|  | TECNICATURA UNIVERSITARIA EN DISEÑO INTEGRAL DE VIDEOJUEGOS  FACULTAD DE INGENIERÍA  Universidad Nacional de Jujuy |  |
| --- | --- | --- |

**FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS**

Trabajo Práctico

N°1

Apellido y Nombre :

Ibáñez Esteban Agustín

*Profesores:*

*Mg. Ing. Ariel Alejandro Vega*

**REGLAMENTO:**

Crear una carpeta denominada TP01\_XXXX donde XXXX es el apellido\_nombre del

estudiante. Al producto final, subirlo en su repositorio y compartir el enlace en formulario.

**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

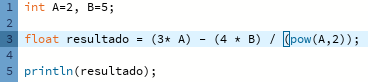
Resolución necesaria en Word:

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

6-(4\*B/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

| Aritmética | Algebraica |
| --- | --- |
| 4 / 2 \* 3 / 6 + 6 / 2 / 1 / 5 ^ 2 / 4 \* 2  (((4/2) \* 3) /6) + ((((6/2) /1) / (5^2)) /4) \*2  1,0 + 0,06  1,06 |  |

Captura del processing y resultado:

|  |  |
| --- | --- |

**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 A=2,B=3,C=4

| Aritmética | Algebraica |
| --- | --- |
| b ^ 2 – 4 \* a \* c  ((3^ 2)) - (4\* 2\* 4)  9 - 32  -23 | 3² - 4.2.4  -23 |

Captura del processing y resultado:

|  |  |
| --- | --- |

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

| Aritmética | Algebraica |
| --- | --- |
| 3 \* X ^ 4 – 5 \* X ^ 3 + X 12 – 17  3\*3 ^ 4 -5 \* 3^ 3 + 3 \* 12 -17  243 - 135 + 36 -17  108 + 19  127 | 3 . 3⁴ - 5.3³+ 3.12 -17  3. 81- 5.27 + 36 -17  243 - 135 + 36 - 17  108 + 19  127 |

Captura del processing y resultado:

|  |  |
| --- | --- |

c) (b + d) / (c + 4) b=2 c=6 d=4

| Aritmética | Algebraica |
| --- | --- |
| (b + d) / (c + 4)  (2 + 4)/ (6 + 4)  6/10  0.6 | 2 + 4 / 6 + 4  6 / 10  0,6 |

Captura del processing y resultado:

|  |  |
| --- | --- |

d) (x ^ 2 + y ^ 2) ^ (1 / 2) x=5 y=10

| Aritmética | Algebraica |
| --- | --- |
| (x ^ 2 + y ^ 2) ^ (1 / 2)  (5 ^ 2 + 10 ^ 2) ^ (1/2)  (125 )^ (1/2)  11,180 | (25+100)^1/2  125^1/2  11,180 |

Captura del processing y resultado:

|  |  |
| --- | --- |

**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

| Aritmética | Algebraica |
| --- | --- |
| B \* A – B ^ 2 / 4 \* C  20 - 25/4  13.75 | 5 . 4 - 5² /4 .1  20 - 25/4  13.75 |

Captura del processing y resultado:

|  |  |
| --- | --- |

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

| Aritmética | Algebraica |
| --- | --- |
| (A \* B) / 3 ^ 2  4 \* 5 / 3 ^ 2  20/9  2,2... | (4 . 5) / 3²  20 / 9  2,2... |

Captura del processing y resultado:

|  |  |
| --- | --- |

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

| Aritmética | Algebraica |
| --- | --- |
| (((B + C) / 2 \* A + 10) \* 3 \* B) – 6  (((5 + 1) / 2\* 4 + 10 ) \* 3 \* 5 ) -6  (((6/2).4+10)\*3\*5) -6  ((22.3.5) -6  330 - 6  324 | (((5 + 1)/ 2 . 4 + 10) . 3 . 5) -6  (((6/2).4+10).3.5)-6  22 . 3 . 5 - 6  324 |

Captura del processing y resultado:

|  |  |
| --- | --- |

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

R1 = y+z

R2 = x >= R1

| SOLUCIÓN |
| --- |
| R1 = 4 + 1 = 5  R2 = 3>= 5  Falso |

Captura del processing y resultado:

|  |  |
| --- | --- |

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

R1 = ++contador1

R2 = contador1 < contador2

| SOLUCIÓN |
| --- |
| R2 = 4<4  R2= FALSO |

Captura del processing y resultado:

|  |  |
| --- | --- |

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

a+b-1 < x\*y

| SOLUCIÓN |
| --- |
| A+b-1 < x\*y  31 + (-1)-1 < 3 \* 2  29 < 12  falso |

Captura del processing y resultado:

|  |  |
| --- | --- |

**Ejercicio 9**: Para x=6, y=8, evaluar el resultado de

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

| SOLUCIÓN |
| --- |
| !(x<5)CC !(y>=7)  !(6<5)CC !(8>=7)  Falso CC falso  falso |

