Montículo Binomial

En la presente memoria se explica el código escrito en Java utilizando el paradigma de orientación a objetos.

Operaciones disponibles

Las operaciones disponibles para el usuario de esta estructura son las siguientes:

-insertar(Elem) – inserta un elemento el montículo con coste amortizado O(1)

-minimo() – devuelve el mínimo del montículo con coste O(1)

-borraMinimo() – borra el mínimo del montículo con coste O(log(n))

-decrecerClave(Nodo, Elem) – decrece la clave del nodo con el valor elemento en coste O(log(n))

Todas estas operaciones internamente están programadas en base a los algoritmos proporcionados en las transparencias de los montículos unión.

Estructura y descripción de las clases

Se constan de los siguientes paquetes:

-mb – contiene las siguientes clases:

- MonticuloBinomial.java - implementación del montículo
- MonticuloVacioException.java - excepción que se lanza cuando el montículo está vacío y se quiere borrar u obtener el mínimo

-mb_costes – es una copia casi exacta del paquete mb y tiene las siguientes clases:

- MonticuloBinomial.java – igual que mb.MonticuloBinomial.java pero con la diferencia de que se incluye un entero para almacenar el coste de la última operación y hacer el cálculo, también se incluye un método para obtener este coste
- MonticuloVacioException.java – igual que mb.MonticuloVacioException.java

-test – contiene lo siguiente:

- Test.java – tiene tres métodos, dos para testear la inserción y el borrado y otro para para probar decrecer clave, en todas las pruebas se utiliza una cola de prioridad del repertorio de estructuras proporcionado por Java para poder comprobar el correcto funcionamiento del montículo programado

-graficas – tiene únicamente la siguiente clase:

- CalculoCostes.java – en la función principal se hace una serie de llamadas a métodos encargados de generar un informe de costes a partir de la utilización de la estructura, utiliza el paquete mb_costes
- CalculoTiempos.java – encargada de calcular los tiempos de cada una de las operaciones, utiliza el paquete mb

-vista – para la demo se programó la siguiente ventana:

- VentanaPrincipal.java – ventana para poder interactuar con las operaciones básicas de la estructura

