

ALGORITMO PARA PREVENIR COLISIONES ENTRE ABEJAS ROBOTICAS

Juan Camilo Guerrero Alarcon
Santiago Pulgarin Vasquez
Medellín, 6 de Noviembre

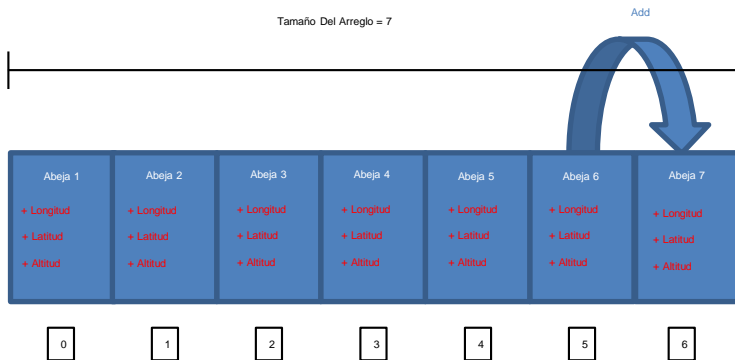
Estructuras de Datos Diseñada

Tamaño Del Arreglo = 6

Abeja 1	Abeja 2	Abeja 3	Abeja 4	Abeja 5	Abeja 6
+ Longitud	+ Longitud	+ Longitud	+ Longitud	+ Longitud	+ Longitud
+ Latitud	+ Latitud	+ Latitud	+ Latitud	+ Latitud	+ Latitud
+ Altitud	+ Altitud	+ Altitud	+ Altitud	+ Altitud	+ Altitud
0	1	2	3	4	5

contiene

Operaciones de la Estructura de Datos



Gráfica 2: Imagen de una operación Add de un ArrayList.

Tabla 1: Tabla para reportar la complejidad

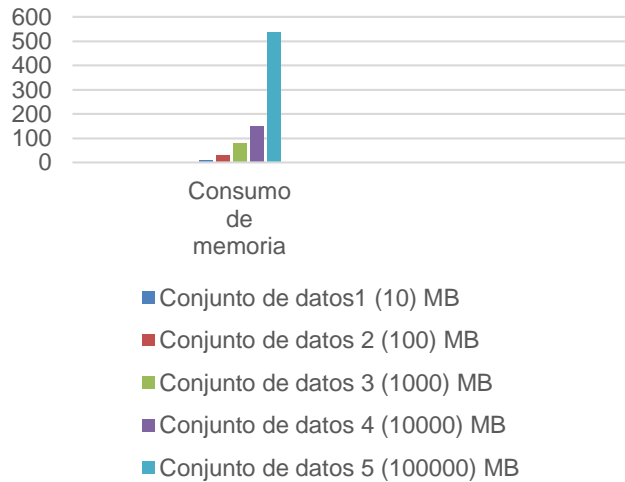
Metodo	Complejidad
distancia()	$O(n)$
leerArchivo	$O(n)$
detectarColisiones()	$O(n^2)$
guardarArchivo()	$O(n)$
main()	$O(1)$

Criterios de Diseño de la Estructura de Datos

La razón por la cual escogimos la estructura de datos Arraylist es debido a que es una de las más usadas, más conocidas y fácil de implementar a tipos de problemas como este, además de su excelente implementación que facilita el análisis de los datos, aunque este tipo de estructura no es muy eficiente y nos genera un déficit en memoria y en tiempo de procesamiento lo que genera poca efectividad para grandes cantidades de datos.

