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

Trabajo Práctico

N° 1

Apellido y Nombre: Barrientos Fabricio Alejandro LU: TUV000469

*Profesores:*

*Mg. Ing. Ariel Alejandro Vega*

*Ing. Carolina Cecilia Apaza*

*Año*

**FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS**

Indice

Contenido

[Ejercicio 1 4](#_Toc164108652)

[Ejercicio 2 4](#_Toc164108653)

[Ejercicio 3 5](#_Toc164108654)

[Ejercicio 4 6](#_Toc164108655)

[Ejercicio 5 6](#_Toc164108656)

[Ejercicio 6 9](#_Toc164108657)

[Ejercicio 7 10](#_Toc164108658)

[Ejercicio 8 11](#_Toc164108659)

[Ejercicio 9 12](#_Toc164108660)

[Ejercicio 10 13](#_Toc164108661)

[Ejercicio 11 14](#_Toc164108662)

[Ejercicio 12 14](#_Toc164108663)

[FASE DE ANALISIS 14](#_Toc164108664)

[FASE DE DISEÑO 15](#_Toc164108665)

[EJERCICIO 13 15](#_Toc164108666)

[FASE DE ANALISIS 16](#_Toc164108667)

[FASE DE DISEÑO 16](#_Toc164108668)

[EJERCICIO 14 17](#_Toc164108669)

[FASE DE ANALISIS 18](#_Toc164108670)

[FASE DE DISEÑO 18](#_Toc164108671)

[EJERCICIO 15 19](#_Toc164108672)

[FASE DE ANALISIS 19](#_Toc164108673)

[FASE DE DISEÑO 20](#_Toc164108674)

[EJERCICIO 16 21](#_Toc164108675)

[FASE DE ANALISIS 22](#_Toc164108676)

[FASE DE DISEÑO 22](#_Toc164108677)

[EJERCICIO 17 23](#_Toc164108678)

[FASE DE ANALISIS 24](#_Toc164108679)

[FASE DE DISEÑO 24](#_Toc164108680)

[EJERCICIO 18 26](#_Toc164108681)

[FASE DE ANALISIS 26](#_Toc164108682)

[FASE DE DISEÑO 26](#_Toc164108683)

[EJERCICIO 19 28](#_Toc164108684)

[FASE DE ANALISIS 29](#_Toc164108685)

[FASE DE DISEÑO 30](#_Toc164108686)

[Ejercicio 20 31](#_Toc164108687)

[FASE DE ANALISIS 31](#_Toc164108688)

[FASE DE DISEÑO 32](#_Toc164108689)

[EJERCICIO 21 33](#_Toc164108690)

[FASE DE ANALISIS 34](#_Toc164108691)

[FASE DE DISEÑO 34](#_Toc164108692)

[EJERCICIO 22 36](#_Toc164108693)

[FASE DE ANALISIS 36](#_Toc164108694)

[FASE DE DISEÑO 37](#_Toc164108695)

[Conclusión 39](#_Toc164108696)

[Fuentes bibliográficas 39](#_Toc164108697)

# Ejercicio 1

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

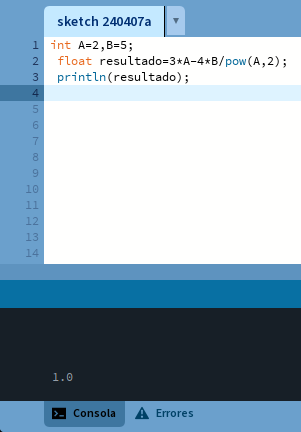
3\* A - 4 \* B / A ^ 2

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

6-(20/4)

6-5

1



# 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)

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

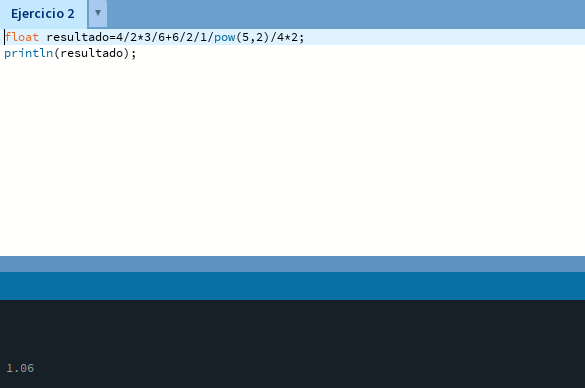
6/6+(3/(25\*4)\*2

1+3/100\*2

1+0,03\*2

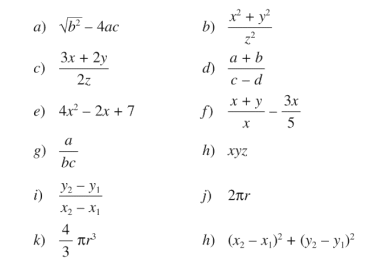
1+0,06

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.



a) (b^2) ^(1/2)-4\*a\*c

b) ((x^2)+(y^2))/(z^2)

c) (3\*x+2\*y)/(2\*z)

d) (a+b)/(c-d)

e) 4\*(x^2)-2\*x+7

f) ((x+y)/x)-(3\*x)/5

g) a/(b\*c)

h) X\*y\*z

i) (y-y)/(x-x)

j) 2\*3,14\*r

k) (4/3)\*(3,14\*(r^3))

# 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 //

b) 3 \* X ^ 4 – 5 \* X ^ 3 + 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:

