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

Tolaba Brian Ezequiel – LU TUV000639

*Profesores:*

*Mg. Ing. Ariel Alejandro Vega*

*Año 2024*

**FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS**

**EJERCICIO 1**

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

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

**Resolución necesaria en Word:**

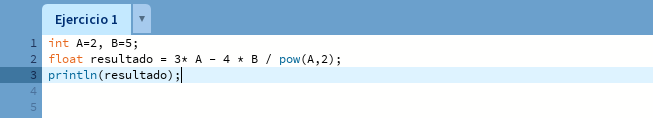
(3\*2) - (4\*5/(2^2))

6-(4\*B/4)

6-5

1

**Captura de Processing**



**EJERCICIO 2**

Evaluar la siguiente expresión 4 / 2 \* 3 / 6 + 6 / 2 / 1 / 5 ^ 2 / 4 \* 2

**Resolución necesaria en Word:**

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

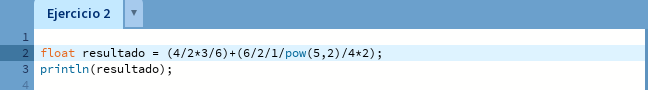
(4/2\*3/6) + (6/2/1/25/4\*2)

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

1 + 0.06

1.06

**Captura de Processing**



**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 \* a \* c

*c*

1. 3 \* X ^ 4 – 5 \* X ^ 3 + X \* 12 – 17
2. (b + d) / (c + 4)
3. (x ^ 2 + y ^ 2) ^ (1 / 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)

**Resolución necesaria en Word**

a=5, b=3, c=4, d=1, x=2, y=6

1. 3^2 – 4\*5\*4

9 – 80

71

1. 3\*2^4 - 5\*2^3+2\*12 – 17

48 – 40 + 24 – 17

15

1. (3+1)/(4+4)

4/8

0.5

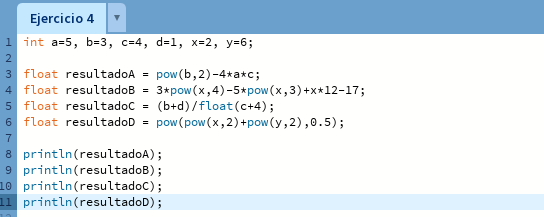
1. (2^2+6^2) ^ (1/2)

(4+36) ^ (1/2)

40^ (1/2)

6.324

**Captura de Processing**



**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

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

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

**Resolución en Word**

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

20 - (25/4\*1)

20 - 6.25

13.75

1. 4\*5/ (3^2)

