

Proyecto 1: Algoritmos y Estructuras de Datos

Santiago Madariaga

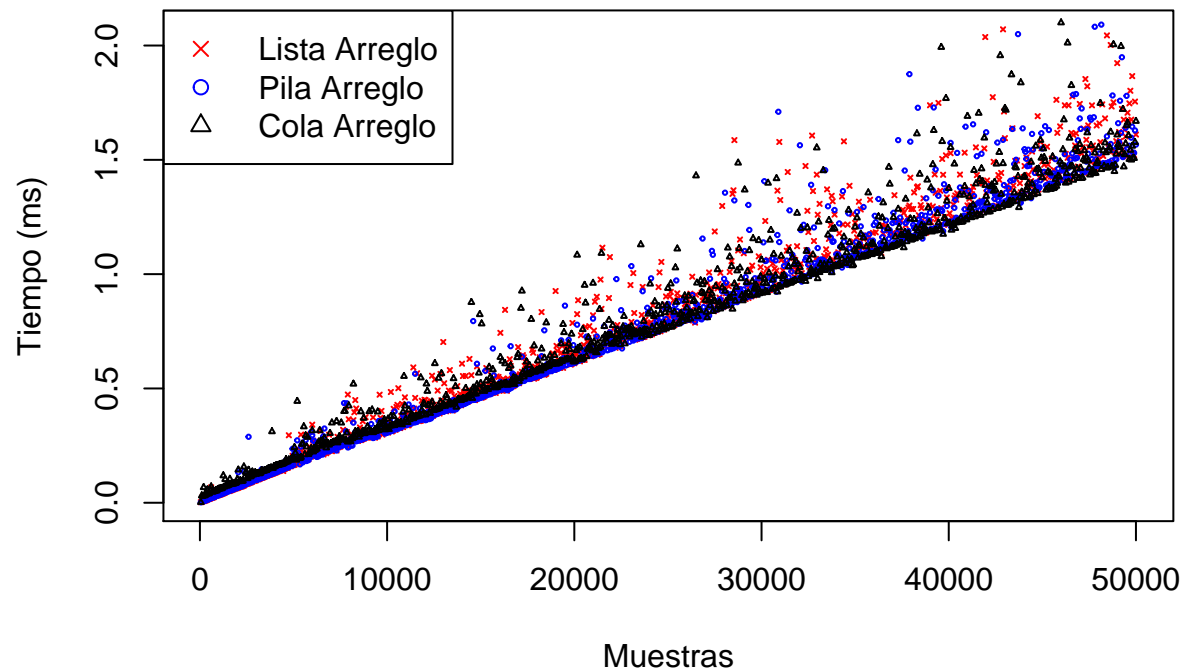
Gustavo Deza

Se generó un **Dataset** de **50.000** números pseudoaleatorios, usando la función **distribution(generator)** de la librería `<"random">`.

