

BEEBOTS, ESTRUCTURA DE DATOS PARA PREDECIR COLISIONES ENTRE DOS O MÁS OBJETOS

Andres Almanzar Restrepo

Santiago Hincapié Murillo

Santiago Valencia Arango

Medellín, Mayo 21 d 2018

Estructuras de Datos Diseñada

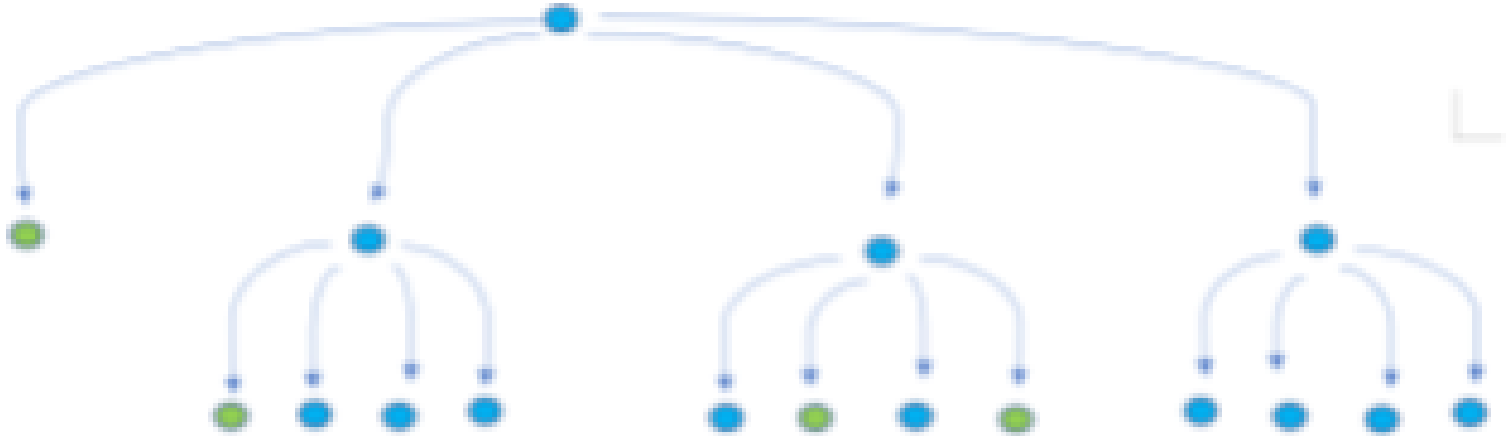
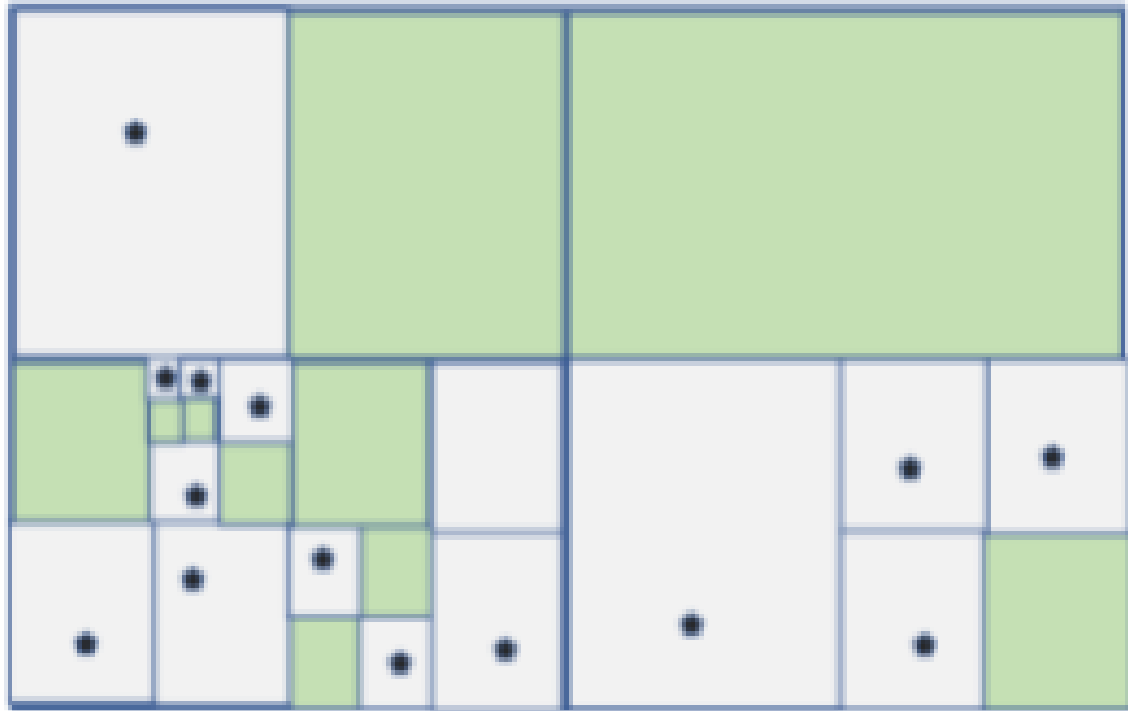