Captura del processing y resultado:

|  |  |
| --- | --- |

**Ejercicio 10**: Para i=22,j=3, evaluar el resultado de

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

| SOLUCION |
| --- |
| !((i>4) || !(j<=6))  !((22>4) || !(3<=6))  !(Verdadero || falso)  !(verdadero)  falso |

Captura del processing y resultado:

|  |  |
| --- | --- |

**Ejercicio 11**: Para a=34, b=12,c=8, evaluar el resultado de

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

| SOLUCION |
| --- |
| !(a+b==c) || (c!=0)CC(b-c>=19)  !(34+12==8) || (8!=0)CC(12-8>=19)  !(46==8) || (8!=0)CC(4>=19)  Verdadero || verdadero || falso  Verdadero || falso  verdadero |

Captura del processing y resultado:

|  |  |
| --- | --- |

**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.

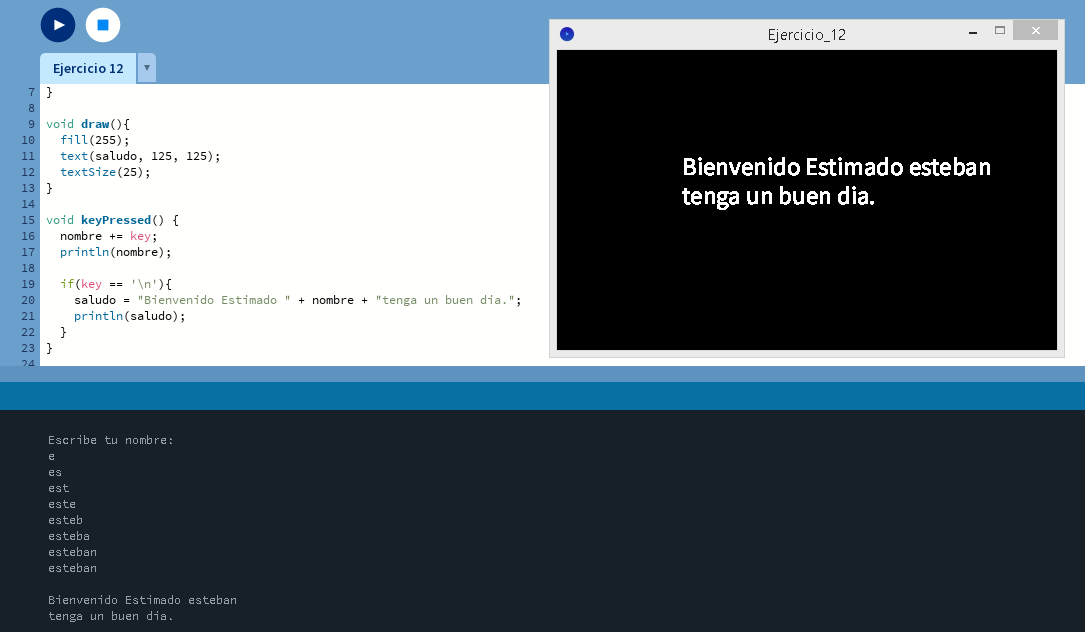
**Análisis:**

| Datos de Entrada: // cadena  Datos de Salida: //cadena de texto  Proceso: ¿quien debe realizar el proceso?:  La computadora  ¿Cual es el proceso que resuelve?  Ingresa caracteres que devolverá un saludo con los caracteres asignados  Y su vista en pantalla. |
| --- |

**Diseño:**

| Entidad que resuelve el problema: Algoritmo |
| --- |
| Variables:  Nombre:String //almacena nombre  Saludo: String// almacena la cadena de saludo |
| Nombre del Algoritmo: saludar\_nombre |
| Proceso del algoritmo:   1. Leer nombre 2. Saludo "Bienvenido Estimado " + nombre + "tenga un buen dia." 3. Mostrar saludo 4. fin |

Captura del processing y Resultado:

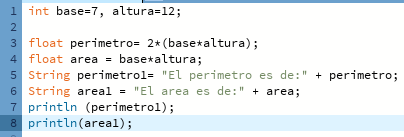
**Ejercicio 13**: Será común resolver problemas utilizando variables. Calcule el perímetro y área de un rectángulo dada su base y su altura.