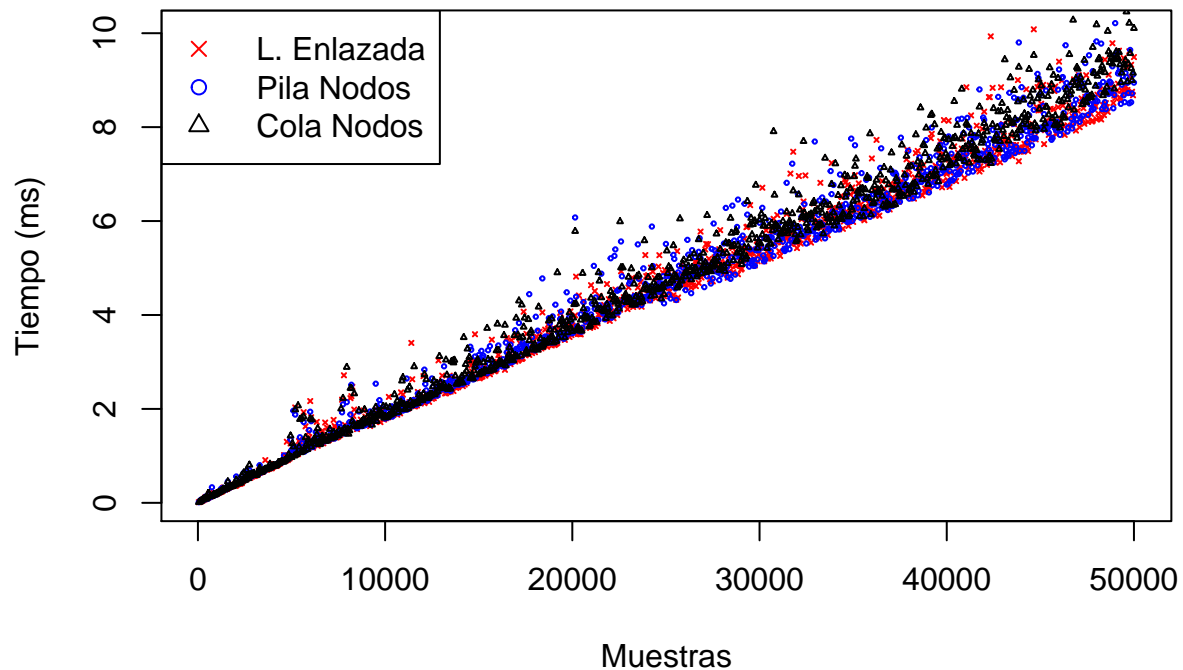
Se realizó un proceso de carga preliminar de los **1.000** datasets resultantes del **muestreo progresivo**. Esto para tener una idea preliminar de la complejidad de cada estructura implementada.

```
prl<-read.csv("~/Proyecto 1/test.txt")

plot(prl$Muestras,prl$ListaArreglo,col="red", xlab = "Muestras",
      ylab = "Tiempo (ms)", pch = 4, cex = 0.3)
points(prl$Muestras,prl$PilaArreglo,col="blue", pch = 1, cex = 0.3)
points(prl$Muestras,prl$ColaArreglo,col="black", pch = 2, cex = 0.3)
legend( x="topleft",
        legend=c("Lista Arreglo", "Pila Arreglo","Cola Arreglo"),
        col=c("red","blue","black"), lwd=1, lty=c(NA, NA, NA),
        pch=c(4,1, 2) )
```



```
plot(prl$Muestras,prl$ListaEnlazada,col="red", xlab = "Muestras",
      ylab = "Tiempo (ms)", pch = 4, cex = 0.3)
points(prl$Muestras,prl$PilaNodos,col="blue", pch = 1, cex = 0.3)
points(prl$Muestras,prl$ColaNodos,col="black", pch = 2, cex = 0.3)
legend( x="topleft",
        legend=c("L. Enlazada", "Pila Nodos","Cola Nodos"),
        col=c("red","blue","black"), lwd=1, lty=c(NA, NA, NA),
        pch=c(4,1, 2) )
```



```
Alista<-read.csv("~/Proyecto 1/Alista_t.txt")
```

Se puede observar, en el eje “y”, que hay una diferencia significativa en los tiempos de ejecución dependiendo del tipo de base de diseño de la estructura (si está basada en nodos, o en un vector).