20/9

2.222

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

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

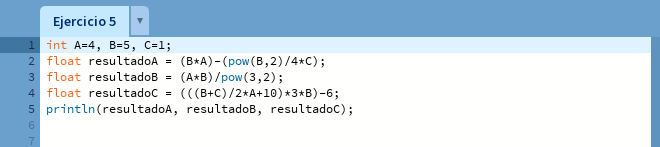
((12+10) \*3\*5) -6

(22\*3\*5) -6

330-6

324

**Captura de Processing**



**EJERCICIO 6**

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

R1 = y + z

R2 = x >= R1

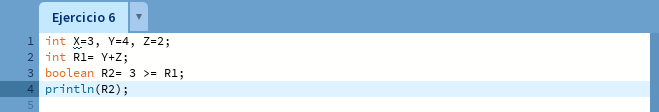
**Resolución en Word**

R1= 4+1 = 5

R2= 3>= R1

false

**Captura de Processing**



**EJERCICIO 7**

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

R1 = ++contador1

R2 = contador1 < contador2

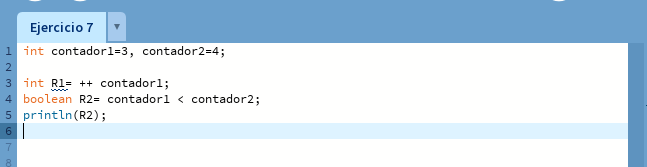
**Resolución en Word**

R1 = 1+3

R2 = 4 < 4

false

**Captura de Processing**



**EJERCICIO 8**

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

a+b-1 < x\*y

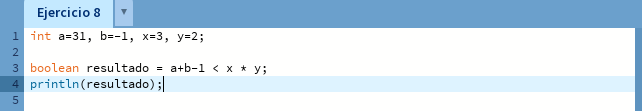
**Resolución en Word**

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

29 < 6

false

**Captura de Processing**



**EJERCICIO 9**

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

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

**Resolución en Word**

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

False && false

false

**Captura de Processing**



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

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

**Resolución en Word**

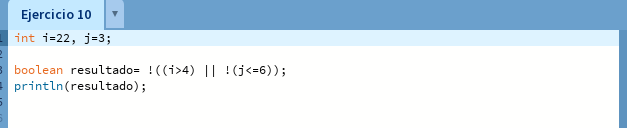
!((22>4) || !(3<=6))

!(true|| false)

!(true)

false

**Captura de Processing**



**EJERCICIO 11**

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

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

**Resolución en Word**

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

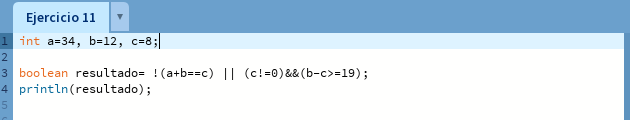
!(46==8) || (8!=0)&&(4>=19)

true || true && false

true|| false

true

**Captura 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

**Fase de Análisis:**

DATOS DE ENTRADA nombre ingresado

DATOS DE SALIDA saludo con el nombre

PROCESO

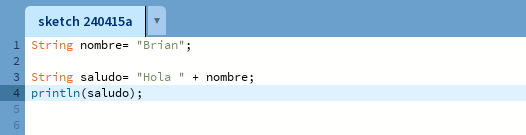
¿Quién debe realizar el proceso?: computadora

¿Cuál es el proceso que resuelve?: Ingresar un nombre que devolverá un saludo con el nombre en la pantalla.

**Fase de Diseño:**

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL PROBLEMA: computadora** |
| **VARIABLES:**  **nombre: string**  **saludo: string** |
| **NOMBRES DEL ALGORITMO: Saludo**  **PROCESO DEL ALGORITMO:**  1. inicio  2. Leer nombre  3. saludo ← “Hola, ” + nombre  4. Mostrar saludo  5. fin |

**Captura de processing:**



**EJERCICIO 13**

**Fase de Análisis:**

DATOS DE ENTRADA: base y altura de un rectángulo

DATOS DE SALIDA perímetro y área del rectángulo

PROCESO

¿Quién debe realizar el proceso?: calculadora

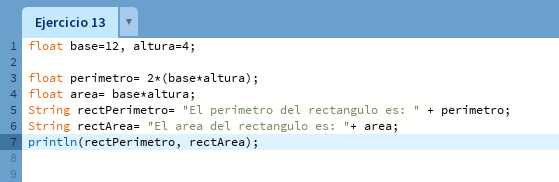
¿Cuál es el proceso que resuelve?: calcula el perímetro y el área de un rectángulo utilizando las fórmulas para el mismo

P=2(base + altura) y  A=base \* altura

**Fase de Diseño:**

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL PROBLEMA: Calculadora** |
| **VARIABLES:** **Variables:**  **- base: float**  **- area: float**  **- rectPerimetro: float**  **- rectArea: float** |
| **NOMBRES DEL ALGORITMO: Perimetro area rectangulo**  **PROCESO DEL ALGORITMO:**   1. *inicio* 2. *Leer base* 3. *Leer area* 4. *perimetro ← 2\*(base + altura)* 5. *area ← base \* altura* 6. *rectPerimetro ← "El perimetro del rectangulo es: " + perímetro* 7. *rectArea← “"El area del rectangulo es: "+ area* 8. *Mostrar rectPerimetro y rectArea* 9. fin |