1. B \* A – B ^ 2 / 4 \* C

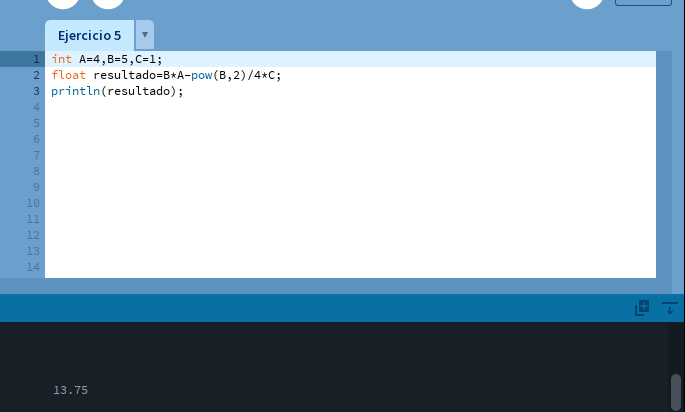
(5\*4)-(5^2/4\*1)

20-25/4\*1

20-25/4

20-6,25

13,75

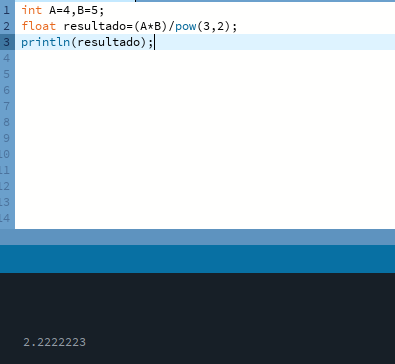


1. (A \* B) / 3 ^ 2

(4\*5)/3^2

20/9

2,2222223



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

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

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

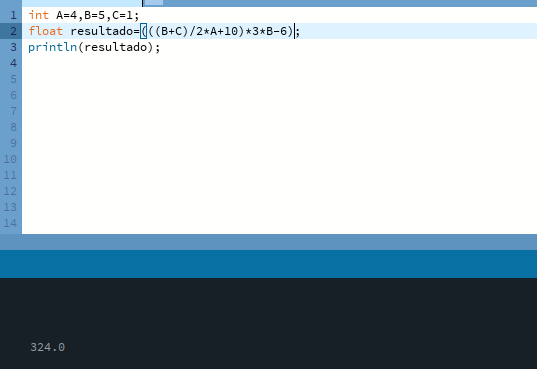
((3\*4+10)\*3\*5)-6

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

(22\*3\*5)-6

330-6

324

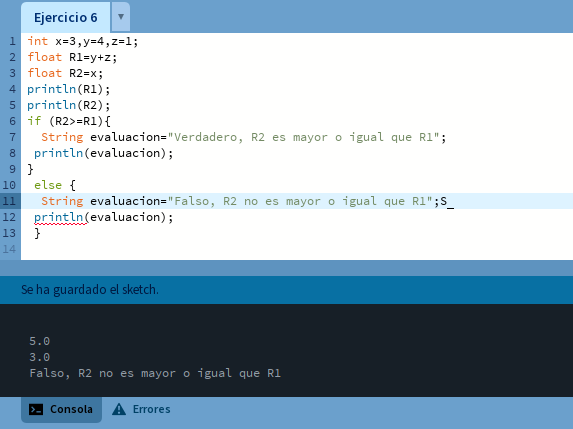


# Ejercicio 6

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

R1 = y+z

R2 = x >= R1

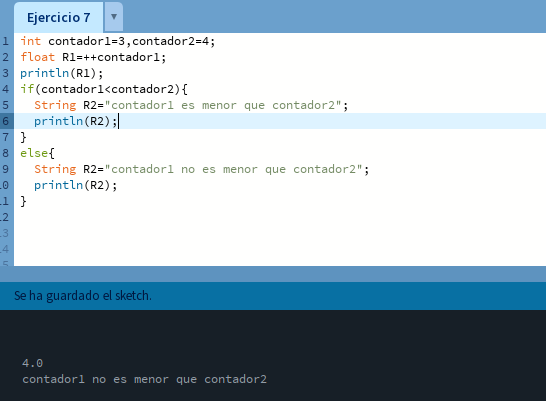


# Ejercicio 7

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

R1 = ++contador1

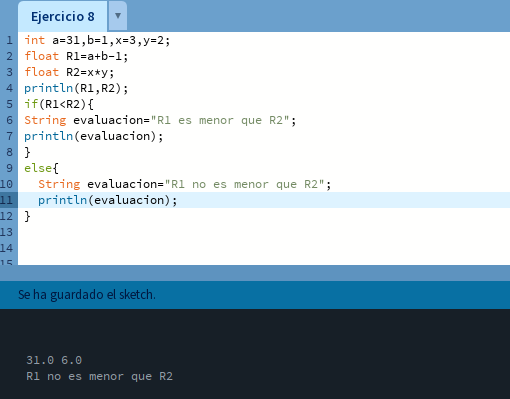
R2 = contador1 < contador2



# Ejercicio 8

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

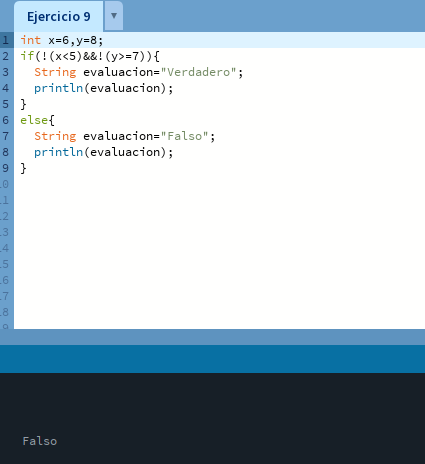
a+b-1 < x\*y



# Ejercicio 9

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

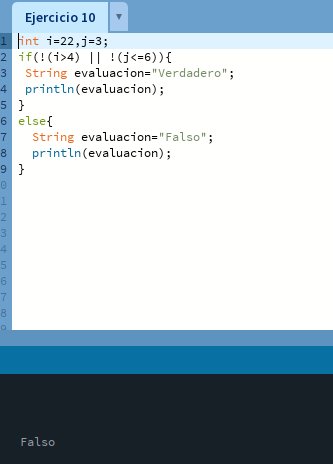
!(x<5)CC !(y>=7)



# Ejercicio 10

Para i=22,j=3, evaluar el resultado de

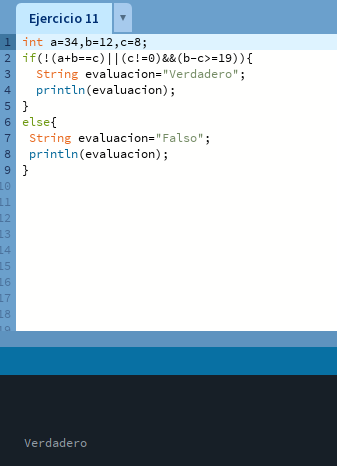
!((i>4) || !(j<=6))



# Ejercicio 11

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

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



# 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

Especificacion del Problema

Mostrar un nombre escrito por teclado en pantalla con un saludo mas el nombre indicado

ANALISIS

Datos de Entrada:

