**Análisis y Diseño**

**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. El lienzo debería**

**verse así:**

**FASE DE ANÁLISIS**

**1) Definición del Problema:** dibujar en la extensión de un lienzo de medidas

(440,220) rectángulos que cubren su superficie y guarden sus respectivas e

idénticas distancias de 20 pixeles uno de otro.

**2) Análisis**

**Datos de entrada:**

1-Establecer la variables: int

altoRectangulo,anchoRectangulo,distanciaER;

Establecer Pvector posicionRectangulo = para inicializar el punto

de inició de la línea de dibujo tanto en eje x como en y.

1-Configurar el lienzo:Dimensiones del lienzo específicas al

tamaño del lienzo con size(440,220).

2-Dimensiones del rectángulo: Definidas por las variables

“altoRectangulo” = 20 y “anchoRectangulo” = 40 que determinan

el alto y ancho de cada rectángulo dibujado.

3-Distancia entre Rectángulos: La variable “distanciaER” =20

define la separación entre los rectángulos tanto del eje X como en Y.

4-color de fondo y de relleno: Aunque los colores estarán fijados

dentro del código (“background”) (#FFF5FA)) para el fondo y “fill”

(255,0,0), para los rectángulos considerare como datos de entrada si

quiero modificar el programa para que acepte colores dinámicos.

**Datos de salida:**

1-Visualizacion de rectángulos en el lienzo: El resultado directo es la

representación gráfica de los rectángulos dispuestos en el lienzo,

según las especificaciones de las entradas. No tengo un dato de

salida en el sentido tradicional del procesamiento de datos ya que

no tengo valores numéricos sino dibujos en este caso según los

parámetros de entrada.

**FASE DE DISEÑO**

| **ENTIDAD:** DIBUJAR RECTÁNGULOS EN UN LIENZO |
| --- |
| **Variables**  altoRectangulo:20  anchoREctangulo:40  distanciaER:20  tamaño lienzo(440,220) |
| **NOMBRE DEL ALGORITMO:** dibujar\_rectangulosenlienzo  Inicio  Establecer un bucle for para anidados:  Dibujar rectángulo:  Cambiar de posición Rectángulo  Rellenar con color lienzo y rectángulo  **Fin** |
| //Configurar el lienzo  Int altoRectangulo,anchoRectangulo,distanciaER;  PVector posicionRectangulo;  Establecer tamaño del lienzo: size(440,220);  altoRectangulo=20  anchoREctangulo=40  distanciaER=20  posicionRectangulo = new PVector(distanciaER,distanciaER);  background();  bucle for para anidados para iterar //para trabajar con figuras bidimensionales  // Este bucle se seguirá ejecutando mientras j sea menor que 10, dado que j comienza en 1  Y el bucle se repite hasta que j es menor que 10 , el bucle interno iterara 9 veces en total.  Cada iteración representa una fila de rectángulos. Y j++ incrementara el valor en 1despues  de cada iteración del bucle.  for(int j(valor)=1;j(valor)&lt;(menor)10;j(valor)++(se le añade 1)){  //Este bucle controla los rectángulos dentro de cada fila  Int i=1 inicializa el contador de rectángulos, I a 1 el comienzo de cada nueva fila.  I&lt;=7 es la condición de continuación del bucle, y se ejecutara mientras i sea menor o igual a  7 . Así que solo se dibujaran 7 rectangulos en cada fila. Y i++ incrementa el valor de i en 1  después de cada iteración del bucle lo que hará que se mueva el proceso al siguiente  rectangulo de la fila.  for(int i(valor)=1(valor);i(valor)&lt; (menor)=7;i++(se le añade)){  for(int j=1;j&lt;10;j++){  for(int i=1;i&lt;=7;i++){  //Posiciones del rectángulo en eje X y en eje Y  posicionRectangulo.x=distanciaER;  posicionRectangulo.y=posicionRectangulo.y + altoRectangulo + distanciaER;  // dibujar el rectangulo  Public void  dibujarRectangulo(){  fill(color de relleno) a elegir  // ejecutar la posición en el lienzo y con los argumentos posicionRectangulo y  anchoRectangulo especificamos la posición del vértice superior izquierdo del rectángulo en  el lienzo.  Rect utilizara los cuatro argumentos para dibujar en el lienzo los rectángulos según las  variables especificadas.  rect(posicionRectangulo.x,posicionRectangulo.y,anchoRectangulo,altoRectangulo);  //Cambiar posición Rectángulo:  cambiarposicionxRectangulo(){  Esta función se encargará de mover la posición del rectangulo hacia la derecha en el lienzo  de manera que se mantenga una separación constante entre los rectángulos en el eje  horizontal.  **Fin** |