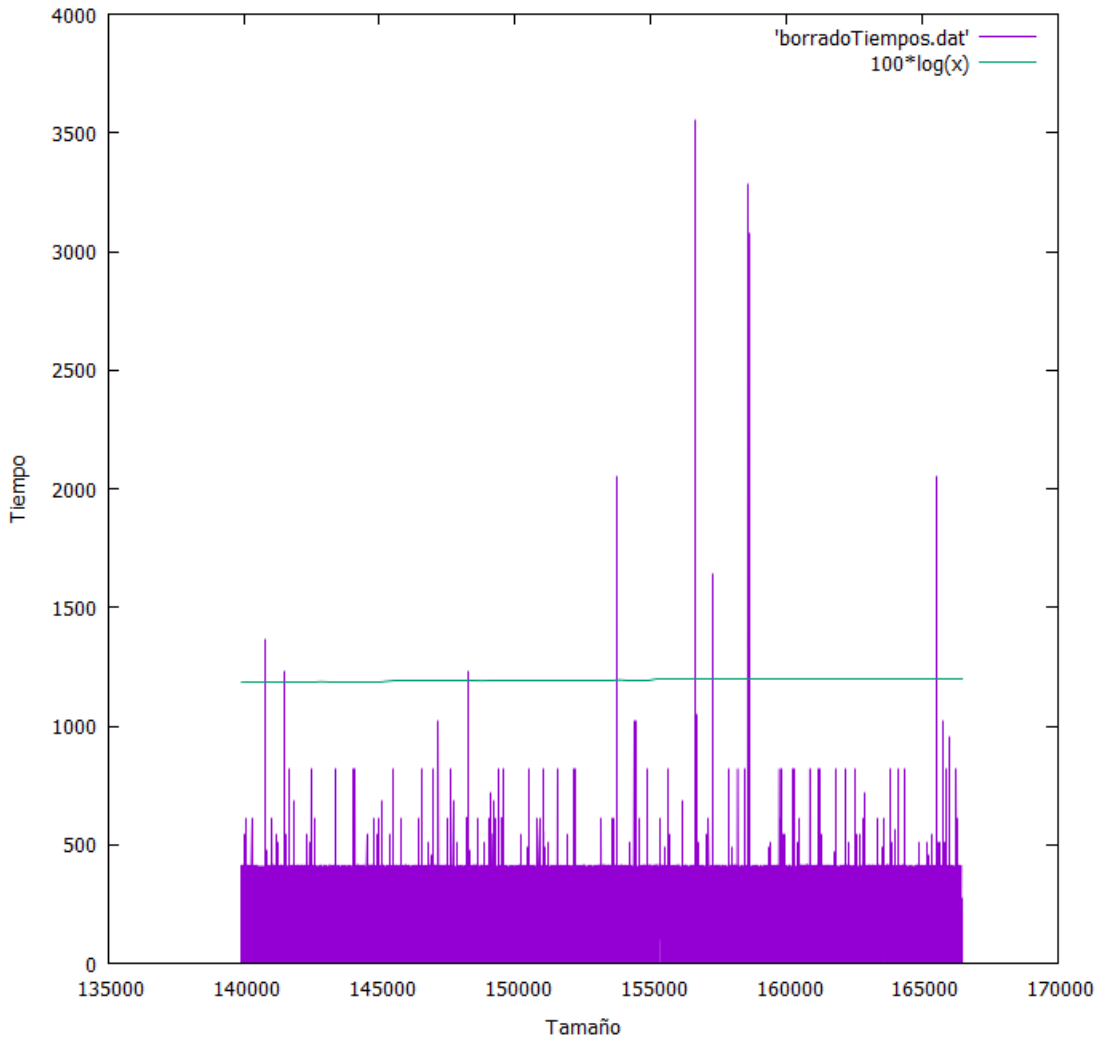
Gráficos

Se proporcionan las imágenes y los scripts para poder generar los gráficos. Si se desea se pueden ejecutar los métodos main de las clases CalculoTiempos.java y CalculoCostes.java y luego ejecutar los scripts desde un terminal de Gnuplot para regenerar los gráficos.