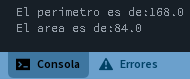
**Análisis:**

| Datos de Entrada: base , altura // decimal  Datos de Salida: área, perímetro// decimal  Proceso:  ¿Quien debe realizar el proceso?  El usuario o calculadora  ¿Cual es el proceso que resuelve?  Calcula el perímetro y el área del rectángulo utilizando una formula especifica |
| --- |

**Diseño:**

| Entidad que resuelve el problema: persona |
| --- |
| Variables:  - Base y Altura : int // almacena valores enteros  - Perimetro : float // almacena valores decimales  - Area : float // almacena valores decimales  - Perimetro1 : cadena //  - Area1 :cadena //  - Mostrar Resultados de Perimetro1 , Area1 : Println // mostrar consola |
| Nombre del Algoritmo: perimetro\_y\_area\_del\_rectangulo |
| Proceso del algoritmo:   1. Leer base y altura 2. Perimetro ← 2\* (base + altura) 3. Area ← base \* altura 4. Perimetro1 ← “El perimetro es de:" + perimetro 5. area1 ← "El area es de:" + area 6. Mostrar Perimetro1 7. Mostrar Area1 8. fin |

Captura del processing y Resultado:



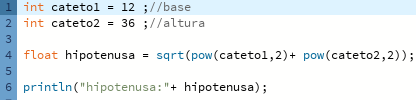
**Ejercicio 14**: Una ayuda importante al momento de resolver problemas con algoritmos es asumir que su gran amigo son las matemáticas. Obtenga la hipotenusa de un triángulo rectángulo conociendo sus catetos.

**Análisis:**

| Datos de Entrada: cateto1, cateto2  Datos de Salida: hipotenusa  Proceso:  ¿Quien debe realizar el proceso?  El usuario o calculadora  ¿Cual es el proceso que resuelve?  Para calcular la hipotenusa de un triangulo rectangulo se requiere los valores de ambos catetos que brinda el resultado y se utiliza el teorema de pitagoras h2 = a2 + b2 |
| --- |

**Diseño:**

| Entidad que resuelve el problema: persona |
| --- |
| Variables:  - Cateto1: int // almacena valores enteros  - Cateto2: int // almacena valores enteros  - Hipotenusa: float // almacena un valor de calculos |
| Nombre del Algoritmo: perimetro\_area\_rectangulo |
| Proceso del algoritmo   1. Leer cateto1 2. Leer cateto2 3. Hipotenusa ← (a^2 +b^2)^(0.5) 4. Mostrar hipotenusa 5. fin |

Captura del processing y Resultado:



**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.

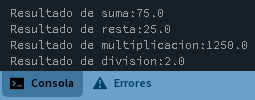
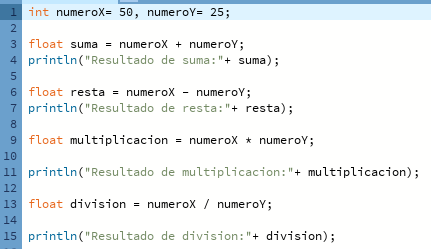
**Análisis:**

| Datos de Entrada: Numerox, NumeroY  Datos de Salida: suma , resta , multiplicación, división  Proceso:  ¿Quien debe realizar el proceso?  El usuario o calculadora  ¿Cual es el proceso que resuelve?  Resuelve problemas como suma , resta , division y multiplicacion a base de 2 numeros |
| --- |

**Diseño:**

| Entidad que resuelve el problema: persona |
| --- |
| Variables:   * Numerox: int // almacena valor entero * Numeroy: int // almacena valor entero * Suma: float // almacena valor de suma * Resta: float // almacena valor de resta * Multiplicacion: float // almacena un valor de multiplicación * División: float // almacena un valor de división |
| Nombre del Algoritmo : Calculador |
| Proceso del algoritmo   1. Leer numerox 2. Leer numeroy 3. Suma ← numeroX + numeroY 4. Mostrar ← "Resultado de suma:"+ suma 5. Resta ← numeroX - numeroY 6. Mostrar ← "Resultado de resta:"+ resta 7. Multiplicación ← numeroX \* numeroY 8. Mostrar ← "Resultado de multiplicación:"+ multiplicación 9. división ← numeroX / numeroY 10. Mostrar ← "Resultado de división:"+ division 11. fin |

Captura del processing y Resultado:



**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.

**Análisis:**

| Datos de Entrada:temperatura en grados Fahrenheit  Datos de Salida:temperatura grados Celsius  Proceso:  ¿Quien debe realizar el proceso?  Por una calculadora o programa matematico.  ¿Cual es el proceso que resuelve?  Resuelve el procedimiento de llevar una temperatura en grados fahrenheit a grados celsius. |
| --- |

