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

*Ing. Carolina Cecilia Apaza*

*Año*

Trabajo Práctico/Actividad

N°

Apellido y Nombre – LU /

Grupo:

Integrantes

AyN /LU

**FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS**

Indice

Punto 1: Enunciado del punto

Desarrollo del punto

: 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))

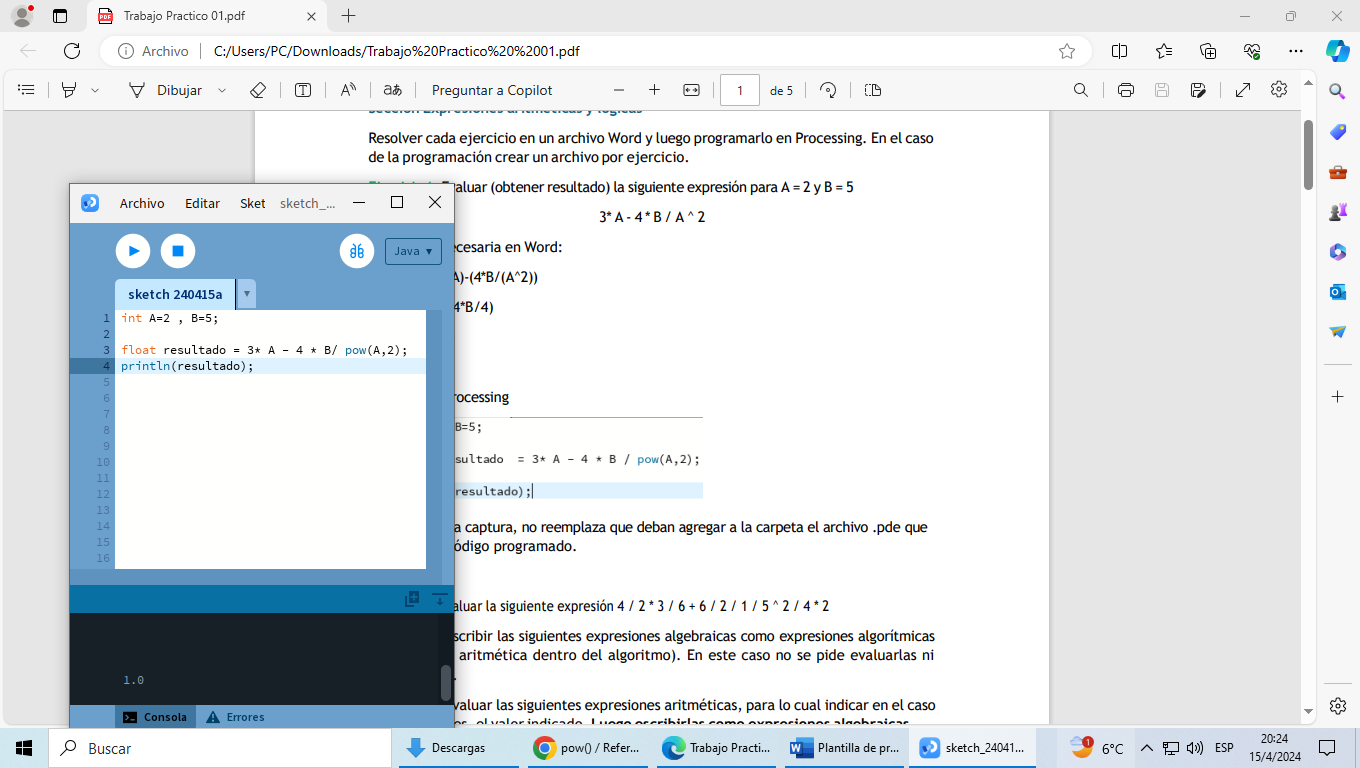
(3\*2)-(4\*B/(2^¨2))

6 – (4 \*5/ 4)

6 – 5

1

Captura de Processing Ojo: Colocar la captura, no reemplaza que deban agregar a la carpeta el archivo .pde que contiene el código programado.