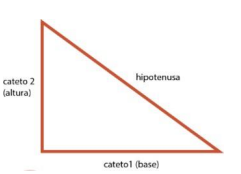
**Captura de processing:**

****

**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 Análisis:**

DATOS DE ENTRADA cateto1, cateto2

DATOS DE SALIDA hipotenusa

PROCESO ¿Quien debe realizar el proceso?: calculadora

¿Cual es el proceso que resuelve?: Para calcular la longitud de la hipotenusa de un triángulo rectángulo se aplica la fórmula: h^2=a^2+b^2 h=√(a^2+b^2 )

**Fase de Diseño:**

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL PROBLEMA: Calculadora** |
| **VARIABLES:**  **- cateto1: int**  **- cateto2: int**  **- hipotenusa: int** |
| **NOMBRES DEL ALGORITMO: hipotenusa triangulo rectangulo**  **PROCESO DEL ALGORITMO:**  **1. Inicio**  **2. Leer cateto1**  **3. Leer cateto2**  **4. hipotenusa ← (cateto1^2 + catero2^2 ) ^(0.5)**  **5. mostrar hipotenusa**  **6. Fin** |

**Captura de processing:**



**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 Análisis:**

DATOS DE ENTRADA numeroA, numeroB

DATOS DE SALIDA suma, resta, multiplicación, división

PROCESO

¿Quién debe realizar el proceso?: calculadora

¿Cuál es el proceso que resuelve?:

numeroA + numeroB

numeroA - numeroB

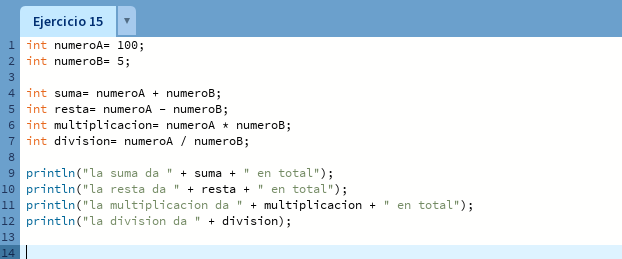
numeroA \* numeroB

numeroA / numeroB

**Fase de Diseño:**

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL PROBLEMA: Calculadora** |
| **VARIABLES:**  -numeroA: int  -numeroB: int  -suma: int  -resta: int  -multiplicación: int  -división: int |
| **NOMBRES DEL ALGORITMO: Calculos**  **PROCESO DEL ALGORITMO:**  1. inicio  2. Leer numeroA  3. Leer numeroB  4. suma ← numeroA + numeroB  6. resta ← numeroA – numeroB  8. multiplicacion ← numeroA \* numeroB  10. division ← numeroA / numeroB  11. mostrar ← "la suma da " + suma + " en total"  12. mostrar ← "la resta da " + resta + " en total"  13. mostrar ← "la multiplicacion da " + multiplicacion + " en total"  14. mostrar ← "la division da " + division  15. Fin |

**Captura de processing:**

****

**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 Análisis:**

DATOS DE ENTRADA grados en fahrenheit

DATOS DE SALIDA grados en celsius

PROCESO

¿Quién debe realizar el proceso?: calculadora o programa

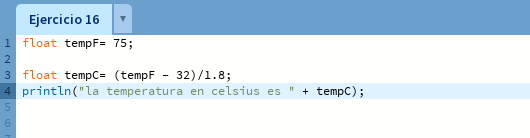
¿Cuál es el proceso que resuelve?: convertir una temperatura dada en grados Fahrenheit a grados Celsius utilizando la fórmula de conversión correspondiente

c=(F-32)/1.8

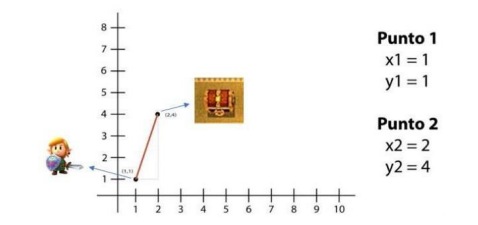
**Fase de Diseño:**

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL PROBLEMA: Programa** |
| **VARIABLES:**   * tempF: float * tempC: float |
| **NOMBRES DEL ALGORITMO: Conversor a Celsius**  **PROCESO DEL ALGORITMO:**  1. inicio  2. Leer tempF  3. tempC ← (tempF – 32) / 1.8)  4. mostrar tempC  5. fin |