Consumo de Tiempo y Memoria

Consumo de Memmoria

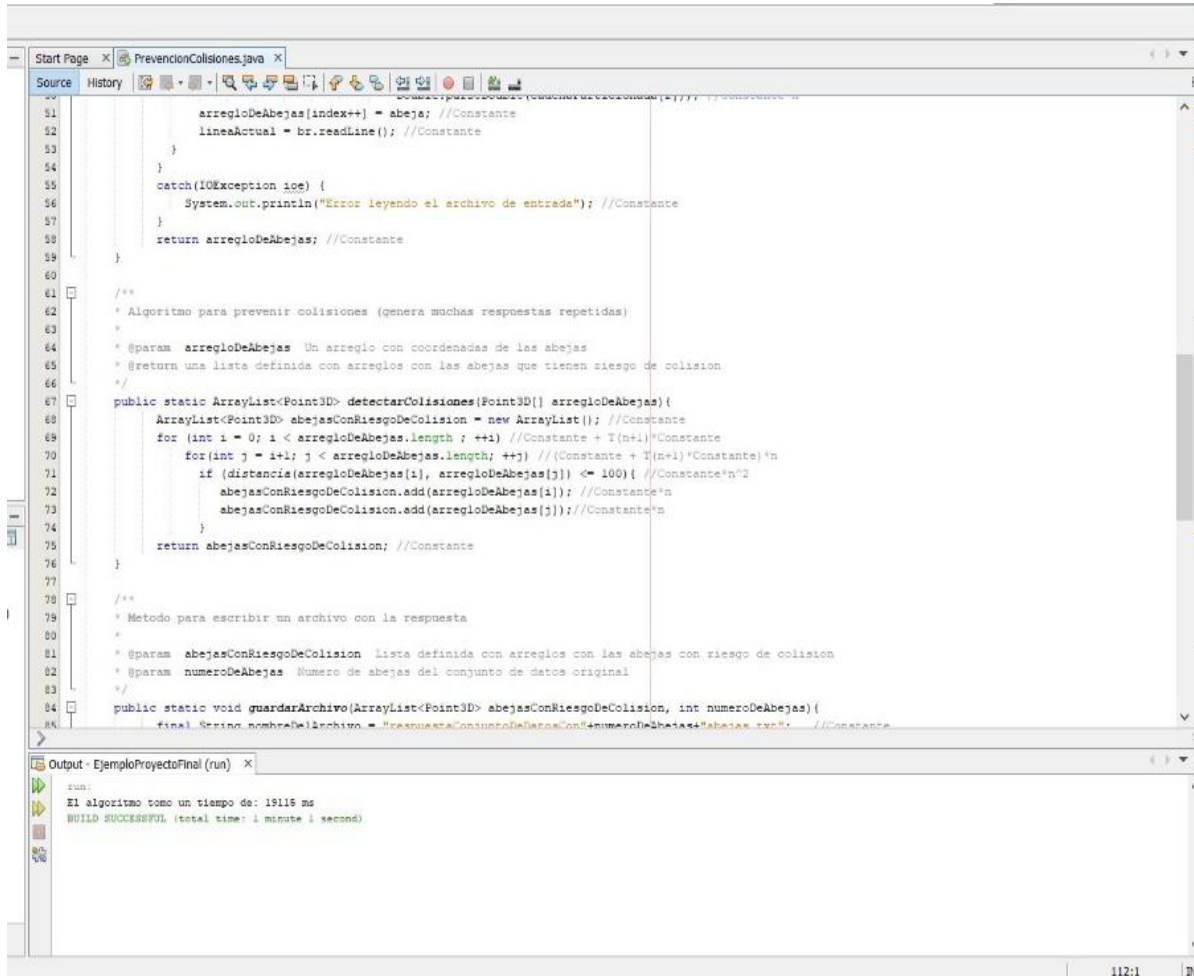


	Mejor Tiempo (segundos)	Peor Tiempo (segundos)	Tiempo Promedio (segundos)	Mejor Memoria (MB)	Peor Memoria (MB)	Memoria Promedio (MB)
Conjunto de datos1 (10)	0	0	0	7	11	9
Conjunto de datos 2 (100)	0,001	0,003	0,001683168	20	40	30
Conjunto de datos 3 (1000)	0,011	0,015	0,013138614	60	100	80
Conjunto de datos 4 (10000)	0,319	0,39	0,327653465	100	200	150
Conjunto de datos 5 (100000)	18,46	20,786	19,54	400	680	540



Inserten sus propias gráficas y sus explicaciones

Software Desarrollado



The screenshot displays an IDE window titled 'PrevisionColisiones.java'. The code implements a collision prevention algorithm for bees. It includes a method to read input data from a file, a method to detect collisions between bees based on a distance threshold, and a method to save the results to a file. The code is written in Java and uses standard data structures like ArrayList and Point3D.

```
51         arregloDeAbejas[index++] = abeja; //Constante
52         lineaActual = br.readLine(); //Constante
53     }
54 }
55 catch(IOException ioe) {
56     System.out.println("Error leyendo el archivo de entrada"); //Constante
57 }
58 return arregloDeAbejas; //Constante
59 }
60
61 /**
62  * Algoritmo para prevenir colisiones (genera muchas respuestas repetidas)
63  *
64  * @param arregloDeAbejas Un arreglo con coordenadas de las abejas
65  * @return una lista definida con arreglos con las abejas que tienen riesgo de colision
66  */
67 public static ArrayList<Point3D> detectarColisiones(Point3D[] arregloDeAbejas){
68     ArrayList<Point3D> abejasConRiesgoDeColision = new ArrayList(); //Constante
69     for (int i = 0; i < arregloDeAbejas.length; ++i) //Constante + T(n+1)*Constante
70         for(int j = i+1; j < arregloDeAbejas.length; ++j) //(Constante + T(n+1)*Constante)*n
71             if (distancia(arregloDeAbejas[i], arregloDeAbejas[j]) <= 100){ //Constante*n^2
72                 abejasConRiesgoDeColision.add(arregloDeAbejas[i]); //Constante*n
73                 abejasConRiesgoDeColision.add(arregloDeAbejas[j]); //Constante*n
74             }
75     return abejasConRiesgoDeColision; //Constante
76 }
77
78 /**
79  * Metodo para escribir un archivo con la respuesta
80  *
81  * @param abejasConRiesgoDeColision Lista definida con arreglos con las abejas con riesgo de colision
82  * @param numeroDeAbejas Numero de abejas del conjunto de datos original
83  */
84 public static void guardarArchivo(ArrayList<Point3D> abejasConRiesgoDeColision, int numeroDeAbejas){
85     final String nombreDelArchivo = "respuestaConjuntoDeAbejasCon"+numeroDeAbejas+"abejas.txt"; //Constante
```

The Output window shows the following message:

```
run:
El algoritmo tomo un tiempo de: 19116 ms
BUILD SUCCESSFUL (total time: 1 minute 1 second)
```