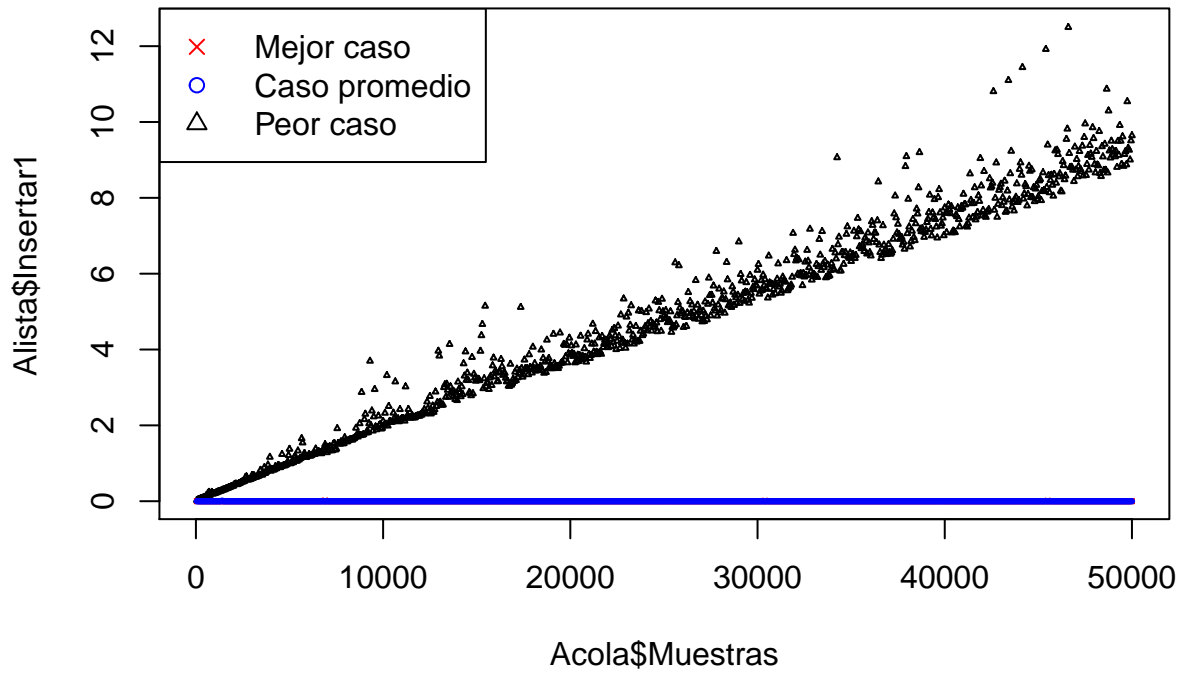
Ahora se mostrarán los tiempos en *inserción*, *eliminación*, *búsqueda* y *ordenamiento* (sólo en listas) de las estructuras implementadas:

Cola basada en Arreglos:

```
Acola<-read.csv("~/Proyecto 1/Acola_t.txt")
plot(Acola$Muestras,Alista$Insertar1,col="black", pch = 2, cex = 0.3)
title("Arreglo->Cola: Insertar")
points(Acola$Muestras,Acola$Insertar0,col="red", xlab = "Muestras",
       ylab = "Tiempo (ms)", pch = 4, cex = 0.3)
points(Acola$Muestras,Acola$InsertarM,col="blue", pch = 1, cex = 0.3)

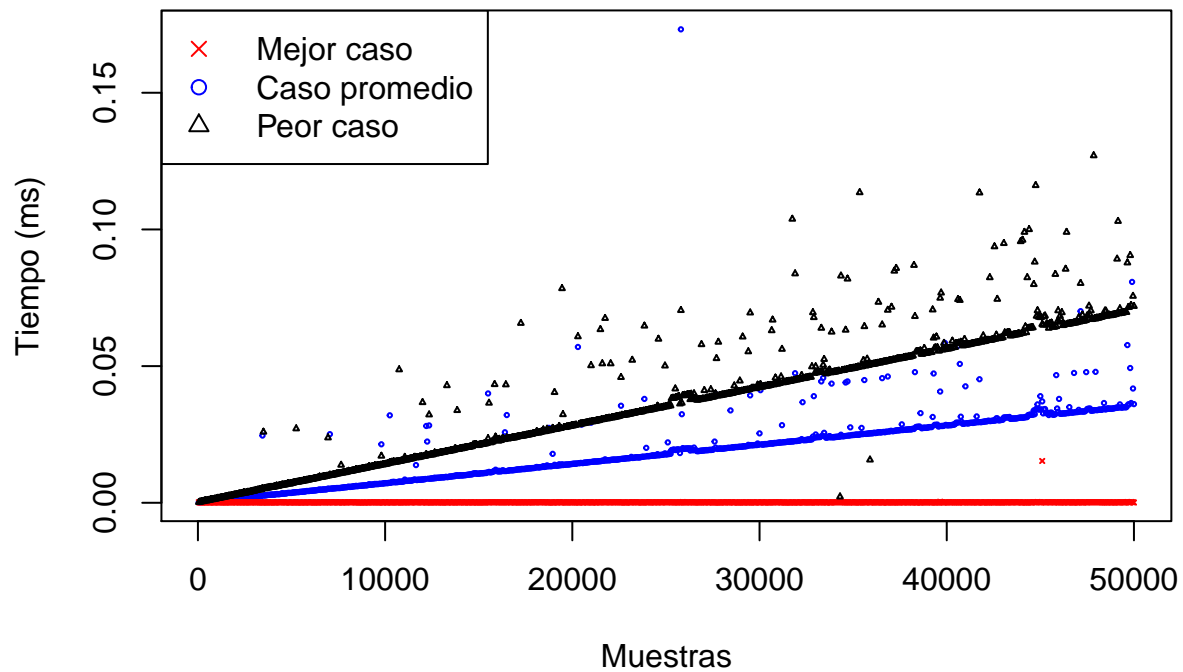
legend( x="topleft",
       legend=c("Mejor caso", "Caso promedio","Peor caso"),
       col=c("red","blue","black"), lwd=1, lty=c(NA, NA, NA),
       pch=c(4,1, 2) )
```