Ejercicio 2: Evaluar la siguiente expresión

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

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

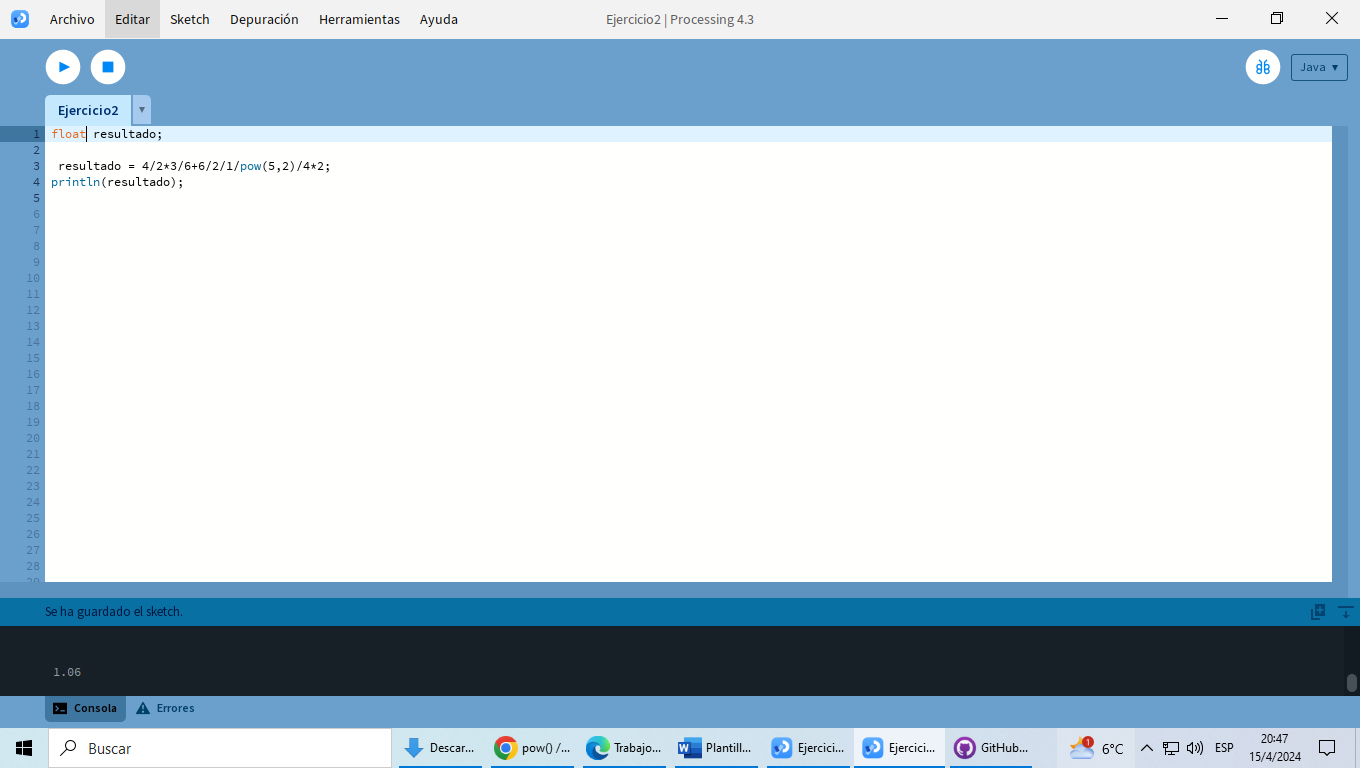
2 \* 3 / 6 + 6 / 2 / 1 / 25 / 4 \* 2

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

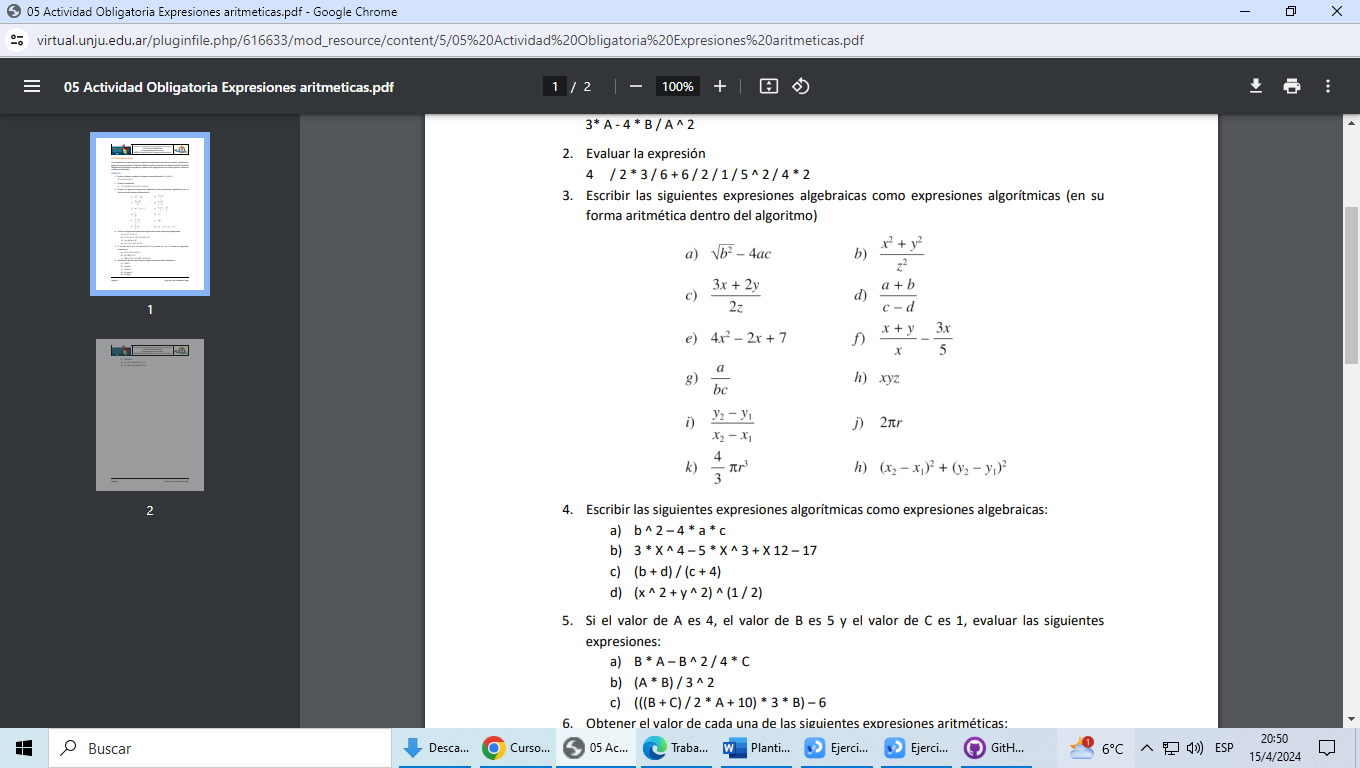
1 + 3 / 1 / 25 /4 \* 2

1 + 3 / 25 / 4 \* 2

1.06



Ejercicio 3:

Escribir las siguientes expresiones algebraicas como expresiones algorítmicas (en su forma aritmética dentro del algoritmo). En este caso no se pide evaluarlas ni programarlas.

1. ( b ^ 2) – 4 \* 2 \*C
2. (x ^ 2 + y ^ 2)/ z ^ 2
3. (3 \* x + 2 \* y)/ 2 \* z
4. ( a + b ) / (c – d )
5. 4 \*x ^ 2 + 2 \* x +7
6. ( x + y )/ x - 3 \* x / 5
7. a / b \* c
8. x \* y \* z
9. (y2 – y1) / (x2 -x1)
10. 2 \* pi \* r
11. 4 / 3 \* pi \* r ^ 3
12. (x2 – x1) ^ 2 - (y2 – y1) ^ 2

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)

desarrollo:

a) b ^ 2 – 4 \* a \* c

b2 – 4.a.c

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

3.x4 – 5.x3+x\*12 -17

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

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

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

5 \* 4 – 5 ^ 2 / 4 \* 1

5 \* 4 – 25 / 4

20 – 25 / 4

20 – 6.25

13.75

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

( 4 \* 5) / 9

20/9

2.2222

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

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

((5 / 2 \* 5 + 10 ) \* 3 \* 4 ) – 6

((2.5 \* 5 + 10 ) \* 3 \* 4 ) – 6

((12.5 +10) \* 3 \* 4 ) -6

( 22.5 \* 3 \* 4) -6

(67.5 \* 4) – 6

270 – 6

254

Ejercicio 6:

Para x=3, y=4; z=1, evaluar el resultado de

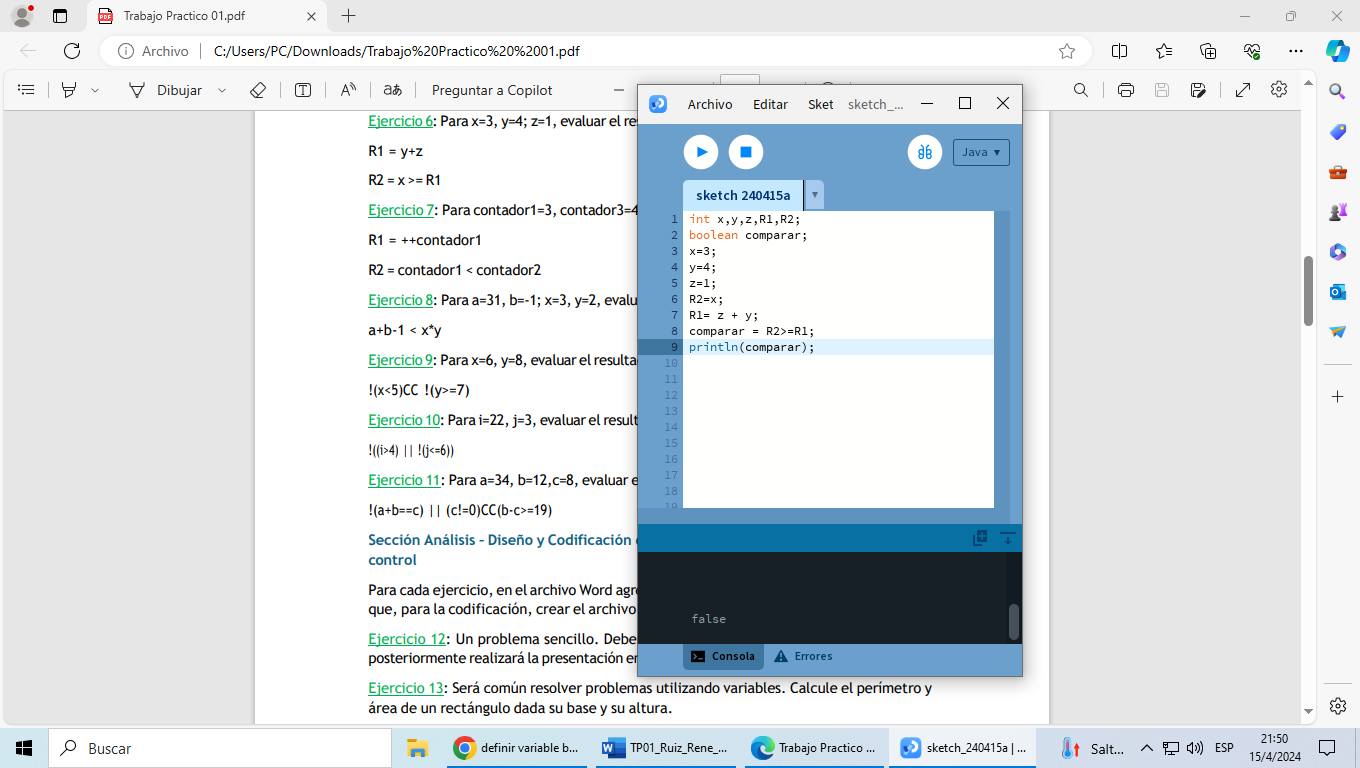
R1 = y+z

R2 = x >= R1

R1 = 4 + 1

R1= 5

R2 = x >= R1

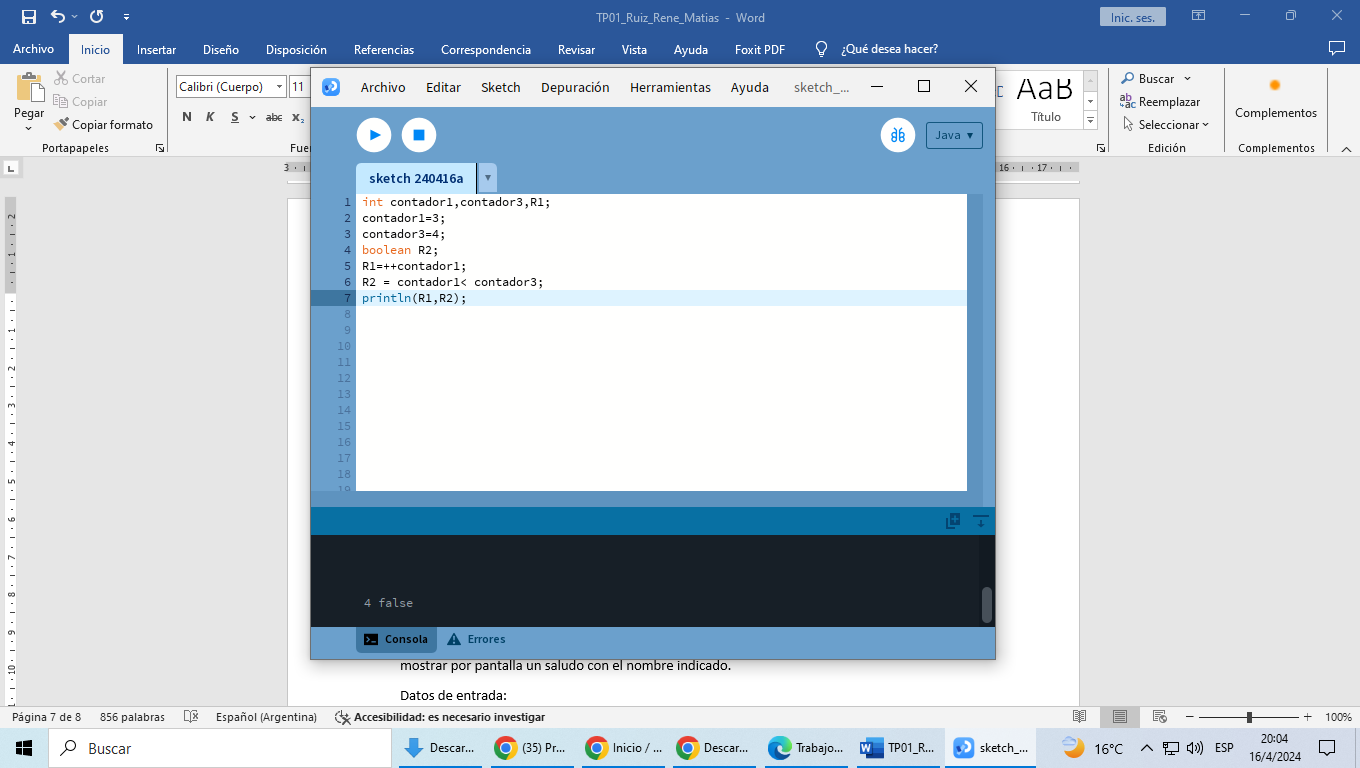


Ejercicio 7:

