrar\_raices**  **Proceso del algoritmo:**  **1. leer a**  **2. leer b**  **3. leer c**  **4. según\_sea ((-b^2)-(4\*a\*c)) hacer**  **5.caso1 ((-b^2)-4\*a\*c)=0**  **6.mostrar “tiene una única raiz”**  **7.resultado ← ((-b)/(2\*a)**  **8.Mostrar resultado**  **9.breack**  **10.caso2 ((-b^2)-(4\*a\*c))>0 hacer**  **11.mostrar “posee dos raices”**  **12.Resultado1← ((-b)+ ((-b^2)-(4\*a\*c))^0.5)/(2\*a)**  **13.Resultado2 ← ((-b)- ((-b^2)-(4\*a\*c))^0.5)/(2\*a)**  **14.Mostrar resultado1**  **15.Mostrar resultado2**  **16.breack**  **17.caso3 ((-b^2)-4\*a\*c)<0 hacer**  **18. mostrar “no tiene solución”**  **19.breack** |

**Captura**



**Ejercicio20:**

**Análisis:**

**Especificación de problema:**

**Se necesita dibujar rectángulos de idénticas medidas (40 de ancho y 20 de alto) dentro de todo el lienzo (440 de ancho y 420 de alto). Utilizando la estructura de control For (para)**

**Datos de entrada:**

**Ancho lienzo, Alto lienzo: entero // dimensiones de la zona de trabajo en pixeles**

**Ancho, Alto: entero // dimensiones de los rectángulos en pixeles**

**Distancia: entero // pixeles de distancia entre rectángulos**

**Datos de salida:**

**Consecución de rectángulos con cierta medida de distancia**

**Proceso:**

**Lienzo= (440,420) // se asigna el valor para el eje “X” y el eje “Y” (ancho y alto)**

**Ancho=40 // se determina el ancho de los rectángulos**

**Alto=20 // se determina el alto de los rectángulos**

**Distancia=20 // se determina la distancia entre los rectángulos**

**Diseño:**

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL PROBLEMA: Lienzo** |
| **VARIABLES:**  **size(): entero // representa el tamaño del lienzo**  **ancho, alto, dist: entero // representan las medidas del y entre los rectángulos** |
| **NOMBRE DEL ALGORITMO: dibujar\_rectangulos** |
| **PROCESO DEL ALGOIRITMO:**   1. **inicio** 2. **size ⟵ (440,420)** 3. **ancho ⟵ 40** 4. **alto ⟵ 20** 5. **dist ⟵ 20** 6. **Para posx=20 hasta posx<440 con paso en posx+= (ancho+dist)** 7. **hacer** 8. **Para posy=20 hasta posy<420 con paso en posy+= (alto+dist)** 9. **hacer** 10. **dibujar rectángulo en (posx, posy, ancho, alto)** 11. **fin\_para** 12. **fin\_para** 13. **fin** |

**Captura**



**Ejercicio21**

**Análisis:**

**Especificaciones del problema: se necesita dibujar escalones con un punto en cada vértice en un lienzo de 500 pixeles de ancho y alto ,utilizando la estructura de control While()**

**Datos de entrada:**

**Coordenada de las líneas, coordenadas del punto: entero // se definen las coordenadas para las líneas y puntos en pixeles**

**Datos de salida:**

**// escalones dibujados con un punto en las esquinas**

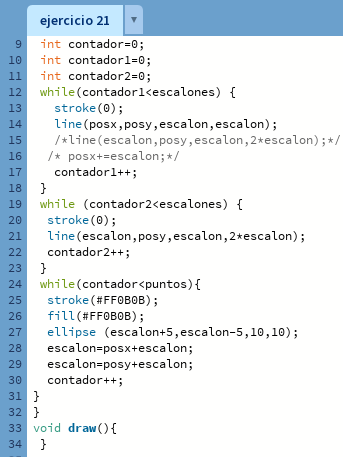
**Proceso:**

**// se dibujan los escalones y puntos determinados por las coordenadas**

**Diseño:**

|  |
| --- |
| **Entidad que resuelve el problema: lienzo** |
| **Variables**  **posx, posy,escalones,escalon,puntos :entero** |
| **Nombre del algoritmo: dibujar\_escalones**  **Proceso del algortmo**  **1.escalones ← 8**  **2.puntos ← 8**  **3.leer posx**  **4.leer posy**  **5.leer escalon**  **6.contador ← 0**  **7.contador1 ← 0**  **8.while(contador1<escalones)**  **9.line(posx,posy,escalon,escalon)**  **10.contador1++**  **11.while()** |

**Captura**

****

**Ejercicio22:**

**Análisis**

**Especificación del problema: se deben dibujar cinco líneas horizontales y en líneas de por medio se deben dibujar círculos con coloreas aleatorios**

**Datos de entrada:**

**Distancia\_lineas, separacion\_circulos: entero**

**Datos salida:**

**// las líneas y círculos dibujados**

**Diseño**

|  |
| --- |
| **Entidad que resuelve el problema: lienzo** |
| **Variables**  **dist, sep, distSep, distSep1 : entero** |
| **Nombre del algoritmo:**  **Proceso del algoritmo:**  **1.leer dist**  **2.leer sep**  **3.leer distSep**  **4.leer distSep1**  **5. do {**  **6. line(0,dist,width,dist)}**  **7.elipse(sep,75,50,50)**  **8.ellipse(sep,distSep,50,50)**  **9.ellipse(sep,distSep,50,50)**  **10.relleno random**  **11.dist+=100**  **12.sep+=120**  **13.distSep=distSep0**  **14.distSep1=distSep1+0**  **15. while(dist<600 && sep<600)**  **16.fin** |

**Captura**