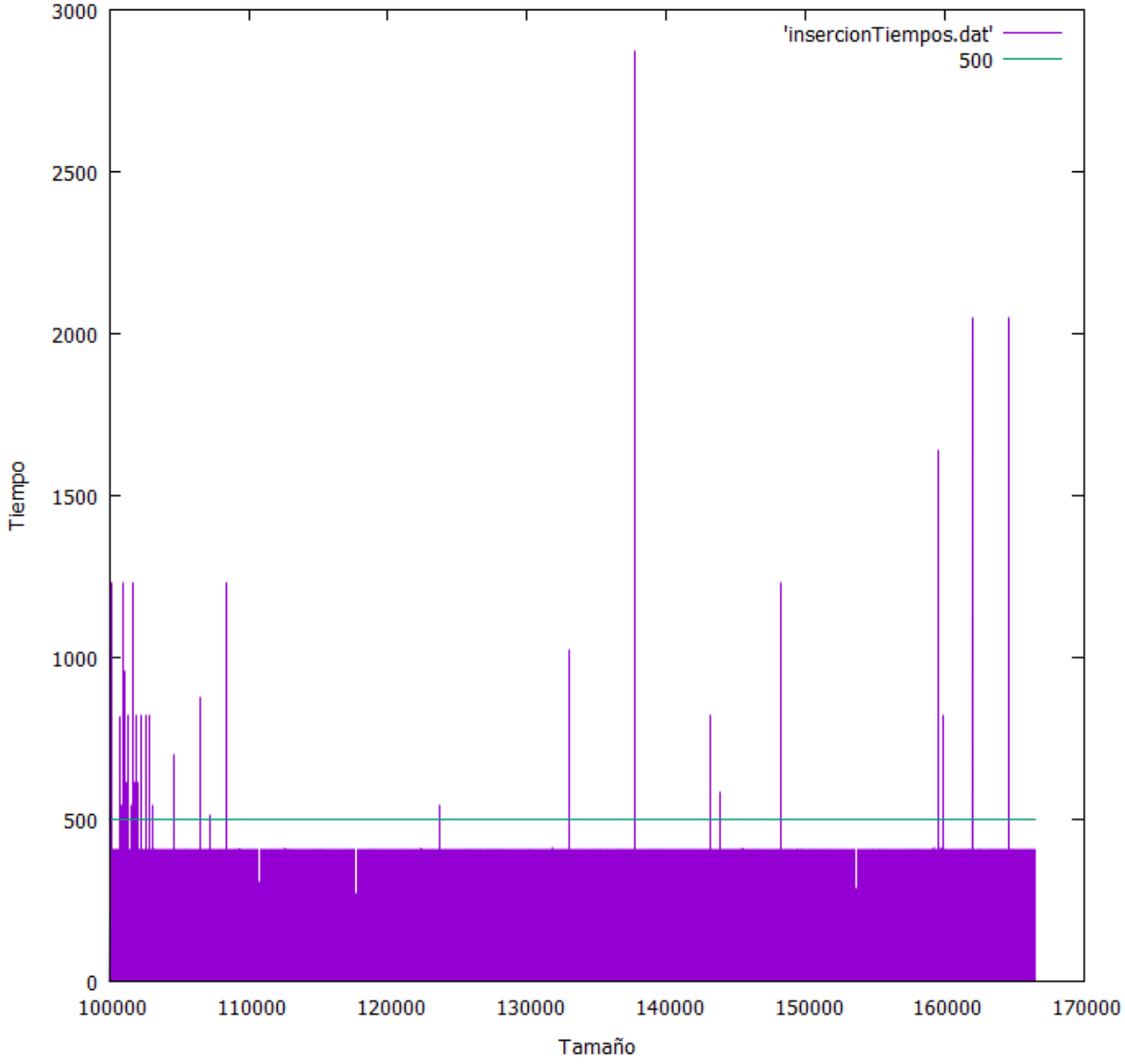
Gráficos de tiempos

Para generar estos gráficos se hizo uso de la clase CalculoTiempos.java. A pesar de haber utilizado el método más preciso que ofrece Java para capturar los tiempos, éste no proporciona tanta precisión como la deseada.