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

R1 = ++contador1

R2 = contador1 < contador2



R1= ++3

R1=4

R2 = 4 < 4

R2= False

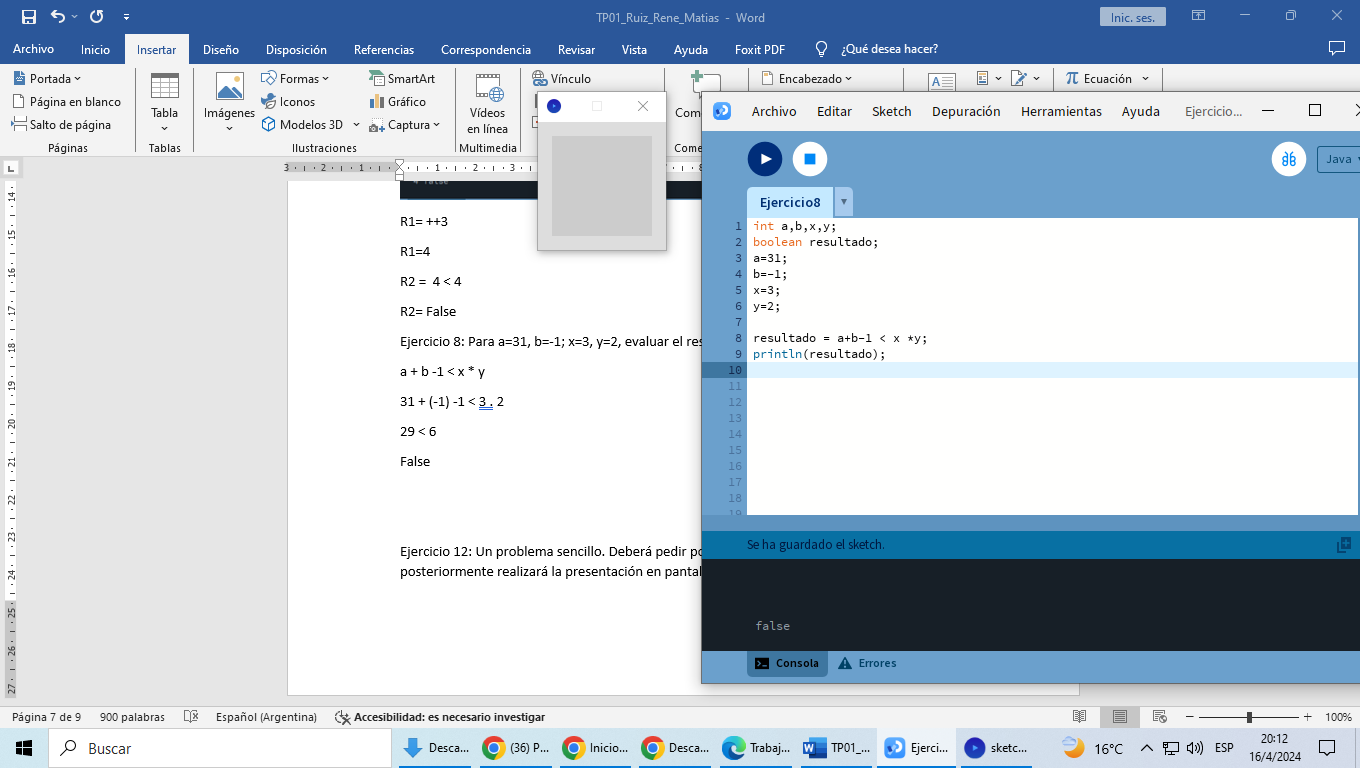
Ejercicio 8: Para a=31, b=-1; x=3, y=2, evaluar el resultado de a+b-1 < x\*y

a + b -1 < x \* y

31 + (-1) -1 < 3 . 2

29 < 6

False

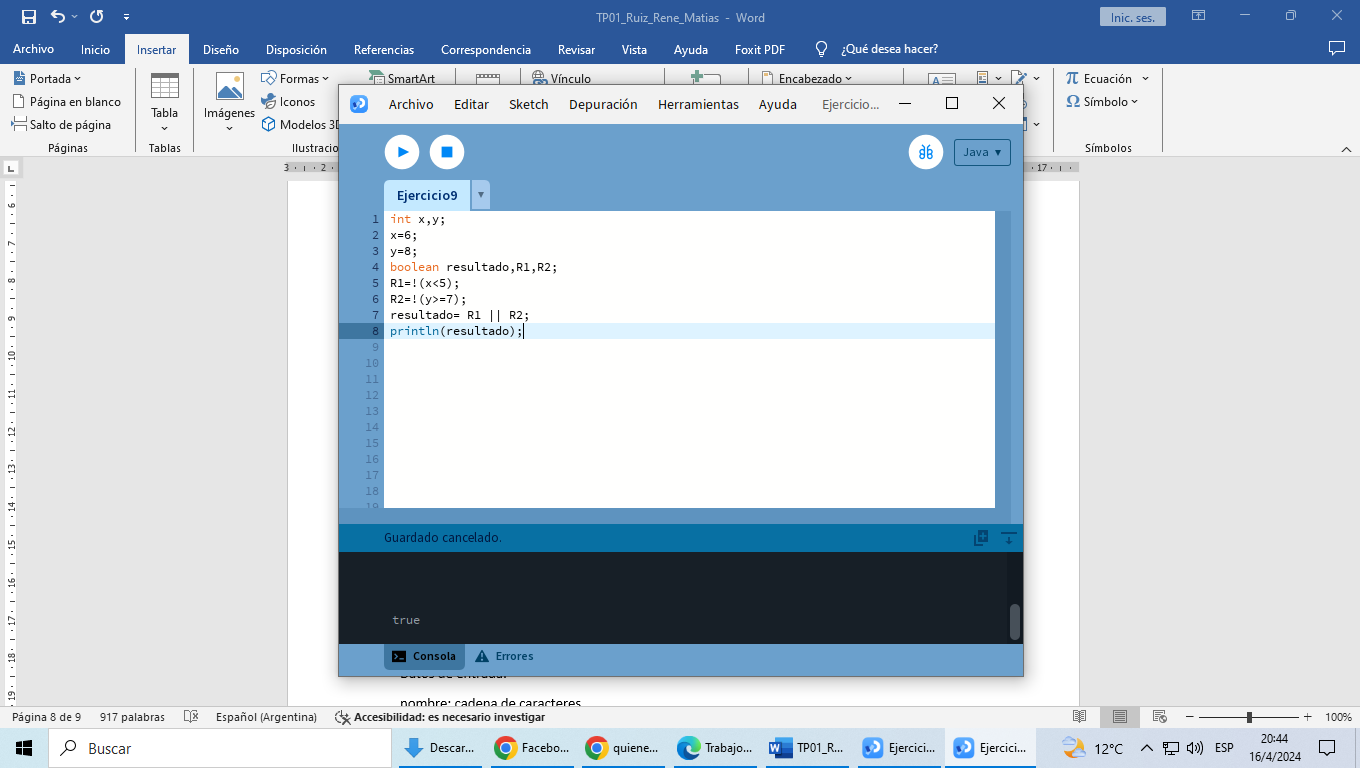


Ejercicio 9: Para x=6, y=8, evaluar elresultado de !(x<5) || !(x=7)

!(6<5) ||!(7>=7)

