

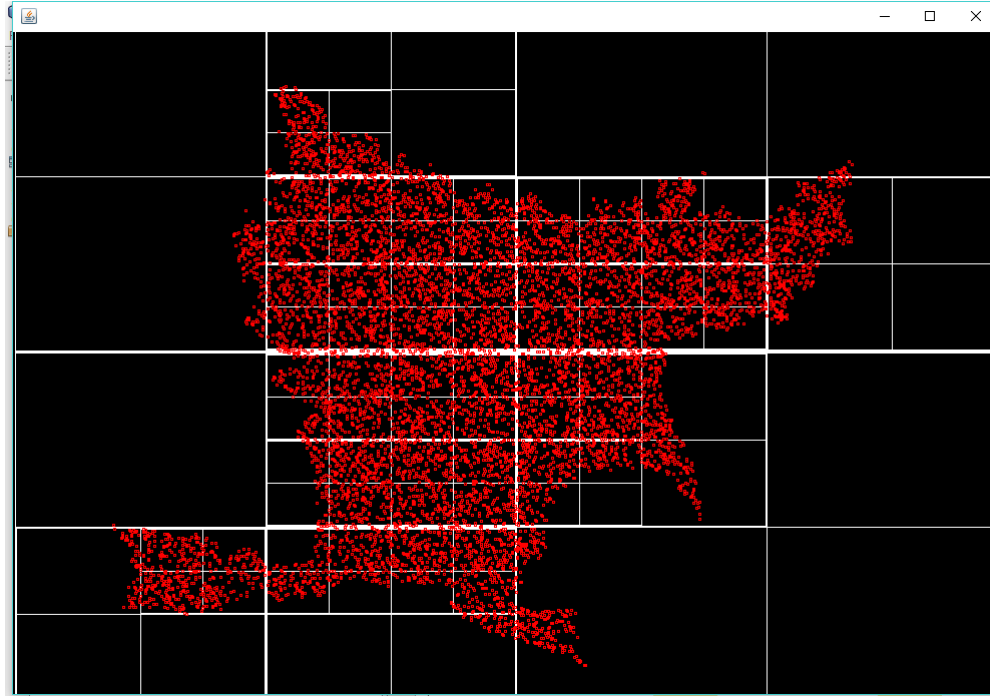
# ***Solución problema de collision de abejas con Quadtree***

***Isaias Labrador Sánchez***

***Manuel García Jimenez***

***Medellin, 25 de mayo, 2018***

# *Estructuras de Datos Diseñada*



**Gráfico 1:** Espacio visto como QuadTree donde cada cuadrado es un nodo del arbol, con su respectiva “capacidad”. (10,000 abejas).

# Operaciones de la Estructura de Datos

Método	Complejidad
Cuadrado	$O(1)$
QuadTree	$O(1)$
Insert(Abeja)	$O(\log(n))$
Leer(String Ruta)	$O(n)$
Circulo(x,y,radio)	$O(1)$
Abeja(x,y,latitud,longitud)	$O(1)$
Query(radio, ab_encont...)	$O(\log(n))$
Convertir_x(Double x)	$O(1)$
Convertir_y(Double y)	$O(1)$