Arreglo->Cola: Insertar



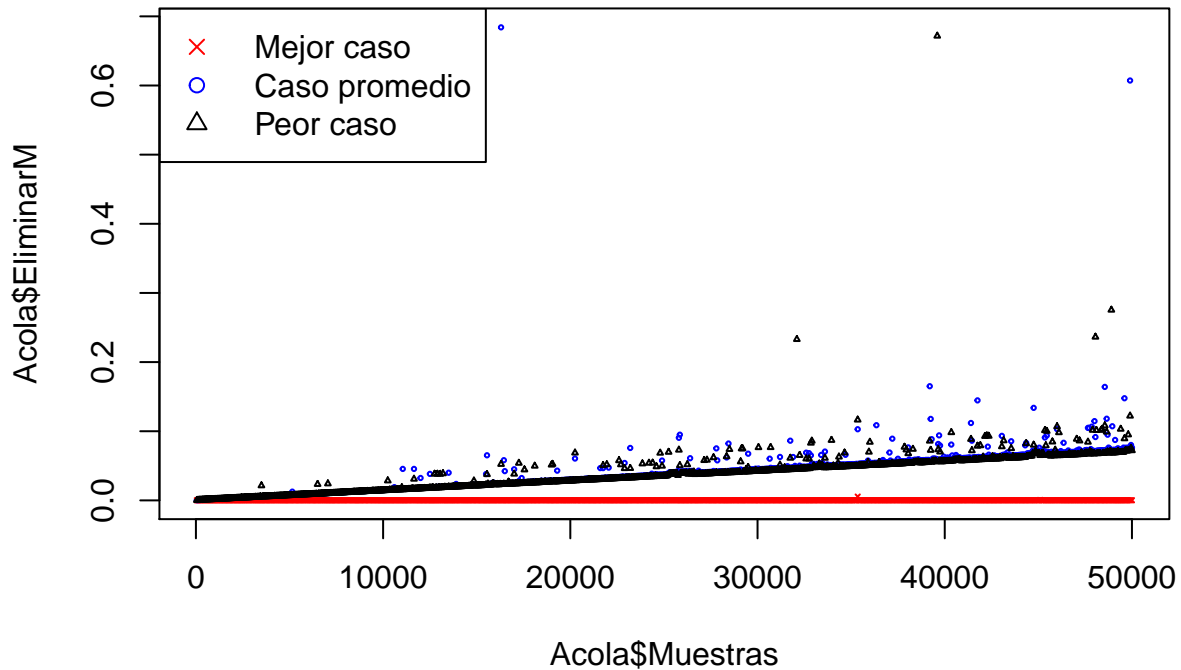
```
#-----
plot(Acola$Muestras,Acola$BuscarM,col="blue", pch = 1, cex = 0.3,
     xlab = "Muestras", ylab = "Tiempo (ms)")
title("Arreglo->Cola: Buscar")
points(Acola$Muestras,Acola$Buscar0,col="red", pch = 4, cex = 0.3)
points(Acola$Muestras,Acola$Buscar1,col="black", pch = 2, cex = 0.3)
legend( x="topleft",
       legend=c("Mejor caso", "Caso promedio","Peor caso"),
       col=c("red","blue","black"), lwd=1, lty=c(NA, NA, NA),
       pch=c(4,1, 2) )
```