**Diseño:**

| Entidad que resuelve el problema: persona |
| --- |
| Variables:  - TempFahrenheit :float // almacena valor decimal  - TempCelsius:float // almacena valor decimal |
| Nombre del Algoritmo: conversor\_de\_temperatura\_celsius |
| Proceso del algoritmo   1. Leer tempFahrenheit 2. TempCelsius ←(tempFahrenheit - 32)/1.8 3. Mostrar tempCelsius 4. fin |

Captura del processing y Resultado:



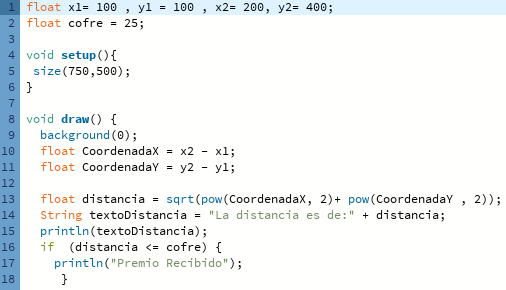
**Ejercicio 17**: Si queremos representar personajes o power ups (premios) en la pantalla debemos primero ubicarlos en alguna posición dentro de la pantalla. Imagine que está en un juego donde un power up desaparece porque el personaje se acerca a una distancia de x unidades, sin importar por donde se 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 (??1, ??1) , mientras que las de la caja de tesoro se halla en la posición (??2, ??2). Si observa con detenimiento se observa la conformación de un triángulo rectángulo, por lo que es posible aplicar Pitágoras para obtener la distanciaPara 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.

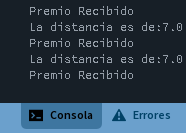
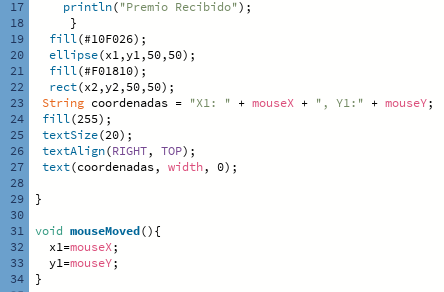
**Análisis:**

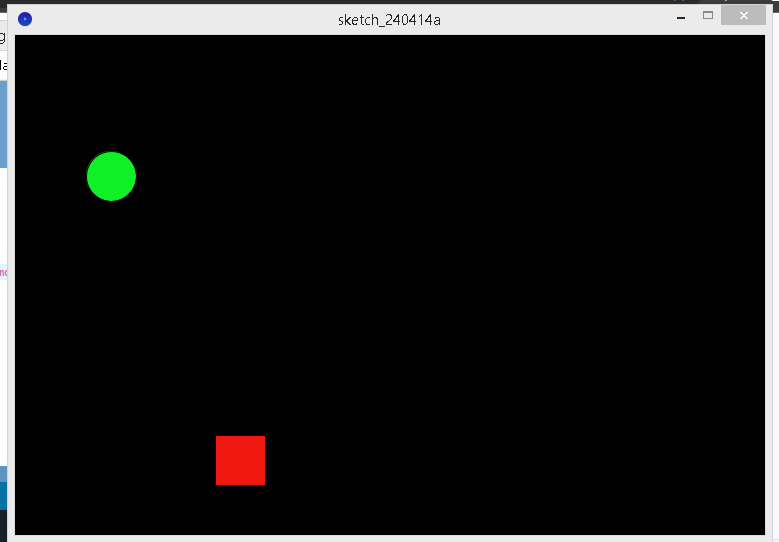
| Datos de Entrada: Coordenadas de link, Coordenadas cofre  Datos de Salida: Distancia entre Link y tesoro  Proceso:  ¿Quien debe realizar el proceso?  El programa informático o la computadora  ¿Cual es el proceso que resuelve?  Calculamos las diferencias en las coordenadas x;y entre los dos puntos |
| --- |