V || F

V



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

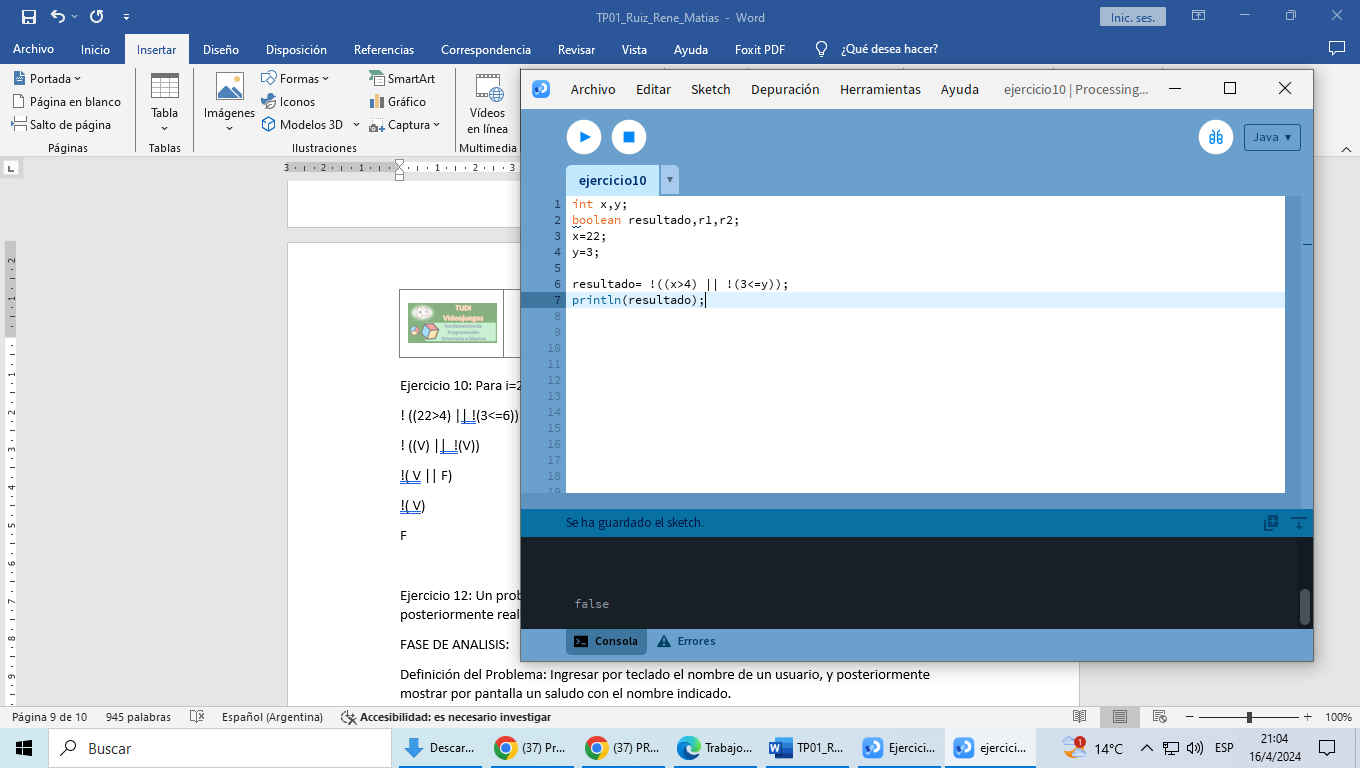
! ((22>4) || !(3<=6))

! ((V) || !(V))

!( V || F)

!( V)

F



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)

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

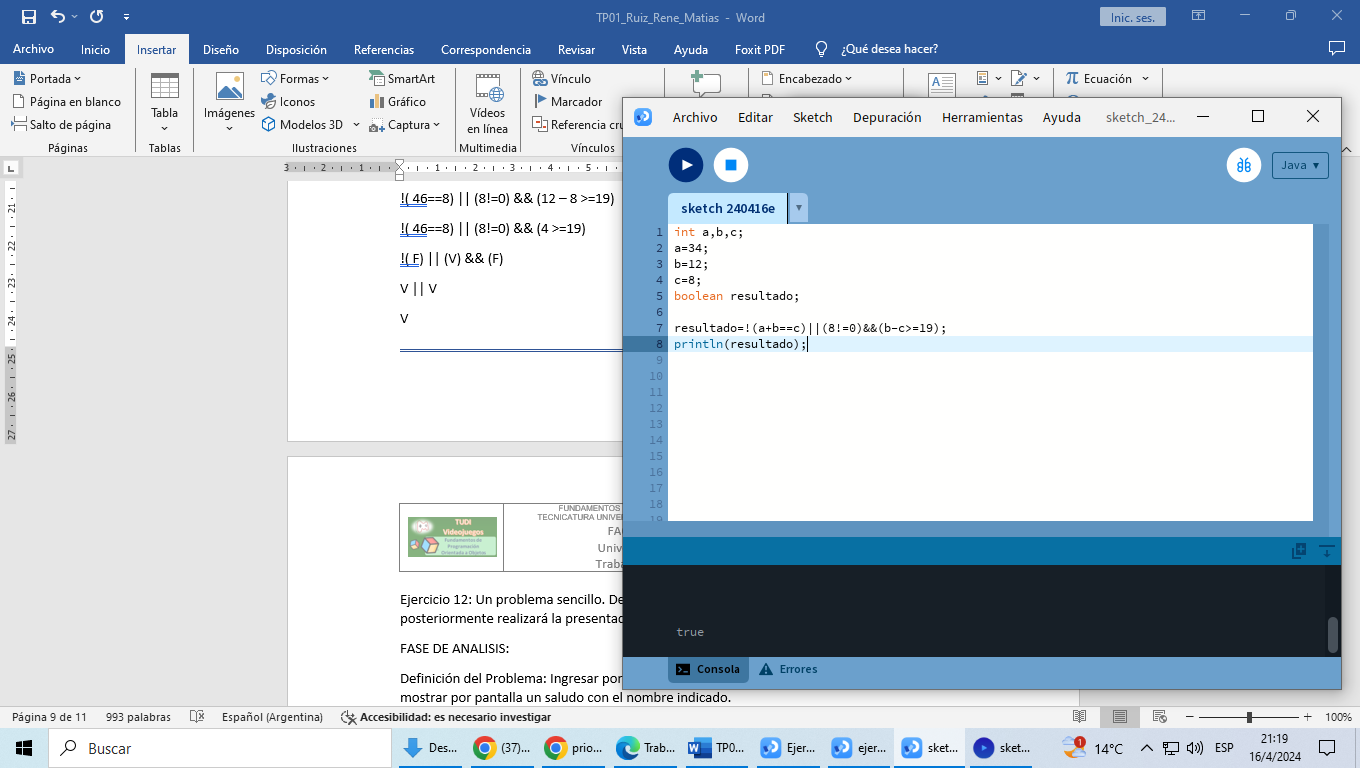
!( 46==8) || (8!=0) && (12 – 8 >=19)

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

!( F) || (V) && (F)

V || V

V



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.

FASE DE ANALISIS:

Definición del Problema: Ingresar por teclado el nombre de un usuario, y posteriormente mostrar por pantalla un saludo con el nombre indicado.

Datos de entrada:

nombre: cadena de caracteres

datos de salida:

saludo por pantalla con el nombre ingresado por teclado.

Proceso:

Programar para que con el nombre ingresado, salga un saludo final.

|  |
| --- |
| Entidad: computadora |
| Variables:  nombre : cadena de caracteres |
| Algoritmo: Armar\_saludo  inicio  Ingreso el nombre  Muestro saludo + nombre  fin |

Ejercicio 13: Será común resolver problemas utilizando variables. Calcule el perímetro y área de un rectángulo dada su base y su altura

FASE DE ANALISIS:

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

*Datos de entrada*:

alto,largo : float

*datos de salida*:

El perímetro y el Área de un rectangulo.