nombre: String

Datos de Salida

El saludo junto con el nombre

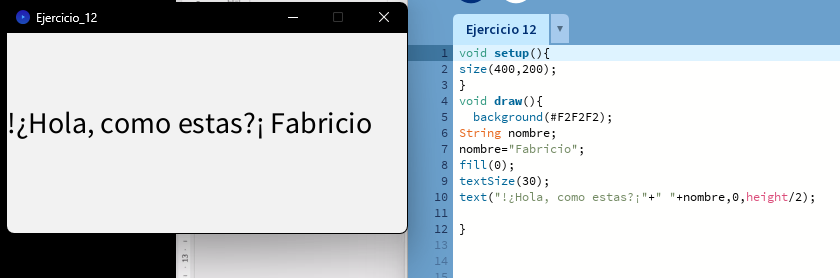
Proceso

Se le pide un nombre al usuario y se lo combina con un saludo

### FASE DE DISEÑO

|  |
| --- |
| Entidad: Lienzo |
| Variables:  nombre: String |
| Nombre Algoritmo: mostrar\_nombre  Inicio  anchoLienzo=440  altoLienzo=200  Leer nombre  Asigna un color al texto en pantalla  Asigna un tamaño al texto en pantalla  Muestra un texto en pantalla que es igual a =saludo+nombre junto con su posición en x e y  fin |

CODIFICACION



# 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

Especificacion del Problema

Calcular el perímetro y área de un rectángulo dada su base y altura

ANALISIS

Datos de Entrada

baseRect=int

alturaRect=int

Datos de Salida

perimetroRect=float

areaRect=float

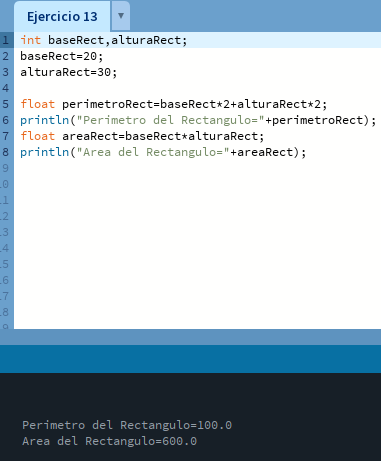
Proceso

Obtener perímetro y área del rectángulo usando las formulas de obtención de perímetro y área.

### FASE DE DISEÑO

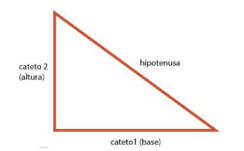
|  |
| --- |
| Entidad: Calculadora |
| Variables:  baseRect:int  alturaRect:int  perimetroRect=float  areaRect=float |
| Nombre Algoritmo: calcular\_perimetro  Inicio  Leer baseRect  Leer alturaRect  Calcular perímetro del rectángulo usando la formula (base\*2+altura\*2)  Leer perimetroRect  Mostrar resultado en consola  --------------------------------  Nombre Algoritmo: calcular\_area  Inicio  Leer baseRect  Leer alturaRect  Calcular el área del rectángulo usando la formula (base\*altura)  Leer areaRect  Mostrar resultado en consola  Fin |

