

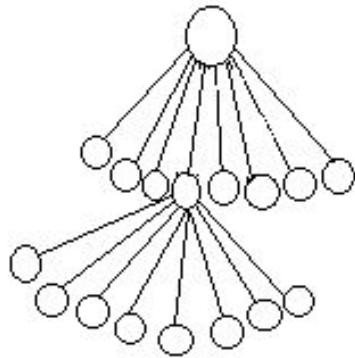
# ***Localización de abejas robóticas en 3D***

***Pablo Alberto Osorio Marulanda***

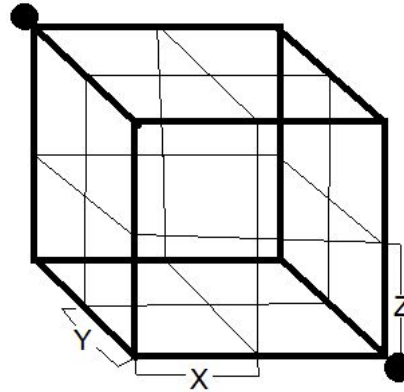
***Verónica Mendoza Iguarán***

***Medellín, 05/11/2018***

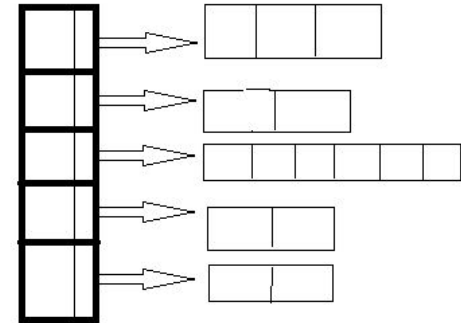
# Estructuras de Datos Diseñada



**Gráfico 1.** Representación de la división del espacio para la búsqueda de posibles abejas a colisionar.



**Gráfico 2.** Forma en que se toman las coordenadas, y representación mediante puntos negros de la posición mínima y máxima que se toma del espacio.

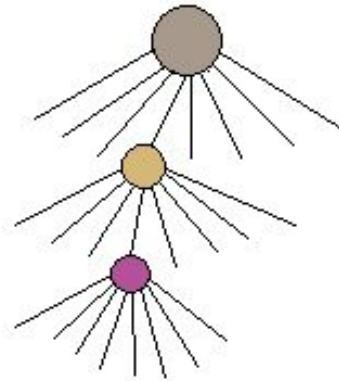
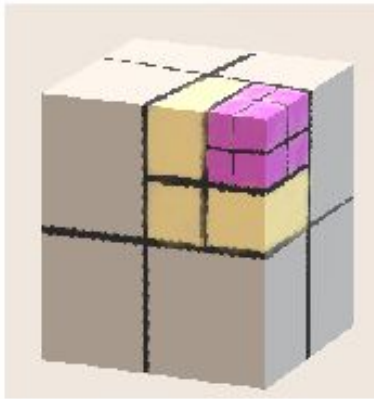


**Gráfico 3.** Representación de la linkedList que contiene linkedLists con las abejas.

Se divide el espacio en el que se encuentran las abejas en ocho subespacios, de forma similar a la división de un árbol; esto con el fin de minimizar tiempos de operación y mayor facilidad en la búsqueda.

Las abejas con posible riesgo de colisión, se almacenan en una linkedList que contiene al nodo respectivo de acuerdo a sus coordenadas.

# Operaciones de la Estructura de Datos



**Gráfico 2:** Abstracción de la división del espacio.

Método	Complejidad( $O(n)$ )
leerArchivo()	$O(n)$
detectarColisiones()	$O(n)$ para el mejor de los casos (Abejas en un espacio pequeño) $O(n \log(n))$ en promedio y para el peor de los casos
guardarArchivo()	$O(n)$
inicio()	$O(m)$ (Con $m$ el número de veces que se debe sumar 100 a 0 para llegar a un entero arbitrario)
insert()	$O(\log(n))$
returnig()	$O(k)$ (Con $k$ el número de espacios que tengan más de dos abejas dentro de ellos)
ListaPeligro()	$O(l)$ (Con $l \leq n$ )

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

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

- La estructura que tiene el programa para determinar las colisiones es esencial para el buen desarrollo de las operaciones que este ejecuta.
- La estructura de datos elegida, divide el espacio recursivamente en ocho subespacios; como la ramificación de un árbol en el que cada nodo equivale a un espacio menor al anterior. Esto permite que la búsqueda sea más exacta pues se analiza cada sitio sin excepciones, además de que la búsqueda es más rápida, el espacio de memoria puede ser menor pues se limita la búsqueda a las dimensiones de un cubo que se crea con las coordenadas de la abeja en el punto minimo y maximo.