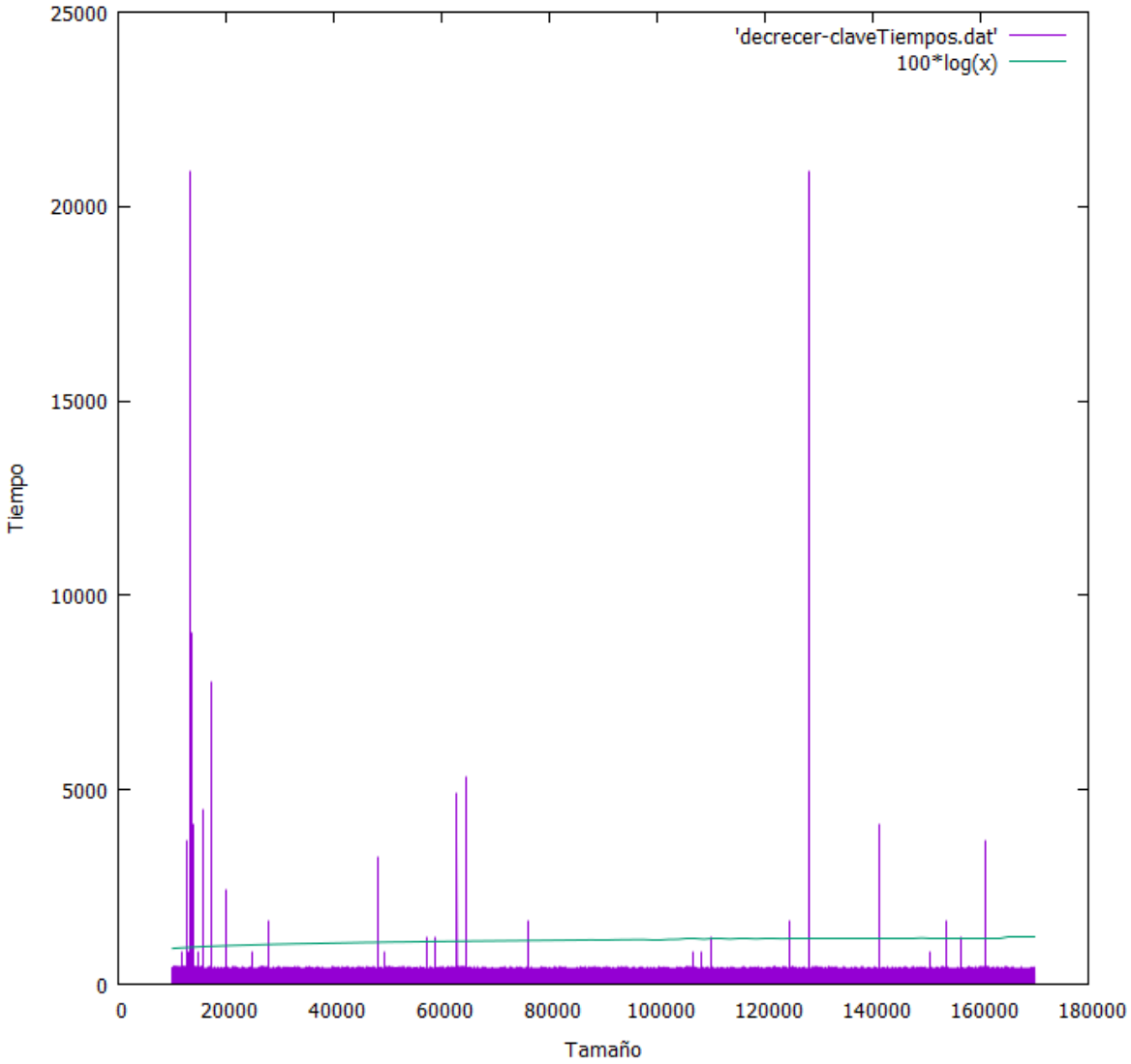
Borrado.-



Inserción.-


'insercionTiempos.dat'
500
Tiempo
Tamaño
0
500
1000
1500
2000
2500
3000
100000
110000
120000
130000
140000
150000
160000
170000