CODIFICACION



# 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

Especificacion del Problema

Calcular la hipotenusa de un triangulo rectangulo

ANALISIS

Datos de Entrada

cateto1=int

cateto2=int

Datos de Salida

hipotenusaTriangulo=float

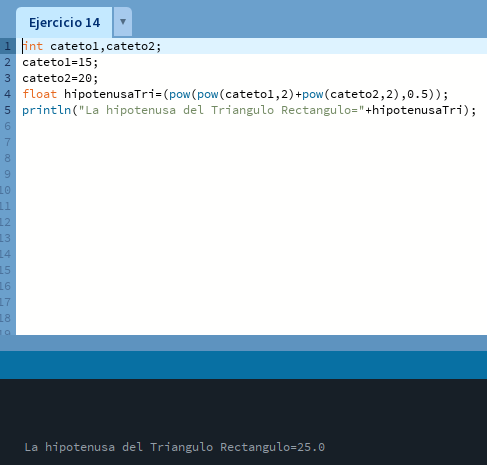
Proceso

Se utiliza el teorema de pitagoras para calcular la hipotenusa

### FASE DE DISEÑO

|  |
| --- |
| Entidad: calculadora |
| Variables  cateto1=int  cateto2=int  hipotenusaTri=float |
| Nombre Algoritmo: calcular\_Hipotenusa  Inicio  Leer cateto1  Leer cateto2  Calcular hipotenusaTri con el teorema de pitagoras  Leer hipotenusaTri  Mostrar resultado en consola  fin |

CODIFICACION



# 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

Especificacion del Problema

Realizar una suma,resta,división,multiplicación dados dos números.

ANALISIS

Datos de Entrada

Numero1=int

Numero2=int

Datos de Salida

Suma=int

Resta=int

Multiplicacion=int

División=int

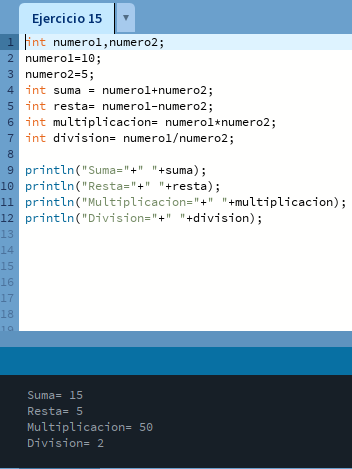
Proceso

Sumar,restar,dividir,multiplicar los números dados

### FASE DE DISEÑO

|  |
| --- |
| Entidad: calculadora |
| Variables:  Numero1=int  Numero2=int  Suma=int  Resta=int  Multiplicacion=int  División=int |
| Nombre Algoritmo: calcular\_suma  Inicio  Leer numero1  Leer numero2  Sumar los dos números num1+num2  Mostrar resultado en consola  Fin  ----------------------------------------------------  Nombre Algoritmo:calcular\_resta  Inicio  Leer numero1  Leer numero2  Restar los dos números num1-num2  Mostrar resultado en consola  Fin  --------------------------------------------------------  Nombre Algoritmo: calcular\_multiplicacion  Inicio  Leer numero1  Leer numero2  Multiplicar los dos números num1\*num2  Mostrar resultado en consola  Fin  -----------------------------------------------------  Nombre Algoritmo:calcular\_division  Inicio  Leer numero1  Leer numero2  Dividir los dos números num1/num2  Mostrar resultado en consola  Fin |

CODIFICACION



# 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.



## FASE DE ANALISIS

Especificacion del Problema

Convertir una temperatura Fahrenheit en grados celsius

ANALISIS

Datos de Entrada

tempFahrenheit=float

Datos de Salida

tempCelsius=float

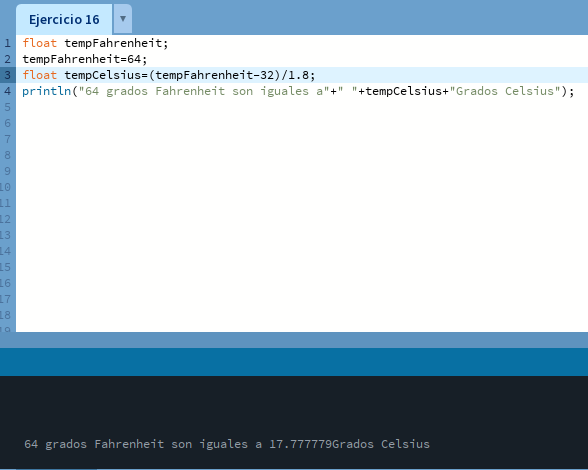
Proceso

Usar la formula de conversión para obtener el resultado

### FASE DE DISEÑO

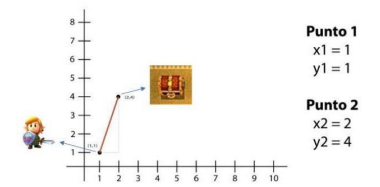
|  |
| --- |
| Entidad: calculadora |
| Variables  tempFahrenheit=float  tempCelsius=float |
| Nombre Algoritmo: calcular\_Fahrenheit\_a\_Celsius  Inicio  Leer tempFahrenheit  Calculamos su valor en grados Celsius utilizando la formula ((tempFahrenheit-32)/1.8)  Leer tempCelsius  Mostramos el resultado en consola  Fin |

CODIFICACION



# 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 Link con un Circulo, y al tesoro con un cuadrado. Además, mueva a Link mediante el mouse.