Gráfico 1: Se observa un grafo con nodos. Al principio todos los nodos salen de un nodo padre o raíz.

Gráfico 2: inicia desde un rectángulo padre. Aquellos cuadrantes que se pueden dividir en más nodos son de color azul y aquellos que no lo pueden hacer son verdes. Cada punto que se observa en los distintos cuadrantes, son los objetos, en nuestro proyecto son las abejas robóticas.



Operaciones de la Estructura de Datos

Abeja(int x, int y, int val)

Nombre de Abeja: abeja1

Coord. Abeja (21 123 1)

Abeja(int x, int y, int val)

Nombre de Abeja: abeja2

Coord. Abeja (231 43 2)

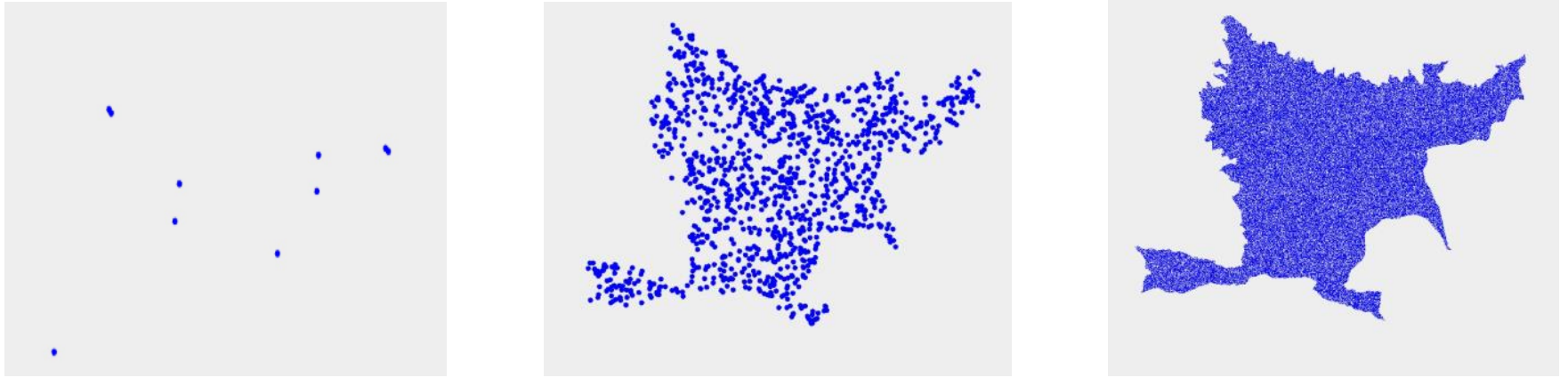
	X	Y	VAL
Abeja1	21	123	1
Abeja2	231	43	2
Abeja3

	X	Y	VAL
Abeja1	21	123	1
Abeja2	231	43	2
Abeja3



Gráfico 3: Metodo abeja, crea una abeja en coordenadas espaciales

Grafico 4: Metodo agregar abeja, convierte las coordenadas espaciales a posiciones en el plano



Gráfica 5: cantidad de abejas en el mapa de Bello

Abeja: crea una abeja en coordenadas espaciales

AgregarAbeja: lee las cordenas espaciales y las convierte a posiciones en el plano

Mostrar: dibuja las abejas en el plano

Principal: llama a agregar abejas y luego a mostrar

Quadtree: tiene los métodos insertar, dividir, distancia entre abejas, colisionan (booleano)

Criterios de Diseño de la Estructura de Datos

La elección del QuadTree para el desarrollo del proyecto radica en diversos factores:

Representación de imágenes gracias a la estructura que posee.