**Tabla 1:** Complejidad de las operaciones de la estructura de datos

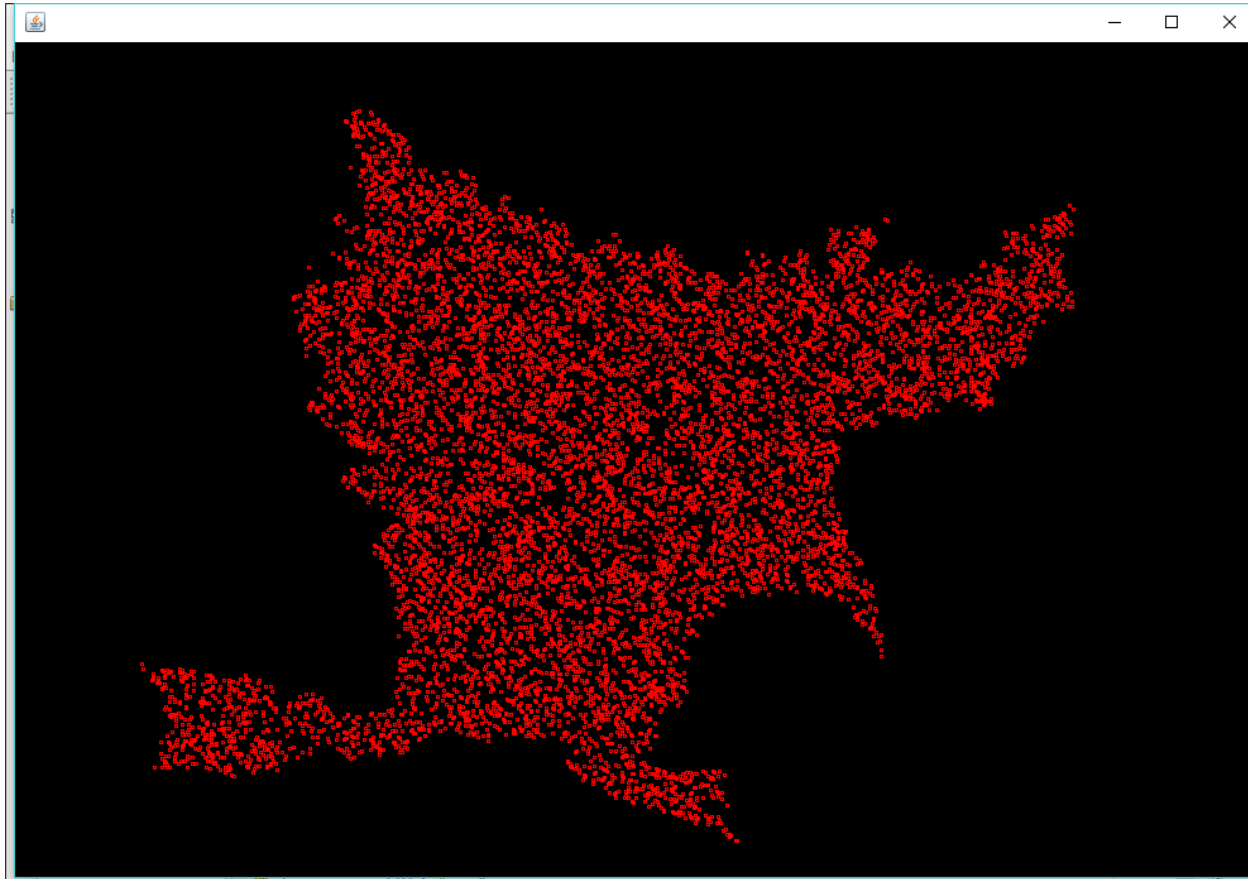
# ***Criterios de Diseño de la Estructura de Datos***

- Permite ver los objetos puestos dentro de ella como puntos con coordenadas bidimensionales, lo cual hace mas apropiado para este tipo de problemas que un árbol binario.
- Al posicionar los objetos teniendo en cuenta su posición, objetos cercanos geométricamente quedan cerca dentro del árbol, lo cual hace que al buscar objetos cercanos se puedan ignorar los que están lejos, lo que hace que esta acción sea  $O(\log(n))$ . Esto es mucho mejor que comprar todos con todos ( $O(n^2)$ ).

# *Consumo de Tiempo y Memoria*

Abejas	10	100	1000	10000
Memoria	3,1 Mb	3,27 Mb	3,5 Mb	4,1 Mb
Tiempo	10 ms	23 ms	75 ms	886 ms

# Software Desarrollado



**Gráfico 4:** Grafica de las abejas(10.000) en Bello, Antioquia.