Arreglo→Cola: Buscar



```
#-----  
plot(Acola$Muestras,Acola$EliminarM,col="blue", pch = 1, cex = 0.3)  
title("Arreglo->Cola: Eliminar")  
points(Acola$Muestras,Acola$Eliminar0,col="red", xlab = "Muestras",  
       ylab = "Tiempo (ms)", pch = 4, cex = 0.3)  
points(Acola$Muestras,Acola$Eliminar1,col="black", pch = 2, cex = 0.3)  
legend( x="topleft",  
       legend=c("Mejor caso", "Caso promedio","Peor caso"),  
       col=c("red","blue","black"), lwd=1, lty=c(NA, NA, NA),  
       pch=c(4,1, 2) )
```

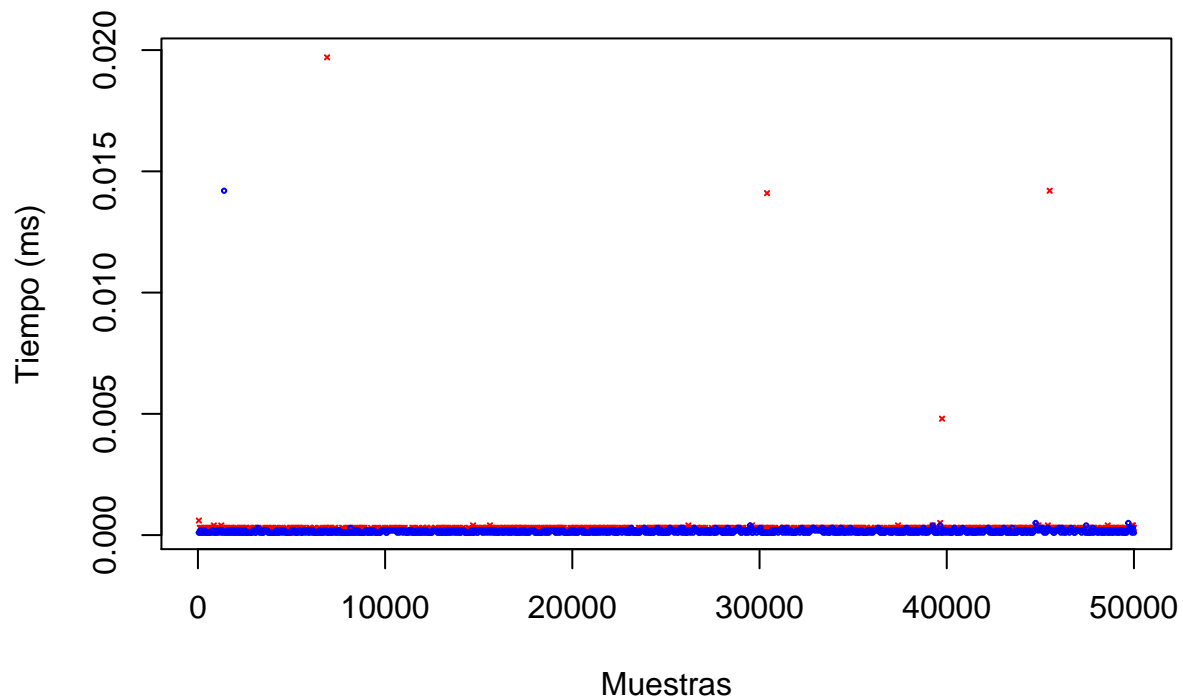
Arreglo→Cola: Eliminar



Se puede observar que, en el peor caso de inserción (donde se debe redimensionar el arraylist), el tiempo es excesivamente superior al de los demás casos, dejándolos fuera de escala.

```
plot(Acola$Muestras,Acola$Insertar0,col="red", xlab = "Muestras",  
      ylab = "Tiempo (ms)", pch = 4, cex = 0.3)  
title("Arreglo→Cola: Insertar")  
points(Acola$Muestras,Acola$InsertarM,col="blue", pch = 1, cex = 0.3)
```