# Consumo de Tiempo y Memoria

Tiempo de ejecución:

Octree	10	100	1000	10000	100000	1000000
Lectura archivo	0,26 ms	0,27 ms	1,09 ms	10,33 ms	124,14 ms	1143,05 ms
Deteccion de colisiones	0,08 ms	0,08 ms	0,29 ms	2,14 ms	23,27 ms	237,02 ms
guardar archivo	0,14 ms	0,27 ms	0,39 ms	12,13 ms	117,98 ms	1186,36 ms

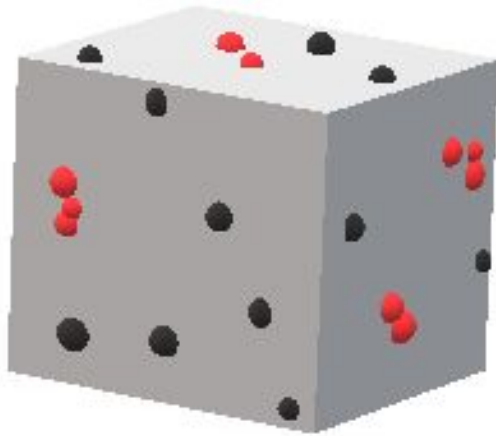
Memoria:

Octree	10	100	1000	10000	100000	1000000
Lectura archivo	0,00026 MB	0,00027 MB	0,00109 MB	0,001033 MB	0,12414 MB	1,14305 MB
Deteccion de colisiones	0,00008 MB	0,00008 MB	0,00029 MB	0,000214 MB	0,02327 MB	0,23702 MB
guardar archivo	0,00014 MB	0,00027 MB	0,00039 MB	0,001213 MB	0,11798 MB	1,18636 MB

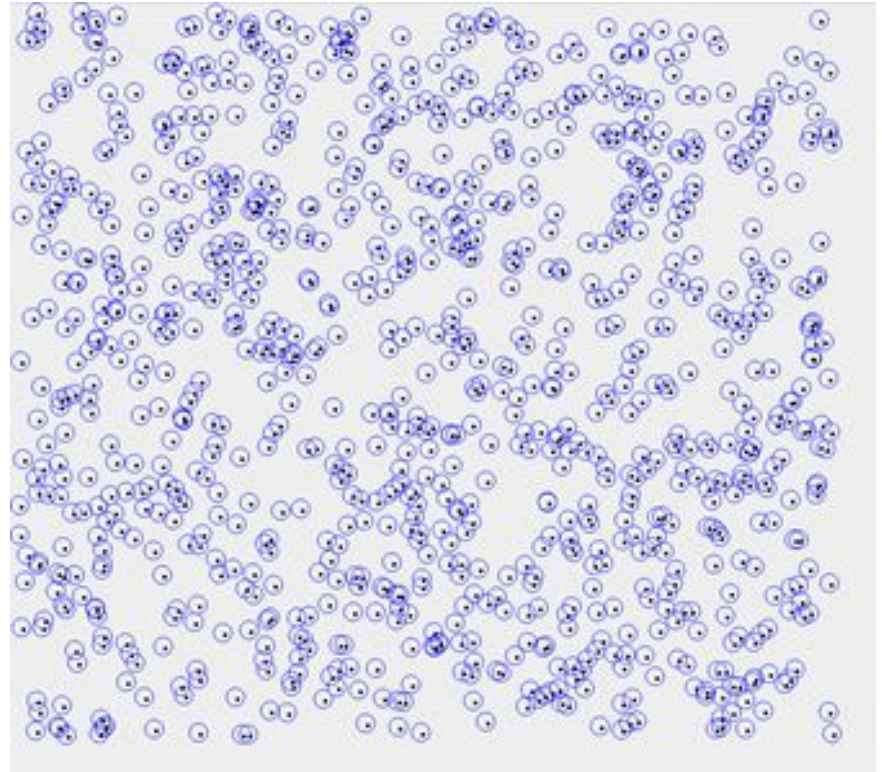
Análisis de los resultados:

Octree	Mejor tiempo	Peor Tiempo	Tiempo promedio	Mejor Memoria	Peor Memoria	Memoria promedio
Lectura archivo	0 ms	1389 ms	694,5 ms	0 MB	1,389 MB	0,6945 MB
Deteccion de colisiones	0 ms	338 ms	169 ms	0 MB	0,338 MB	0,169 MB
guardar archivo	0 ms	1320 ms	660 ms	0 MB	1,32 MB	0,66 MB

# Software Desarrollado



**Gráfico 3:** Modelo de las abejas en el espacio. Los círculos rojos representan las abejas con peligro de colisión.



**Gráfico 4:** Conjunto de 1000 abejas