Proceso:

Perímetro= alto \*2 + largo\*2

Area= alto\*largo

|  |
| --- |
| ENTIDAD QUE RESUELTE EL PROBLEMA: Calculadora |
| VARIABLES:  Alto, largo, perímetro, área: Real |
| Nombre del algoritmo: calcular\_area  Ingresar el alto y el largo del rectangulo  area= alto\*largo  mostrar area |
| Nombre del algoritmo: calcular\_perimetro  Ingresar el alto y el largo del rectángulo  perímetro= alto \* 2 + largo \* 2  mostrar perimetro |

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.

FASE DE ANALISIS:

*Definición del Problema*: Calcular la hipotenusa de un triangulo rectangulo, conociendo sus catetos.

*Datos de entrada*:

catetoa,catetob: real

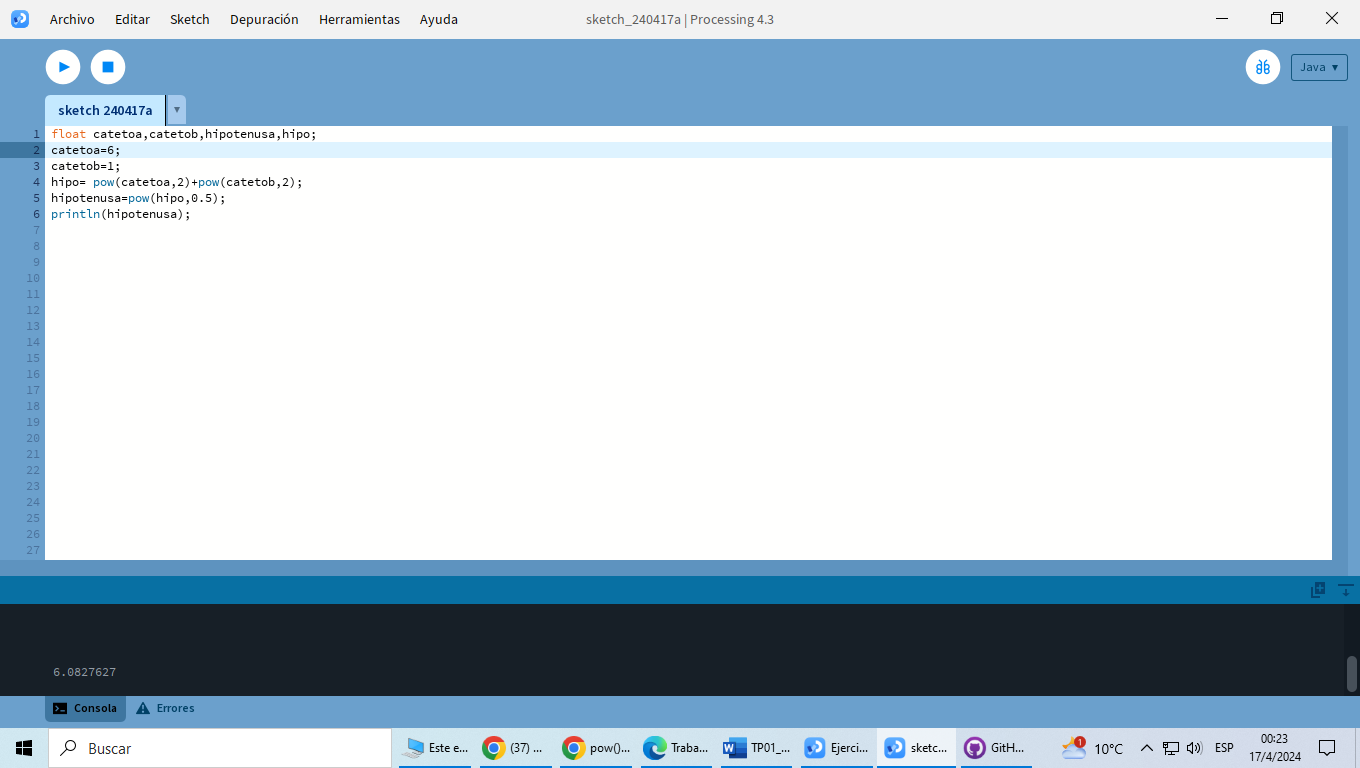
*datos de salida*:

hipotenusa: Real

Proceso:

Hipotenusa2 = catetoa2+catetob2

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA: calculadora |
| Variables:  catetoa,catetob,hipotenusa:real |
| NOMBRE DEL ALGORITMO: calcular\_hipotenusa  PROCESO DEL ALGORITMO:  Ingresar los catetos del triangulo  Leer catetoa, catetob  hipotenusa: (catetoa2 +catetob2)1/2  mostrar hipotenusa |



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.

FASE DE ANALISIS:

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

*Datos de entrada*:

a,b: real // números que se ingresa

datos de salida:

suma, resta, multiplicación y división de los dos números.

Proceso:

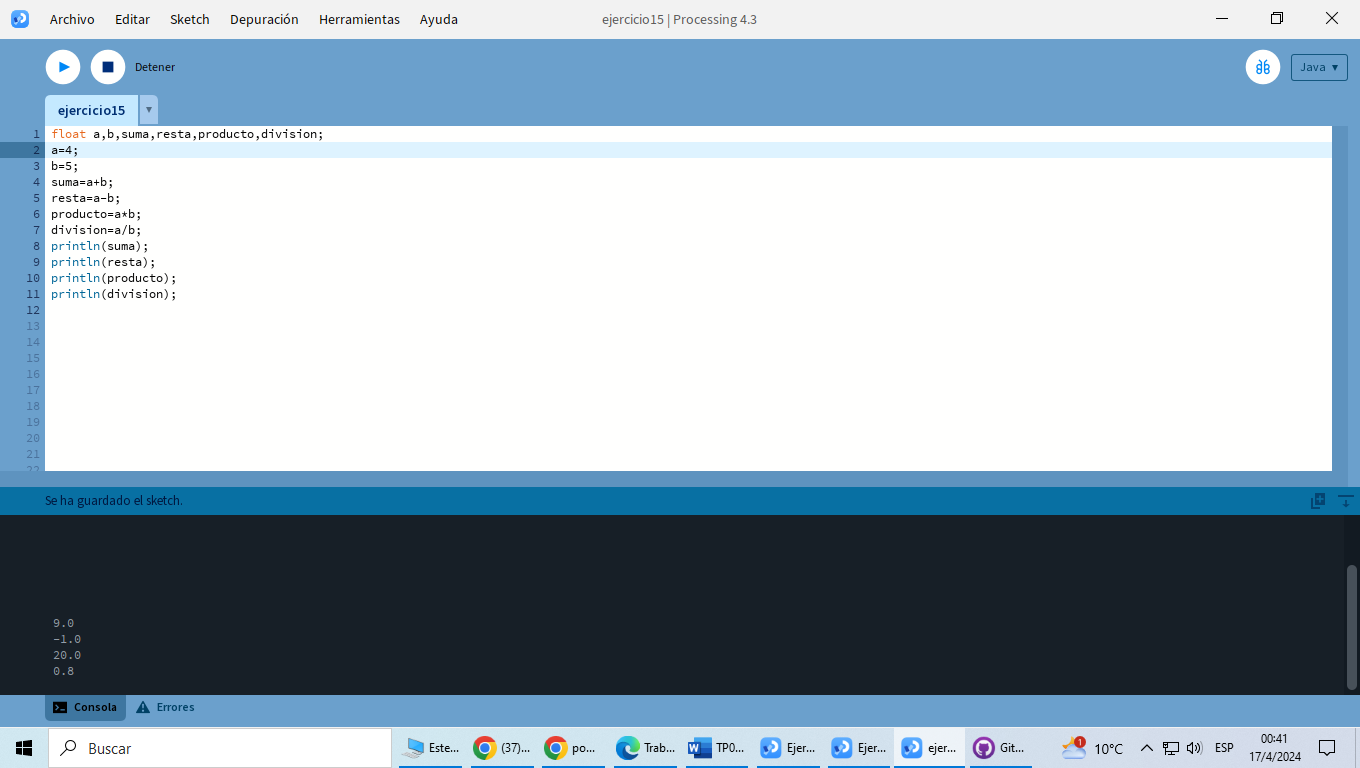
suma=a+b

resta=a-b

multiplicación=a\*b

división=a/b

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA: calculadora |
| VARIABLES:  a,b,suma,resta,producto,división: real |
| NOMBRE DEL ALGORITMO: calcular\_suma  PROCESO DEL ALGORITMO  Leer a,b  suma=a+b  mostrar suma |
| NOMBRE DEL ALGORITMO: calcular\_resta  PROCESO DEL ALGORITMO:  Leer a,b  resta:a-b  mostrar resta |
| NOMBRE DEL ALGORITMO: calcular\_producto  PROCESO DEL ALGORITMO:  Leer a,b  producto:a\*b  mostrar producto |
| NOMBRE DEL ALGORITMO: calcular\_division  PROCESO DEL ALGORTIMO:  Leer a,b  division:a/b  mostrar division |



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

temperaturaCelsius =(temperaturaFahrenheit -32)/1.8

FASE DE ANALISIS:

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

*Datos de entrada*:

temperaturaFahrenheit: real

datos de salida:

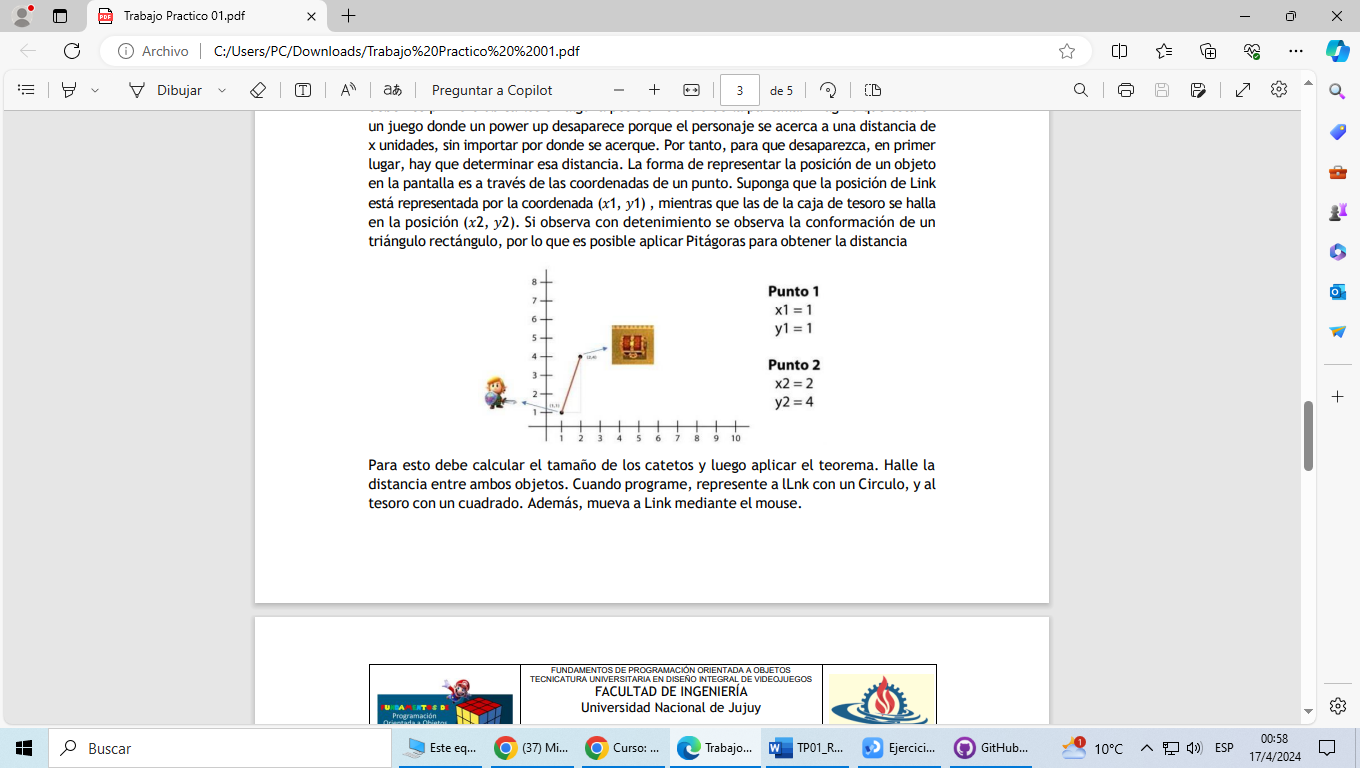
temperaturaCelsius

Proceso:

temperaturaCelsius =(temperaturaFahrenheit -32)/1.8

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA: Calculadora |
| VARIABLES:  temperaturaFahrenheit, temperaturaCelsius: real |
| NOMBRE DEL ALGORITMO: convertir\_celcius  PROCESO DEL ALGORITMO:  Leer temperaturaFahrenheit  temperaturaCelsius= (temperaturaFahrenheit -32)/1.8  mostrar temperaturaCelsius |

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



FASE DE ANALISIS:

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

Datos de entrada:

posicionLink, posicionCaja: coordenadas

datos de salida:

distancia entre link y caja

Proceso:

Calcular lado1 = y2 -y1

Calcular lado2= x2-x1

Hipotenusa=(lado12+lado22)1/2

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA calculadora |
| VARIABLES:  X2,x1,y2,y1,lado1,lado2,hipotenusa |
| NOMBRE DEL ALGORITMO: calcular\_hipotenusa  PROCESO DEL ALGORITMO:  Leer las coordenadas de la posición de Link y caja del tesoro  Lado1= y2 -y1  lado2= x2-x1  Hipotenusa=(lado12+lado22)1/2  Mostrar hipotenusa |

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.