Arreglo→Cola: Insertar



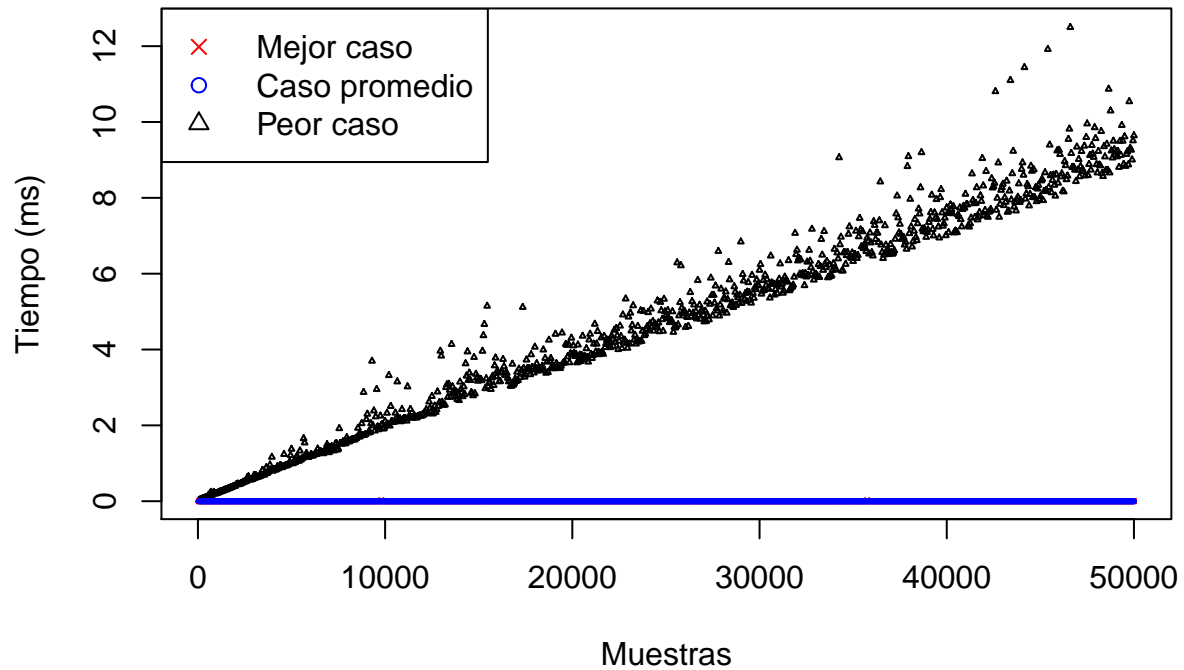
Removiendo el peor caso de la inserción se puede observar que no hay una diferencia significativa. Ya que simplemente se encola el elemento siguiente.

Listas basadas en arreglos/vectores:

```
plot(Alista$Muestras,Alista$Insertar1,col="black", pch = 2, cex = 0.3,
     xlab = "Muestras", ylab = "Tiempo (ms)")
title("Arreglo→Lista: Insertar")
points(Alista$Muestras,Alista$Insertar0,col="red", pch = 4, cex = 0.3)
points(Alista$Muestras,Alista$InsertarM,col="blue", pch = 1, cex = 0.3)

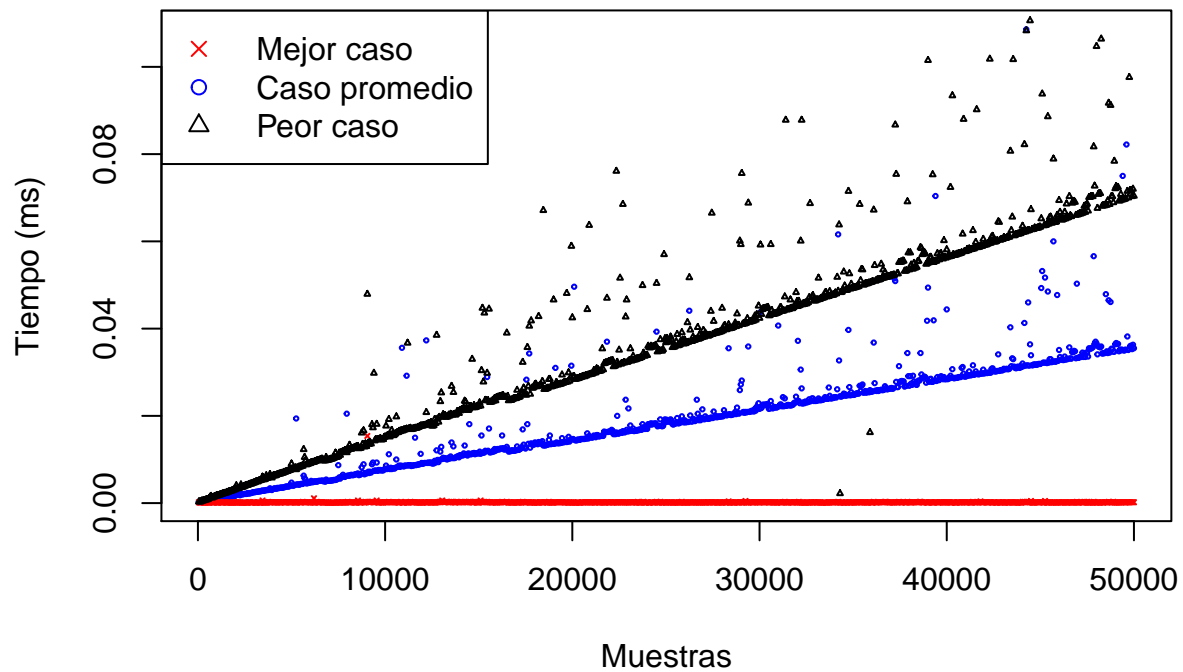
legend( x="topleft",
       legend=c("Mejor caso", "Caso promedio","Peor caso"),
       col=c("red","blue","black"), lwd=1, lty=c(NA, NA, NA),
       pch=c(4,1, 2) )
```