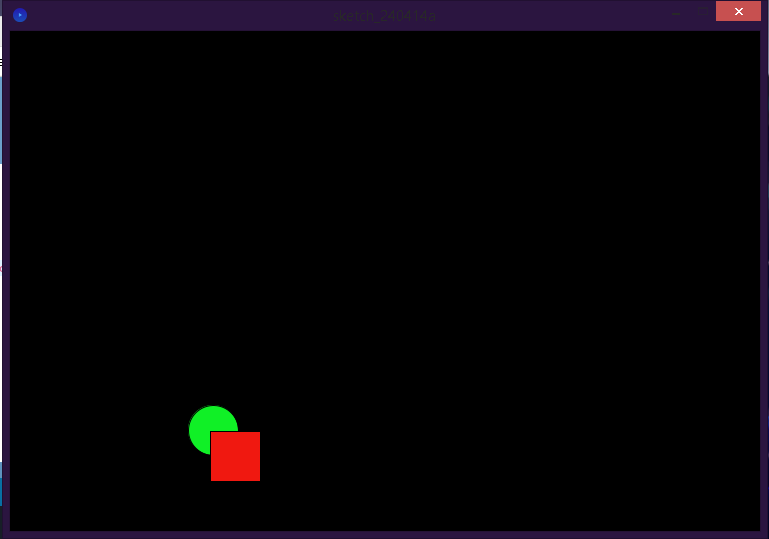
**Diseño:**

| Entidad que resuelve el problema: persona |
| --- |
| Variables:  - X1: float // almacena un valor decimal  - Y1 : float // almacena un valor decimal  - X2 : float // almacena un valor decimal  - Y2 : float // almacena un valor decimal  - CoordenadaX : float // almacena el resultado de un calculo  - CoordenadaY : float // almacena el resultado de un calculo  - Distancia : float // almacena el resultado de un calculo  - Cofre : float // almacena un valor |
| Nombre del Algoritmo: distancia |
| Proceso del algoritmo   1. Leer x1 2. Leer y1 3. Leer x2 4. Leer y2 5. Cofre ← 25 6. CoordenadaX ← x2 - x1 7. CoordenadaY ← y2 - y1 8. Distancia ← ((CoordenadaX)2+ (CoordenadaY )2)2 9. Mostrar "La distancia es de:" + distancia 10. Si distancia <= cofre 11. Mostrar distancia “Premio Recibido” 12. fin |

Captura del processing y Resultado:







**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.

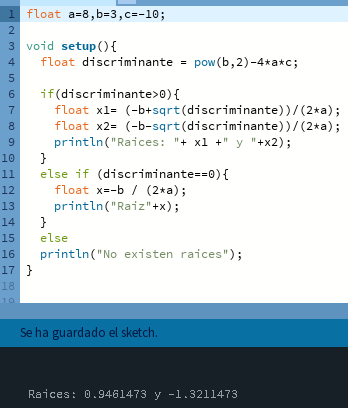
**Análisis:**

| 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?  El programa informático o la computadora  ¿Cual es el proceso que resuelve?  Calcular discriminante de la ecuacion cuadratica. |
| --- |

**Diseño:**

| Entidad que resuelve el problema: persona |
| --- |
| Variables:  - A: float // almacena un valor decimal  - B: float // almacena un valor decimal  - C: float // almacena un valor decimal  - Discriminante: float //almacena un valor de calculos |
| Nombre del Algoritmo encontrar\_raices |
| Proceso del algoritmo   1. Leer a 2. Leer b 3. Leer c 4. Discriminante ← b^2 - 4\*a\*c 5. Si (discriminante > 0) entonces 6. Raiz 1 ← (-b+(discriminante))^/(2\*a) 7. Raiz2 ← (-b-(discriminante))^/(2\*a) 8. Mostrar “las raices son” + raiz1+ y + raiz2 9. Si\_no si (discriminante == 0) entonces 10. Mostrar “la raiz doble es:” + raiz 11. si\_no 12. Mostrar “no hay raices reales” 13. fin |

Captura del processing y Resultado:



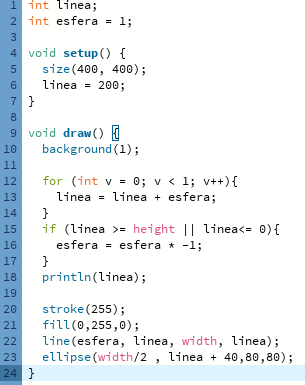
**Ejercicio 19**: Declare las variables necesarias para dibujar una línea que se dibuja desde las coordenadas iniciales del lienzo y se extiende por todo el ancho. Sobre el punto medio de la línea y a una distancia de 40px (en sentido vertical desde la línea) dibuje una elipse que tenga como ancho 80px 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.