FASE DE ANALISIS:

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

Datos de entrada:

Ecuación de segundo grado

datos de salida:

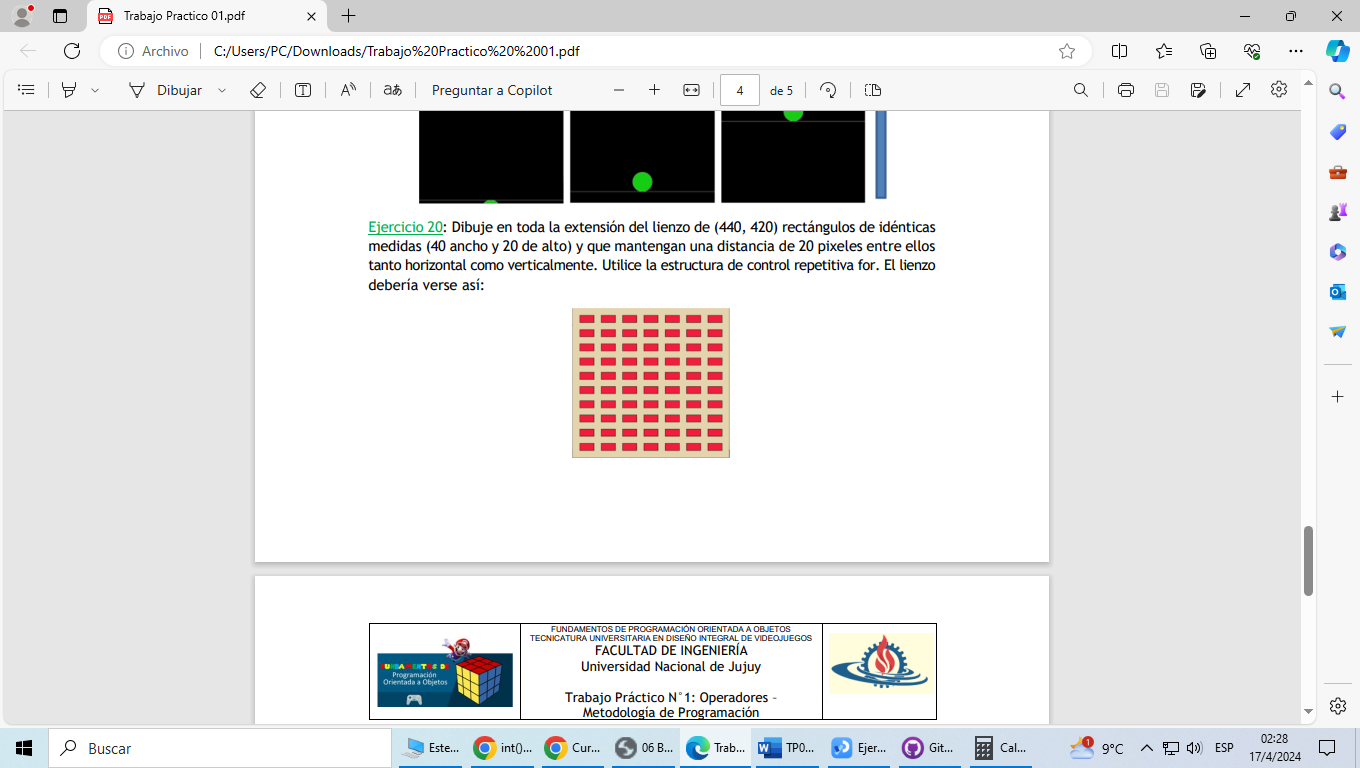
raíces de la ecuación de segundo grado

Proceso:

X2 y x1 =

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA: CALCULADORA |
| VARIABLES:  a,b,c: real  determinante:real  =raíz 2  x1=raíz 1 |
| NOMBRE DEL ALGORITMO: calcular\_raices  PROCESO DEL ALGORITMO:  Leer a,b,c  Discriminante=  Según sea discriminante hacer  Caso determinante>0:  x2=  x1=  Caso determinante=0:  X2 y x1 =  Caso determinante < 0:  Mostrar No tiene solución en los números reales |

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:



FASE DE ANALISIS:

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

Datos de entrada:

Tamaño del lienzo, alto y ancho del rectangulo, distancia entre rectangulos

datos de salida:

dibujo en todo el lienzo rectángulos de

Proceso:

Dibujar un rectangulo.

Dibujar una fila de rectángulos, continuar hasta ocupar la extensión de todo el lienzo

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA: lienzo |
| VARIABLES:  X,y : coordenadas  Alto,ancho, distanciaEntreRec: real  altolienzo, ancholienzo  altolienzo<-440;  anchoLienzo<-420;  distanciaEntreRec=20;  ancho=40;  alto=20;  para ( x=coordenadasx; x<ancho; x con paso (ancho+distanciaEntreRec)){  para ( y=coordenadasy; y<alto; y con paso(alto+distanciaEntreRec)){    dibujar rectangulo en (x,y,ancho,alto) |

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

FASE DE ANALISIS:

Definicion del Problema: Dibujar escalones sobre el lienzo y sobre cara escalon un punto rojo

Analisis:

Datos de Entrada:

puntoA, puntoB, puntoC, puntoD: coordenadas cartesianas

Datos de salida:

El dibujo de la línea horizontal

El dibujo de la línea vertical

El dibujo del punto

Proceso:

Dibujar una línea horizontal entre los puntos A y B

Dibujar una línea vertical entre los puntos B y C

Dibujar un punto en la posición x ==posicion en x de B, y = posición en y de b – 5 unidades

Actualizar las coordenadas del puntoA con las del puntoC

Repetir desde el principio hasta que la coordenada en y de puntoA sea mayor que el alto del lienzo.

|  |
| --- |
| Entidad: Programa |
|  |
|  |

FASE DE DISEÑO

|  |
| --- |
| ENTIDAD QUE RESUELVE EL PROBLEMA:Escalon |
| Variables:  puntoA, puntoB, puntoC, puntoD: coordenadas cartesianas |
| Nombre del ALGORITMO: dibujar\_escalon  PROCESO DEL ALGORITMO:  ALGORITMO:  Dibujar una línea horizontal entre los puntos A y B  Dibujar una línea vertical entre los puntos B y C  Dibujar un punto en la posición x ==posicion en x de B, y = posición en y de b – 5 unidades  fin |
| Nombre del ALGORITMO: dibujar\_circulo  PROCESO DEL ALGORITMO:  ALGORITMO:  Inicio  Dibujar un punto en la posición: x= posición en x de B, y= posición en y de b -5  fin |
| NOMBRE DEL ALGORITMO: actualizar\_coordenadas  AlGORITMO:  Inicio  puntoA.x <- puntoC.x  puntoA.y <- puntoC.y  FIN |

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)