## FASE DE ANALISIS

Especificacion del Problema

Se busca usar hacer que el cofre desaparezca cuando link se encuentre a una determinada distancia pero para eso hay que hayar la distancia entre link y el cofre y además poder mover a link con el mouse

ANALISIS

Datos de Entrada

X1=float

X2=float

Y1=float

Y2=float

Datos de Salida

distEntreLinkCofre=float

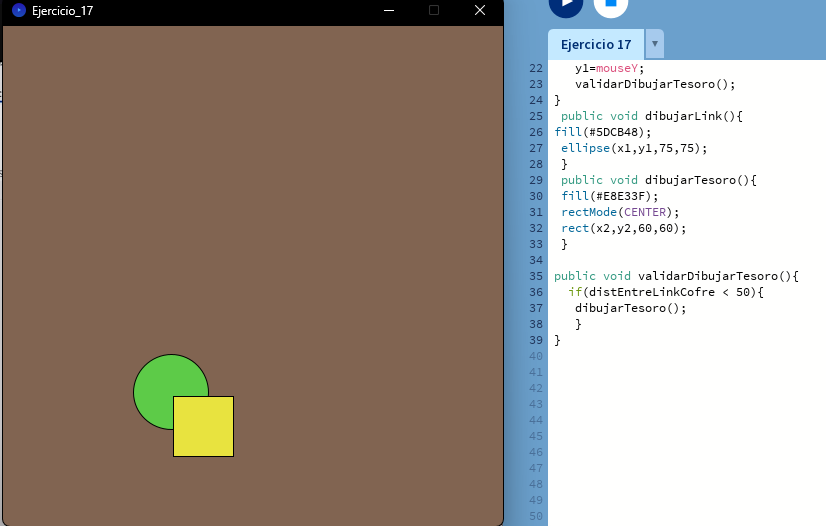
Proceso

Primero debemos saber la distancia entre link y el cofre usando teoremas de pitagoras formando los catetos necesarios con las posiciones de los objetos,

### FASE DE DISEÑO

|  |
| --- |
| Entidad:Lienzo y calculadora |
| Variables:  X1=float  X2=float  Y1=float  Y2=float  Cateto1=float  Cateto2=float  distEntreLinkCofre=float |
| Nombre Algoritmo: calcular\_distancia\_cofre\_link  Inicio  Leer cateto1  Leer cateto2  Calcular distancia entre link y el cofre usando teorema de pitagoras con los catetos  Leer distEntreLinkCofre  Mostrar resultado en pantalla  Fin  -----------------------------------------------------  Nombre Algoritmo:desaparecer\_cofre  Inicio  AnchoLienzo=500  AltoLienzo=500  Se dibuja un circulo que representa a Link  circuloX=x1  circuloY=y1  anchoCirc=75  altoCirc=75  Para poder mover a link con el mouse  A x1 se le asigna el eje x del mouse  A y1 se le asigna el eje y del mouse  Se dibuja un cuadrado que representa al tesoro  cuadradoX=x2  cuadradoY=y2  AnchoCuad=60  AltoCuad=60  Si la distEntrLinkCofre es menor que 50 se dibujara un tesoro sino se cumple no lo hará  Fin |

CODIFICACION



# 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

Especificacion del Problema

Obtener las raíces de una ecuación de segundo grado y analizar su discriminante

ANALISIS

Datos de Entrada

A=int

B=int

C=int

Datos de Salida

Discriminante=float

Raiz1=float

Raiz2=float

Proceso

Se resuelven las raíces y se analiza la discriminante

### FASE DE DISEÑO

|  |
| --- |
| Entidad:calculadora |
| Variables  A=int  B=int  C=int  X1=float  X2=float  Discriminante=float |
| Nombre Algoritmo: obtener\_raices  Inicio  Leer a  Leer b  Leer c  Calcular la primera raíz  Leer x1  Mostrar resultado en pantalla  Calcular segunda raíz  Leer x2  Mostrar resultado en pantalla  Fin  --------------------------------------  Nombre Algoritmo: analizar\_discriminante  Inicio  Calcular discriminante  Según discrimnante  Sea mayor que 0  Mostrar en consola que las raíces son reales y diferentes  Romper  En el caso que sea igual a 0  Mostrar en consola las raíces son reales e iguales  romper  En el caso que sea menor que 0  Mostrar en consola las raíces son complejas  Romper  Fin |

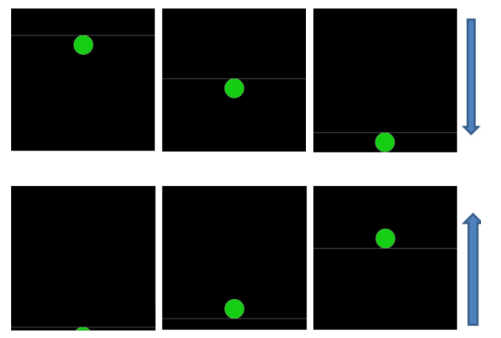
CODIFICACION



No pude saber como terminar este problema sin usar variables tipo int ni float para usar switch asi que no sabría como analizar la discriminante.

# 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.



## FASE DE ANALISIS

Especificacion del Problema

Dibujar una línea que va de arriba hacia abajo constantemente con un circulo en la parte medio de esta que cambie de lugar según hacia donde se mueva ósea hacia arriba o abajo

ANALISIS

Datos de Entrada

anchoCirculo=int

altoCirculo=int

Datos de Salida

La Linea extendiéndose por todo el lienzo con el circulo cambiando su lugar de acuerdo hacia donde va