**Captura de processing:**



**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 Análisis:**

DATOS DE ENTRADA Coordenadas de Link y coordenadas del tesoro

DATOS DE SALIDA Distancia entre Link y tesoro

PROCESO

**Fase de Diseño:**

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL PROBLEMA: calculadora** |
| **VARIABLES:** |
| **NOMBRES DEL ALGORITMO: Distancia link tesoro**  **PROCESO DEL ALGORITMO:** |

**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 Análisis:**

DATOS DE ENTRADA Coeficientes de la ecuación cuadrática: A, B y C

DATOS DE SALIDA Raíces de la ecuación cuadrática.

PROCESO

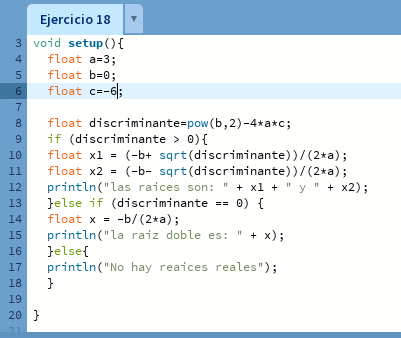
¿Quién debe realizar el proceso?: El programa informático o una calculadora que pueda realizar cálculos matemáticos.

¿Cuál es el proceso que resuelve?: Calcular el discriminante de la ecuación cuadrática utilizando la fórmula

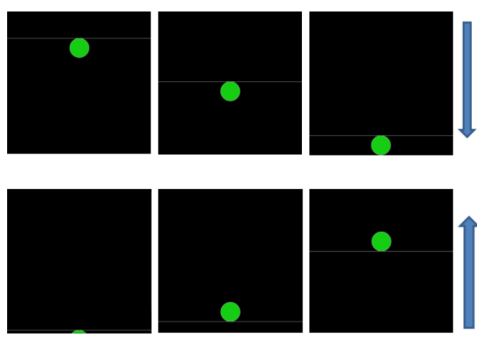
**Fase de Diseño:**

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL PROBLEMA: Calculadora** |
| **VARIABLES:**  *- a : float*  *- b : float*  *- c : float*  *- discriminante: float* |
| **NOMBRES DEL ALGORITMO: 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”   fin |

**Captura de processing:**



**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 Análisis:**

DATOS DE ENTRADA tamaño de la elipsis y direcciones

DATOS DE SALIDA movimiento de la elipse

PROCESO

¿Quien debe realizar el proceso?: computadora

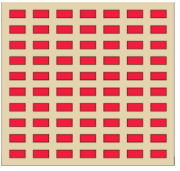
¿Cual es el proceso que resuelve?: según los parámetros dibujar una elipsis verde sobre una línea que sube y baja por la pantalla

**Fase de Diseño:**

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL PROBLEMA: computadora** |
| **VARIABLES:** |
| **NOMBRES DEL ALGORITMO: movimiento de elipsis y linea**  **PROCESO DEL ALGORITMO:**   1. inicio 2. leer Radio 3. Dibujar fondo, línea y elipsis 4. Incrementar el valor y 5. repetir |

**Captura de processing:**



**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 Análisis:**

DATOS DE ENTRADA Dimensiones de los rectángulos y el lienzo

DATOS DE SALIDA Rectángulos dibujados en el lienzo

PROCESO

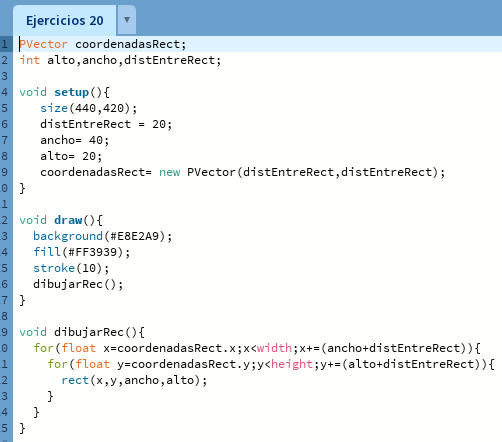
¿Quien debe realizar el proceso?: Programa (processing)

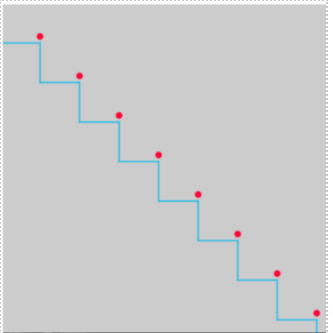
¿Cual es el proceso que resuelve?: dibujar una serie de rectángulos en un lienzo de tamaño específico, manteniendo una distancia específica entre ellos tanto horizontal como verticalmente, definiendo un bucle for para dibujar los rectángulos en el lienzo.

