

# **Análisis comparativo Tiempos de ejecución – Tarea EDD. Sem. 1 – 2018**

Profesor: Nicolás Rosso Chamorro.

Ayudante: John Bidwell Boitano.

Alumno: Javier Urzúa Ahumada.

Para este análisis se implementó la librería *Time*, mediante el siguiente código:

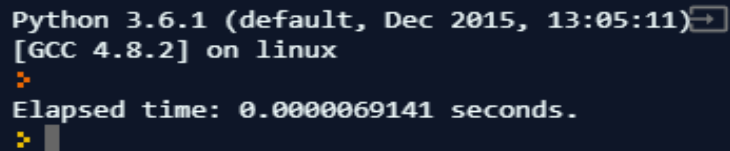
```
1  from time import time
2
3  start_time = time()
4  #funcion()
5  elapsed_time = time() - start_time
6  print("Elapsed time: %.10f seconds." % elapsed_time)
7
```

Que, al reemplazar *#función()* por determinada función o método, nos muestra el tiempo de ejecución en segundos. Dicho esto, se compararon los tiempos de ejecución de los métodos “agregar”, “encontrar”, “imprimir” y “borrar” de las distintas estructuras de datos evaluadas en esta tarea.

### Comparación método “agregar”:

-Lista (simplemente enlazada):

```
119
120     start_time = time()
121     b1.agregar_ordenado("Tenya", "Iida", 969, "Ingenium@ua.mail.com")
122     elapsed_time = time() - start_time
123     print("Elapsed time: %.10f seconds." % elapsed_time)
```



```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
>
Elapsed time: 0.0000069141 seconds.
>
```

-Árbol binario de búsqueda (ABB):

```
257     start_time = time()
258     b1.agregar(["Tenya", "Iida", 969, "Ingenium@ua.mail.com"])
259     elapsed_time = time() - start_time
260     print("Elapsed time: %.10f seconds." % elapsed_time)
```

```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
>
Elapsed time: 0.0000076294 seconds.
> █
```

-Hash:

```
96 start_time = time()
97 b1.agregar_ordenado("Tenya", "Iida", 969, "Ingenium@ua.mail.com")
98 elapsed_time = time() - start_time
99 print("Elapsed time: %.10f seconds." % elapsed_time)
100
```

```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
>
Elapsed time: 0.0000205040 seconds.
> █
```

**Comparación metodo “borrar”:**

-Lista (simplemente enlazada):

```
107 start_time = time()
108 b1.borrar(["Tenya", "Iida"])
109 elapsed_time = time() - start_time
110 print("Elapsed time: %.10f seconds." % elapsed_time)
111
```

```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
>
Elapsed time: 0.0000030994 seconds.
> █
```

-Árbol binario de búsqueda (ABB):



```
258 start_time = time()
259 b1.borrar(969)
260 elapsed_time = time() - start_time
261 print("Elapsed time: %.10f seconds." % elapsed_time)
262
```

```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
>
Nombre: Tenya Iida numero: 969 email:
Ingenium@ua.mail.com
-----
Elapsed time: 0.0000832081 seconds.
>
```

-Hash:

```
98 start_time = time()
99 b1.borrar("Tenya","Iida")
100 elapsed_time = time() - start_time
101 print("Elapsed time: %.10f seconds." % elapsed_time)
102
```

```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
>
Eliminando a: Tenya Iida

Borrado.
-----
Elapsed time: 0.0001401901 seconds.
>
```

**Comparación método “encontrar”:**

-Lista (simplemente enlazada):

```
108 start_time = time()
109 b1.buscar("Tenya","Iida")
110 elapsed_time = time() - start_time
111 print("Elapsed time: %.10f seconds." % elapsed_time)
112
```

```
Python 3.6.1 (default, Dec 2015, 13:05:
[GCC 4.8.2] on linux
>
Buscando contacto....

Tenya Iida , numero: 969 , email:
Ingenium@ua.mail.com
-----
Elapsed time: 0.0001137257 seconds.
>
```

-Árbol binario de búsqueda (ABB):

```
258 start_time = time()
259 b1.find(969)
260 elapsed_time = time() - start_time
261 print("Elapsed time: %.10f seconds." % elapsed_time)
262
```

```
Python 3.6.1 (default, Dec 2015, 13:05:
[GCC 4.8.2] on linux
>
Nombre: Tenya Iida numero: 969 email:
Ingenium@ua.mail.com
-----
Elapsed time: 0.0000905991 seconds.
>
```

-Hash:

```
98 start_time = time()
99 b1.buscar("Iida")
100 elapsed_time = time() - start_time
101 print("Elapsed time: %.10f seconds." % elapsed_time)
102
```

```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
>
Buscando contacto...

numero: 969 email: Ingenium@ua.mail.com
Elapsed time: 0.0019342899 seconds.
>
```

### Comparación método “imprimir”:

-Lista (simplemente enlazada):

```
108 start_time = time()
109 b1.imprimir()
110 elapsed_time = time() - start_time
111 print("Elapsed time: %.10f seconds." % elapsed_time)
112
```

```
-----
Elapsed time: 0.0000741482 seconds.
```

-Árbol binario de búsqueda (ABB):

```
258 start_time = time()
259 b1.imprimir()
260 elapsed_time = time() - start_time
261 print("Elapsed time: %.10f seconds." % elapsed_time)
262
263
```

```
-----
Elapsed time: 0.0001754761 seconds.
```

-Hash:

```
98 start_time = time()
99 b1.imprimir()
100 elapsed_time = time() - start_time
101 print("Elapsed time: %.10f seconds." % elapsed_time)
102
```

```
-----
Elapsed time: 0.0002279282 seconds.
```

Se nota que para imprimir y borrar, la lista simple supera con diferencia a Hash y ABB, dado que estos métodos consisten en recorrer el conjunto de datos, y al ser la lista de orden lineal facilita esta tarea. Hash agrega valores rápidamente, debido a su id de hashing que ubica fácilmente al nodo. Mientras que ABB encuentra valores rápidamente, debido a las cualidades de la estructura.