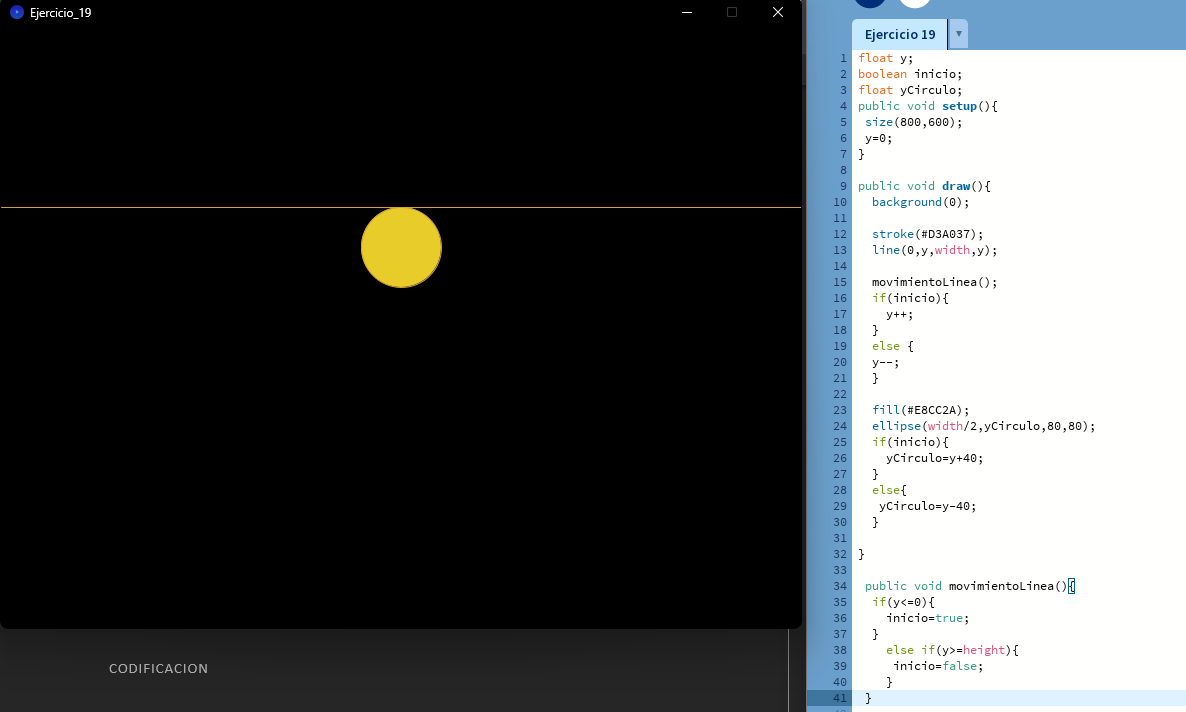
Proceso

Dibujar una línea que se mueva hacia abajo y regrese hacia arriba junto con el circulo cambiando su posición de acuerdo a esto

### FASE DE DISEÑO

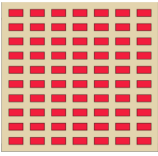
|  |
| --- |
| Entidad: Lienzo |
| Variables:  Y=float  Inicio=boolean  yCirculo=float |
| Nombre Algoritmo: mover\_linea\_circulo  Inicio  Leer y  Leer inicio  Leer yCirculo  AnchoLienzo=800  AltoLienzo=600  Se dibuja una línea (x=0,y,ancho,y)  Donde si y<=0  El valor de y aumenta  Sino el valor de y decrece  Luego se crea un circulo (ancho/2,yCirculo,80,80)  El cual si y<=0  El valor de yCirculo aumenta en 40  Sino el valor de yCirculo decrece en 40  Fin |

CODIFICACION



# 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

Especificacion del problema

Dibujar rectángulos en todo el lienzo teniendo estos una distancia de 20 entre ellos tanto horizontalmente como verticalmente.

ANALISIS

Datos de Entrada

coordenadasRect:coordenadas cartesianas

ancho,alto,distanciaEntreRect: entero

Datos de Salida

Los rectángulos dibujados

Proceso

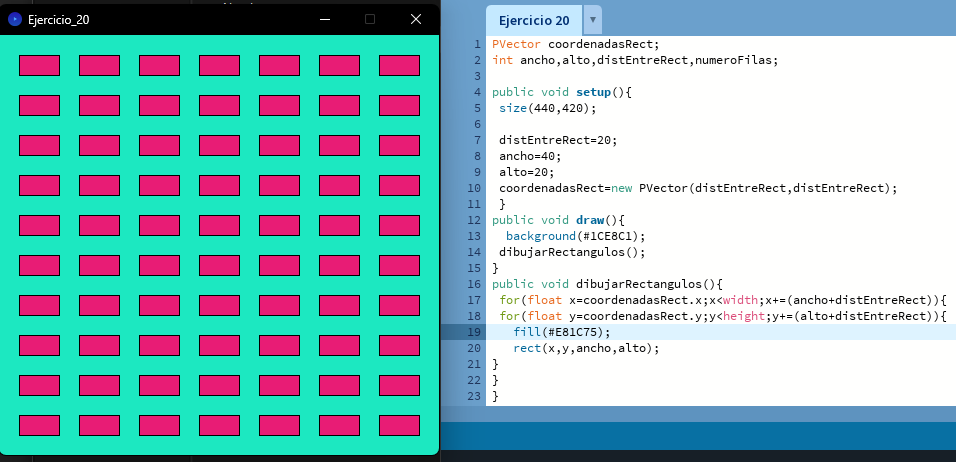
Dibujar los rectángulos

### FASE DE DISEÑO

|  |
| --- |
| Entidad: Lienzo |
| Variables  CoordenadasRect: Coordenadas  Ancho,alto,distEntreRect: int  anchoLienzo,altoLienzo:int |
| Nombre-Algoritmo: Dibujar\_Rectangulos  Algoritmo  Inicio  anchoLienzo🡨440  altoLienzo🡨420  distEntreRect🡨20  ancho🡨40  alto🡨20  Para x🡨coordenadasRect.x hasta anchoLienzo con paso (ancho+distEntreRect)  Para y🡨coordenadasRect.y hasta altoLienzo con paso (alto+distEntreRect)  Hacer  Dibujar un rectángulo en (x,y) con dimensiones ancho y alto  Fin-Para  Fin |