Detección eficiente de la colisión entre objetos en un campo 2D(dos dimensiones)

Al compararlo con otras estructuras de datos, posee mejor organización del espacio de los objetos. (esto debido a su constante división en cuadrantes).

-Se encarga de descomponer de manera recursiva el espacio.

Complejidad

Método	Complejidad
insert	$O(1)$
colision	$O(1)$
chocan	$O(n^2)$
agregarAbejas	$O(n)$
<u>ComprobarColisiones</u>	$O(\log(n))$

Tabla 1: análisis de complejidad

Consumo de Tiempo

Tiempo	N = 10	N = 100	N = 1000	N = 10000	N = 10000 0	N = 200000
Insertar	3m s	<u>10</u> <u>ms</u>	47ms	130ms	960ms	1573ms
Choque	0M s	3m s	45ms	250ms	3300m s	11253ms
Comproba r colisiones	0m s	24 ms	41ms	300ms	1500m s	6325 ms

Tabla 2: Tiempos de ejecución de las operaciones de la estructura de datos con diferentes conjuntos de datos

Consumo de Memoria

Memoria	10	100	1000	10000	100000	200000
Insertar	1Mb	1Mb	2Mb	8Mb	16Mb	23Mb
Choque	1Mb	1Mb	2Mb	8Mb	16Mb	20Mb
Comprobar colisiones	1Mb	1Mb	2Mb	9Mb	20Mb	30Mb

Tabla 3: Consumo de memoria de la estructura de datos con diferentes conjuntos de datos

Software Desarrollado

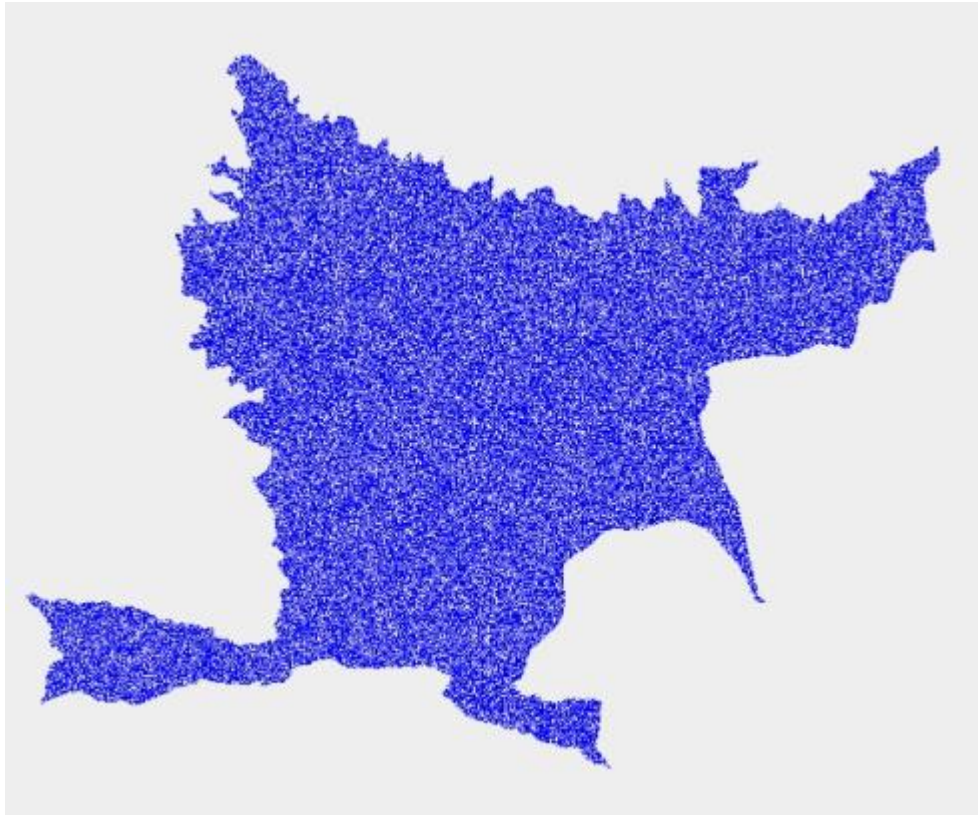


Gráfico 6: abejas en el mapa de bello