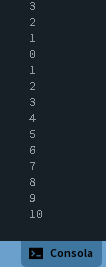
**Análisis:**

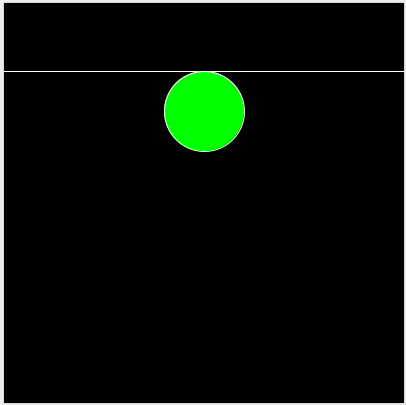
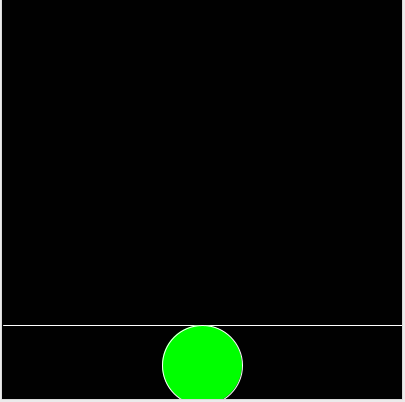
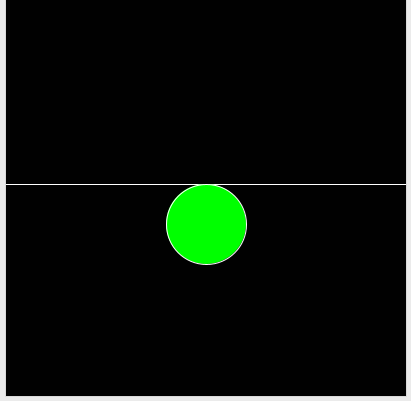
| Datos de Entrada: linea , esfera  Datos de Salida: bucle de la linea y circulo  Proceso:  ¿Quien debe realizar el proceso?  El programa informático o la computadora  ¿Cual es el proceso que resuelve?  Que la linea del circulo empieza desde 1 y llega hasta la maxima posicion otorgada, como no puede pasar, rebota y vuelve a su estado original |
| --- |

**Diseño:**

| Entidad que resuelve el problema: lienzo |
| --- |
| Variables:  - Linea: entero // almacena valor entero  - Esfera : entero // almacena valor entero |
| Nombre del Algoritmo: linea\_en\_movimiento |
| Proceso del algoritmo   1. Leer linea 2. Leer esfera 3. Ancholienzo ← 400 4. AltoLienzo ← 400 5. Para v ← 0 hasta alto incremento 1 6. Linea ← linea + esfera 7. Si ((linea >= ancholienzo || linea<= 0)) 8. Esfera ← esfera \* -1 9. fin\_si 10. Mostrar linea 11. Dibujar linea esfera, linea,altolienzo, linea 12. Dibujar altolienzo/2 , linea + 40,80,80 13. fin |

Captura del processing y Resultado:





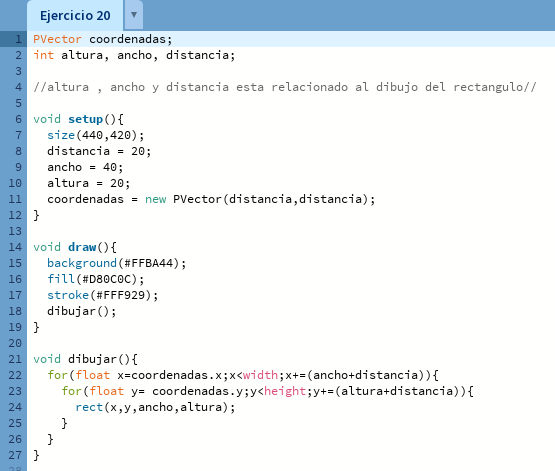
**Ejercicio 20**: Dibuje en toda la extensión del lienzo de (440, 420) rectángulos de idénticas medidas (40 ancho y 20 de alto) y que mantengan una distancia de 20 pixeles entre ellos tanto horizontal como verticalmente. Utilice la estructura de control repetitiva for. El lienzo debería verse así.

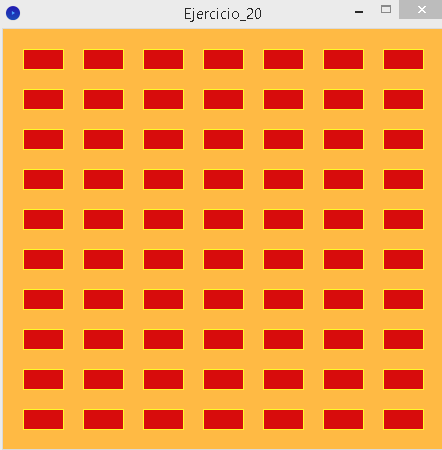
**Análisis:**Dibujar en toda la intencional de un lienzo de (440, 420) rectángulos de idéntica medida.