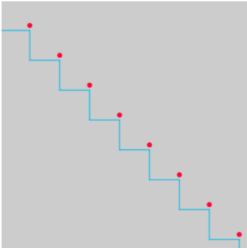
# 

CODIFICACION



# 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.

## FASE DE ANALISIS

Especificacion del Problema

Dibujar líneas que formen escalones y en cada uno un punto.

ANALISIS

Datos de Entrada

PuntoA=PVector

PuntoB=PVector

PuntoC=PVector

PuntoD=PVector

distLinea=float

Datos de Salida

Los escalones dibujados con los puntos

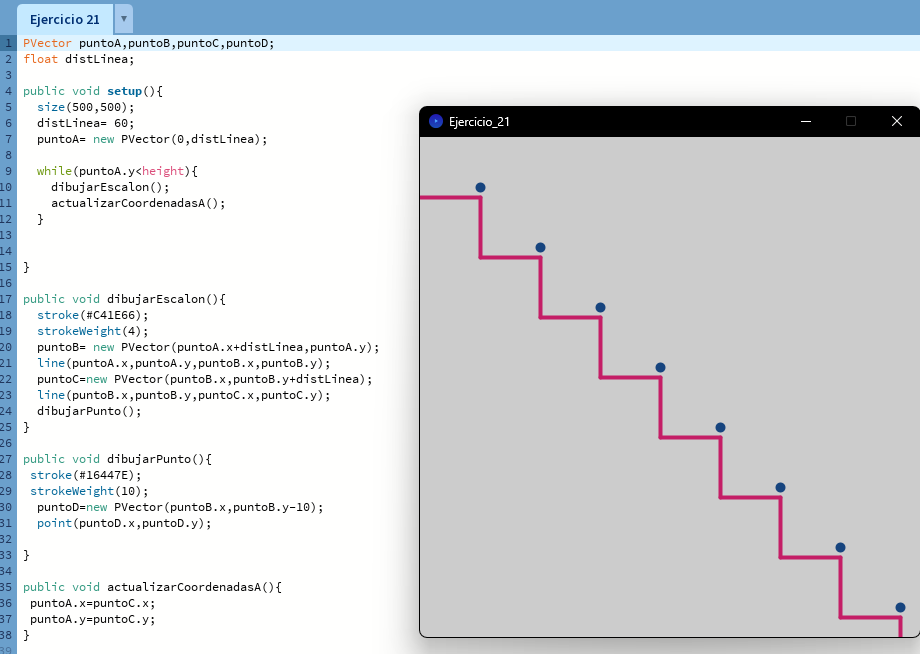
Proceso

Dibujar los escalones y los puntos

### FASE DE DISEÑO

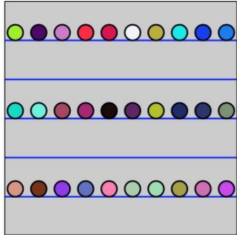
|  |
| --- |
| Entidad: Escalon |
| Variables  puntoA,puntoB,puntoC,puntoD: PVector  distLinea=Entero |
| Nombre Algoritmo:dibujar\_escalon  Inicio  Dibujar una línea horizontal entre los puntos A y B con distancia distLinea  Dibujar una línea vertical entre los puntos B y C con distancia distLinea  Dibujar\_circulo  Fin  ------------------------------------------------  Nombre Algoritmo: Dibujar\_circulo  Inicio  Dibujar un punto en (x=posición en x de B,y=posición en y de B-10  Fin  ---------------------------------------------  Nombre Algoritmo: actualizar\_coordenadas\_A  Inicio  puntoA.x=puntoc.x  puntoA.y=puntoC.y  Fin |

CODIFICACION



# 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.

## FASE DE ANALISIS

Especificacion del Problema

Dividir el lienzo en franjas de igual medida y dibujar los círculos en cada linea intercalada .