**Fase de Diseño:**

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL PROBLEMA: processing** |
| **VARIABLES:**   * coordenadasRect: float * ancho, alto, distanciaEntreRect : int * anchoLienzo, altoLienzo: int |
| **NOMBRES DEL ALGORITMO: ladrillos**  **PROCESO DEL ALGORITMO:**   1. *inicio* 2. *anchoLienzo ← 440* 3. *altoLienzo ← 420* 4. *ancho ← 40* 5. *alto ← 20* 6. *distanciaRect ← 20* 7. *para x ← coordenadasRect.x hasta anchoLienzo con paso (ancho+distEntreRect)* 8. *hacer* 9. *para y ← coordenadasRect.y hasta altoLienzo con paso (alto+distEntreRect)* 10. *hacer* 11. *dibujar rectangulo en (x,y,ancho,alto)* 12. *fin\_para* 13. *fin\_para* 14. *Fin* |

**Captura de processing:**

****

**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 Análisis:**

DATOS DE ENTRADA parámetros e imagen de ejemplo

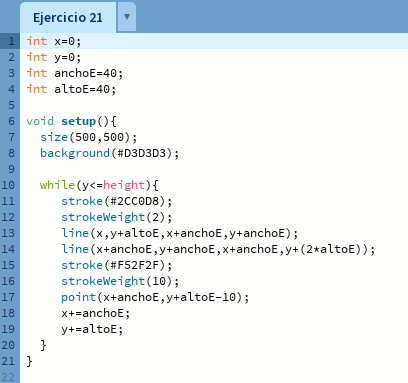
DATOS DE SALIDA Imagen con escalones y puntos rojos

PROCESO

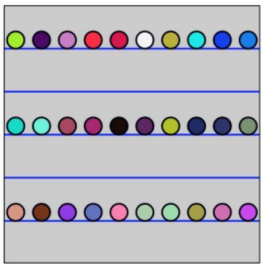
**Fase de Diseño:**

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL PROBLEMA: Processing** |
| **VARIABLES:**   * x, y, anchoE, altoE: int |
| **NOMBRES DEL ALGORITMO: Escalones y puntos**  **PROCESO DEL ALGORITMO:**   1. inicio 2. dibujar lienzo 3. mientras y sea menor que la altura dibujar escalón y punto 4. sumar origen del escalón para originar el siguiente 5. repetir |

**Captura de processing:**



**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 Análisis:**

DATOS DE ENTRADA parámetros del lienzo e imagen de ejemplo

DATOS DE SALIDA Lienzo interlineado con círculos con ajuste de color ramdom

PROCESO

¿Quien debe realizar el proceso?: Programa (processing)

¿Cual es el proceso que resuelve?: El lienzo se divide verticalmente en franjas de igual medida, donde se dibujan líneas en todas ellas. En cada línea de forma alternada, se dibujan círculos con colores aleatorios, los cuales están espaciados uniformemente a lo largo de la línea.

Diseño:

**Fase de Diseño:**

|  |
| --- |
| **ENTIDAD QUE RESUELVE EL PROBLEMA: processing** |
| **VARIABLES:**   * x, y, vacio, eY: int |
| **NOMBRES DEL ALGORITMO: líneas y circulos**  **PROCESO DEL ALGORITMO:**   1. *inicio* 2. *dibujar lienzo* 3. *dibujar líneas* 4. *cada línea y media dibujar círculos con configuración de color ramdom* 5. *repetir mientas no supere la altura del lienzo* |

**Captura de processing:**