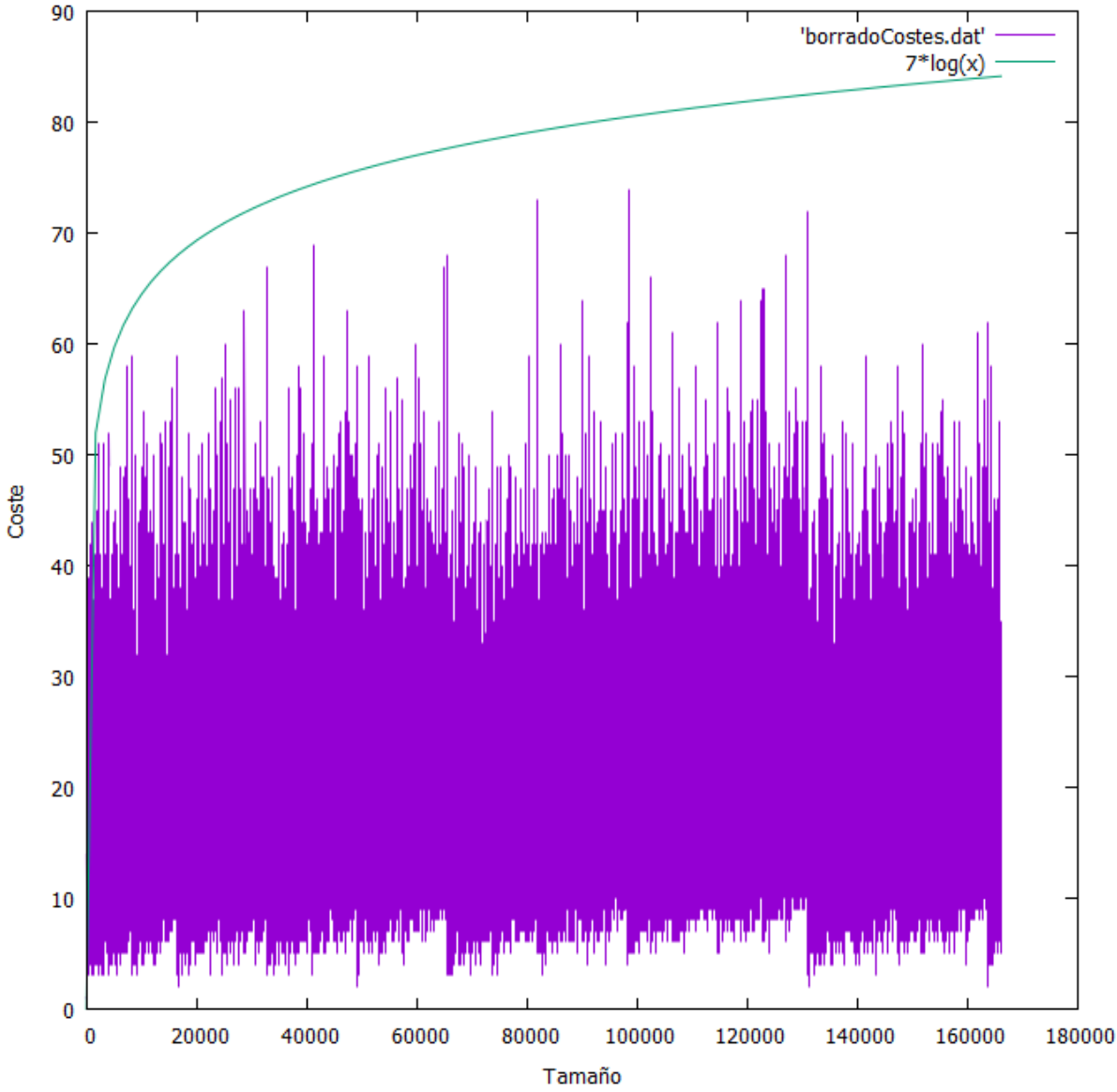
Decrecer clave.-



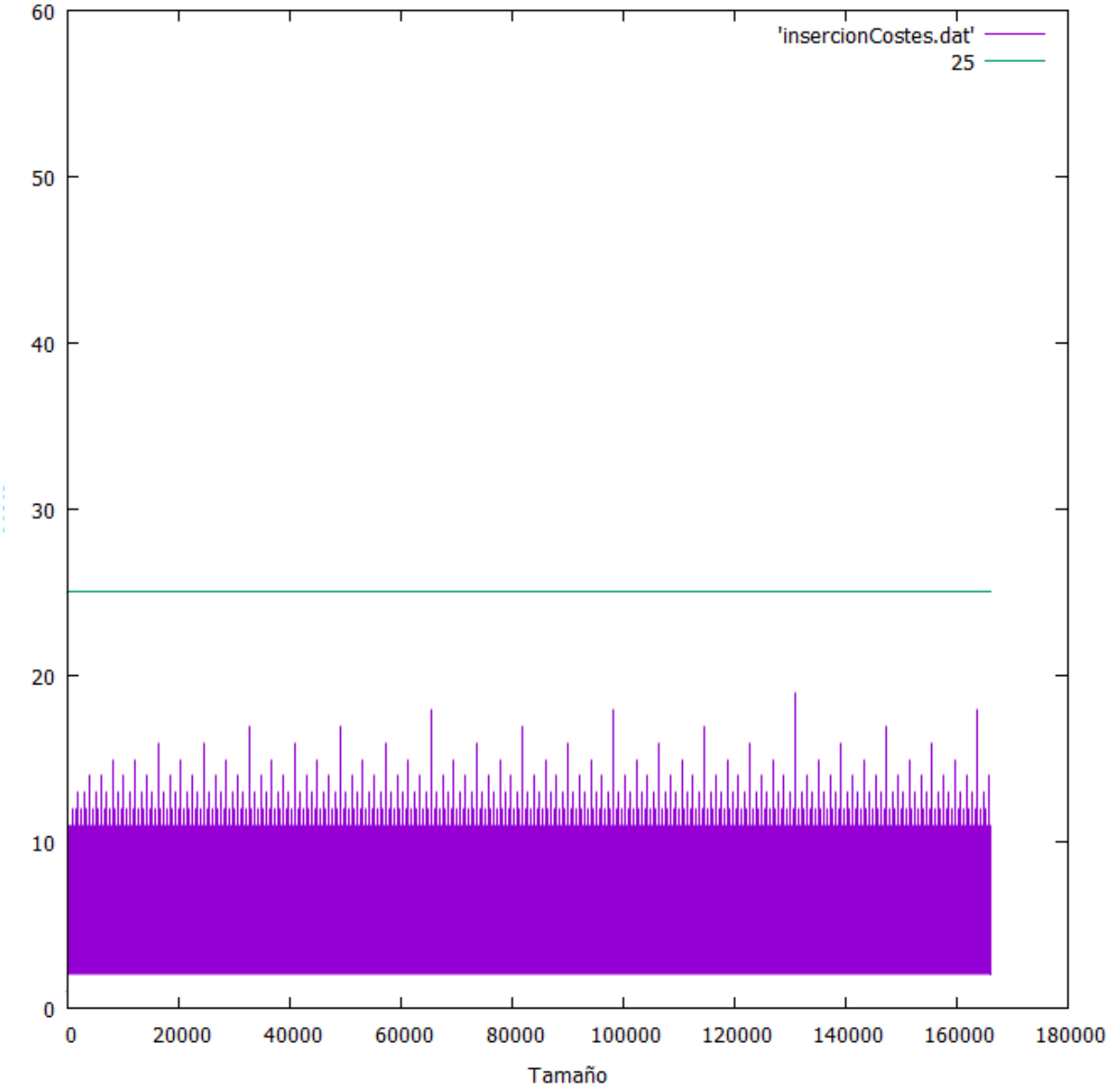
Gráficos de costes

Para generar estos gráficos se hizo uso de la clase CalculoCostes.java.

Borrado.-



Inserción. –


'insercionCostes.dat'
25
60
50
40
30
20
10
0
0
20000
40000
60000
80000
100000
120000
140000
160000
180000
Tamaño

Decrecer clave.-