ANALISIS

Datos de Entrada

numeroLineas=int

distanciaLineas=int

numeroCirculos=int

distanciaCirculosY=int

Datos de Salida

Las líneas distribuidas junto a los círculos dibujados de forma intercalada

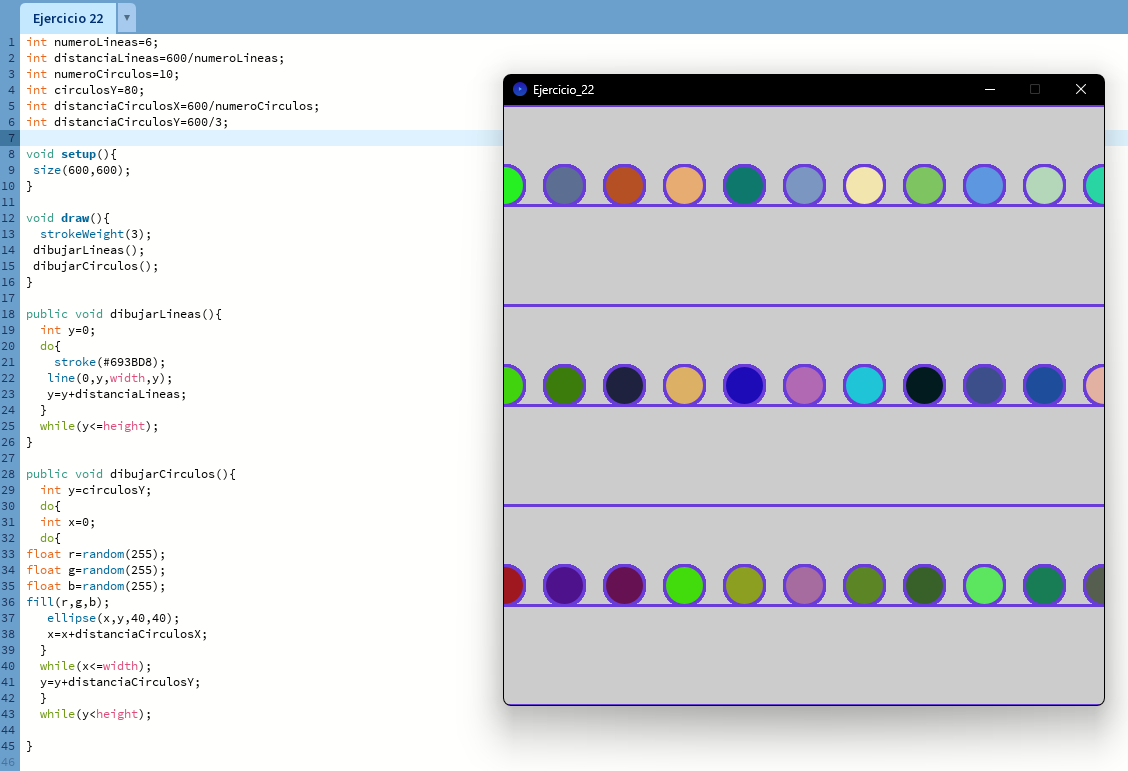
Proceso

Dibujar líneas y círculos

### FASE DE DISEÑO

|  |
| --- |
| Entidad: Lienzo |
| Variables:  numeroLineas=int  distanciaLineas=int  numeroCirculos=int  circulosY=int  distanciaCirculosX=int  distanciaCirculosY=int  y=int  x=int  R=color  G=color  B=color |
| Nombre Algoritmo:Dibujar\_lineas  Inicio  Se le asigna a y el valor de 0  Para hacer una linea (0,y,ancho,y)  Que aumente el valor de y en y=+distanciaLineas  Mientras y sea menor o igual que la altura del lienzo  Fin  ---------------------------------------------------------------------------  Nombre Algoritmo:Dibujar\_circulos  Inicio  Se le asigna a y el valor de circulosY  Y a x el valor de 0  Para hacer un circulo (x,y,40,40)  Que aumente el valor de x en x+distanciaCirculosX  Mientras x sea menor o igual que el ancho del lienzo  Y tambein aumente y en y+distanciaCirculosY  Mientras y sea menor que la altura del lienzo  Fin  -----------------------------------------------------------------------------  Nombre Algoritmo:dibujar\_en\_el\_lienzo\_Lineas\_circulos  Inicio  Se establece el tamaño del lienzo(600,600)  Se aumenta el tamaño de las líneas  Se llama a Dibujar\_Lineas para dibujar las líneas  Se llama a dibujar\_circulos para dibujar los círculos  fin |

CODIFICACION



# Conclusión

Con este trabajo practico pude explorar y conocer mucho mas sobre la programación en processing, los operadores y las metodologías.

# Fuentes bibliográficas

<https://virtual.unju.edu.ar/pluginfile.php/616629/mod_resource/content/5/04%20Expresiones%20aritmeticas.pdf>

<https://virtual.unju.edu.ar/pluginfile.php/616669/mod_resource/content/5/06%20Bifurcaciones%20-%20Expresiones%20Lógicas%20-%20Estructuras%20selectivas.pdf>

<https://youtu.be/Nr3NdAxjqsE>

<https://www.youtube.com/watch?v=_YblzDgoAus>

<https://www.youtube.com/watch?v=YHpW_HMP_PE>

<https://virtual.unju.edu.ar/pluginfile.php/617801/mod_resource/content/3/08%20Estructuras%20Iterativas.pdf>

<https://www.youtube.com/watch?v=dysxlptjAx0>

<https://www.youtube.com/watch?v=4RQV0gMl8y8>

<https://www.youtube.com/watch?v=kApd8u4l0tU>

<https://www.youtube.com/watch?v=r-Olf82ViNE>

<https://virtual.unju.edu.ar/pluginfile.php/633254/mod_resource/content/1/Tutorial%20Processing.pdf>

<https://virtual.unju.edu.ar/pluginfile.php/633255/mod_resource/content/2/Glosario%20Processing.pdf>

<https://virtual.unju.edu.ar/pluginfile.php/633256/mod_resource/content/3/Libro%20Hola%20Mundo%20con%20Processing.pdf>

<https://www.youtube.com/watch?v=ZpLcVUO_rlk>

<https://www.youtube.com/watch?v=BxrJmKdPHRs>

<https://www.youtube.com/watch?v=2UbdPiqAiHY&t=378s>

<https://www.youtube.com/watch?v=l3xL2MA8hSs&t=460s>

<https://www.youtube.com/watch?v=Z0FYCDAC458&t=41s>