| Datos de Entrada: Rectángulos dibujados en el lienzo a base de las especificaciones dadas  Datos de Salida:Los Rectángulos dibujados en el lienzo a base de las especificaciones dadas  Proceso:  ¿Quien debe realizar el proceso?  :El proceso es realizado por un programa como processing  ¿Cual es el proceso que resuelve?  Dibujar varios rectángulos en el lienzo con un ta maño especifico respetando las distancias verticales y horizontales entre ellos, utilizando la variable **for**  para dibujar los rectángulos |
| --- |

**Diseño:**

| Entidad que resuelve el problema: Lienzo |
| --- |
| Variables:  **- coordenadas: float //** almacena un valor de coordenadas cartesianas  **- ancho y la altura del lienzo: size //** almacena valores enteros de la posición  **- ancho, alto y la distancia: int //** almacena valores enteros de la posición |
| Nombre del Algoritmo : repetición\_rectángulos |
| Proceso del algoritmo   1. Leer La Altura, ancho y la distancia // determinando valores enteros 2. Leer el tamaño asignado (x, y) (440, 420) 3. Leer distancia 4. Leer ancho 5. Leer Altura 6. Leer Coordenadas de PVvector 7. Para X coordenadas.x hasta ancho del lienzo (ancho + distancia) 8. Para Y coordenadas.y hasta la altura del lienzo (altura+distancia) 9. Dibujar rectángulo en (x,y,ancho,altura) 10. fin |

Captura del processing y Resultado:



**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.

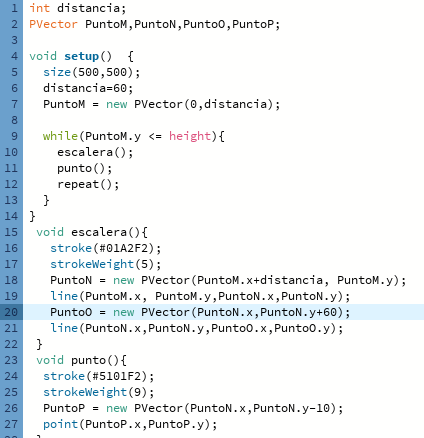
**Análisis:**

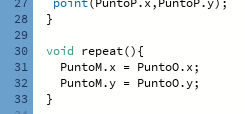
| Datos de Entrada: PuntoM,PuntoN,PuntoO,PuntoP,DISTANCIA  Datos de Salida: Una imagen que consiste en escalones con puntos en los bordes  Proceso:  ¿Quien debe realizar el proceso?  El programa informático o la computadora  ¿Cual es el proceso que resuelve?  El proceso que resuelve consiste en repetir while() para dibujar escalones y puntos. |
| --- |

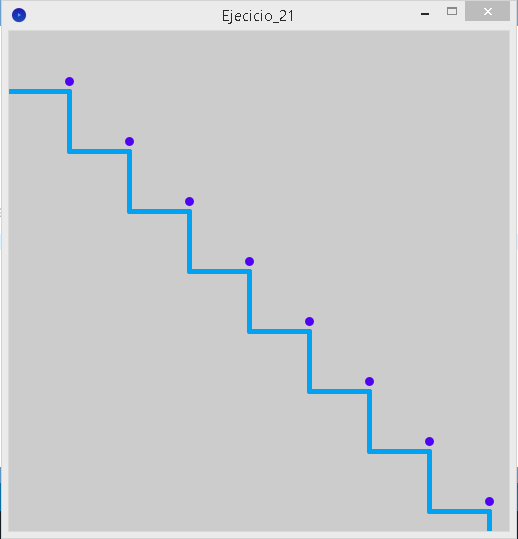
**Diseño:**

| Entidad que resuelve el problema: programa |
| --- |
| Variables:  PuntoM,PuntoN,PuntoO,PuntoP: int // almacena un vector  Distancia : int // almacena un valor entero |
| Nombre del Algoritmo: escalones\_puntos |
| Proceso del algoritmo:   1. Ancho lienzo ← 500 2. Alto lienzo ← 500 3. Distancia ← 60 4. Mientras PuntoM.y sea menor o igual ancholienzo 5. Dibujar linea horizontal en PuntoM.y,PuntoN.x,PuntoN.y 6. Dibujar linea vertical en PuntoN.x,PuntoN.y,PuntoO.x,PuntoO.y 7. Dibujar crculo en (PuntoP.x,PuntoP.y) 8. PuntoM.x ← PuntoO.x 9. PuntoM.y ← PuntoO.y 10. fin |

Captura del processing y Resultado:







**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.

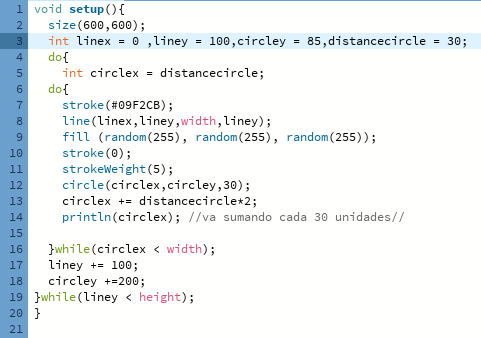
**Análisis:**

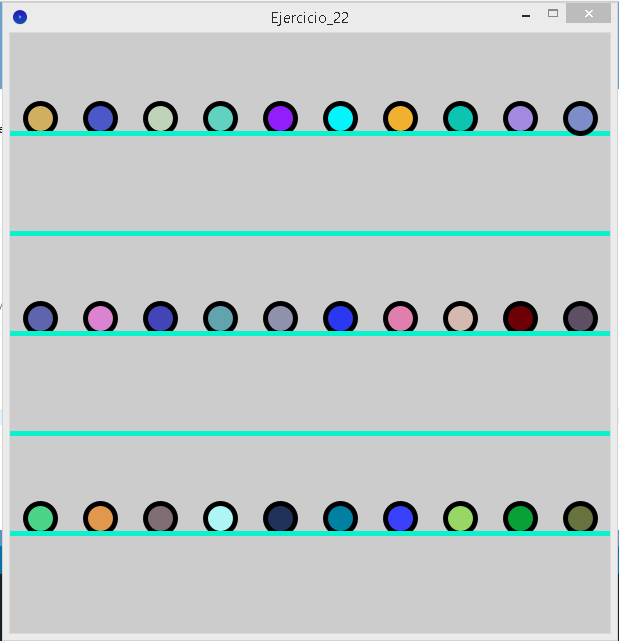
| Datos de Entrada: lineas y circulos  Datos de Salida: circulos de colores aleatorios sobre una linea.  Proceso:  ¿Quien debe realizar el proceso?  El programa en este caso el processing  ¿Cual es el proceso que resuelve?  El lienzo establecido ocupa una serie de lineas de forma vertical de misma medida. En cada linea se dibujan varios círculos con diversos colores a lo largo de la linea , respetando un distanciamiento. En las lineas se alternan de por medio que linea contendrá círculos y cual no, así sucesivamente |
| --- |

**Diseño:**

| Entidad que resuelve el problema: programa processing |
| --- |
| Variables:  -linex ,liney ,circley ,circlex,distancecircle :int // almacenan valores enteros  -Ancholienzo, alto lienzo : int // almacenan valores enteros |
| Nombre del Algoritmo: circulos\_repetidos |
| Proceso del algoritmo   1. Ancho lienzo ← 600 2. Alto lienzo ← 600 3. Linex ←0 4. Liney ← 100 5. Distancecircle ← 30 6. Circley ← 85 7. do 8. circlex← distancecircle 9. do 10. Dibujar line (linex,liney,ancholienzo,liney) 11. Dibujar circle(circlex,circley,30) 12. Circlex ← circlex + distancecircle\*2 13. Fin\_do 14. While(circlex < ancholienzo) 15. Liney ← liney + 100 16. Cricley ←circley +200 17. Fin\_do 18. While (liney < altolienzo) 19. fin |

Captura del processing y Resultado:







Fuentes bibliográficas

Se deben enunciar las fuentes (apuntes de la materia, páginas web, videos de youtube, libro

**- Tutorial de processing: matemáticas**

([PROCESSING: 4.4 Matemáticas I [En español] - YouTube](https://www.youtube.com/watch?v=LUfoF-tG8xw&t=585s))

**- Tutorial de proccesing : font text**

(https://processing.org/reference/textFont\_.html)

**-Colores RGB con codigos**

https://www.rapidtables.org/web/color/RGB\_Color.html

**-Calculadora**

[https://mathsolver.microsoft.com/es/solveproblem/3%20%3C%20%206?r=bi&ref=bi](https://mathsolver.microsoft.com/es/solveproblem/3%20<%20%206?r=bi&ref=bi)

-**Perimetro y area:**

<https://content.nroc.org/DevelopmentalMath/U07L2T2_RESOURCE/U07_L2_T2_text_final_es.html>

**-Perimetro y area de un rectangulo:**

<https://www.neurochispas.com/wiki/perimetro-y-area-de-un-rectangulo-formulas-y-ejercicios/>

**-Sacar hipotenusa :**

<https://www.unprofesor.com/matematicas/como-sacar-la-hipotenusa-de-un-triangulo-6083>.

**-Aplicar Raiz cuadrada en processing:**

<https://processing.org/reference/quad_.html>