Arreglo->Lista: Insertar



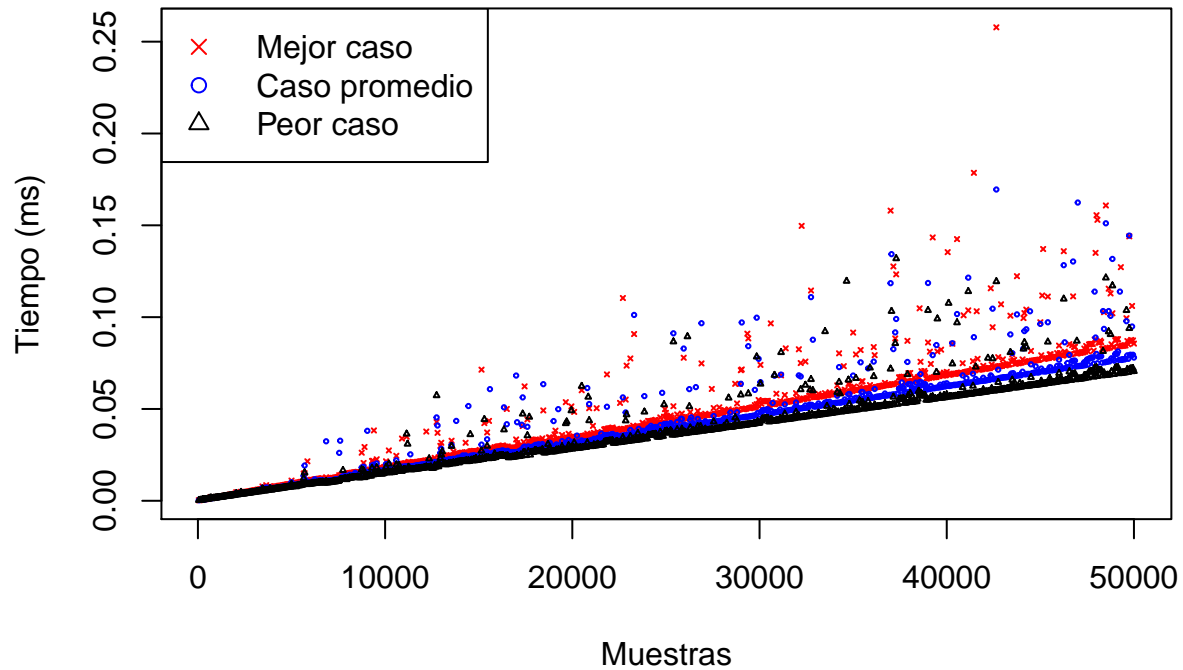
```
#-----  
plot(Alista$Muestras,Alista$BuscarM,col="blue", pch = 1, cex = 0.3,  
      xlab = "Muestras", ylab = "Tiempo (ms)")  
title("Arreglo->Lista: Buscar")  
points(Alista$Muestras,Alista$Buscar0,col="red", pch = 4, cex = 0.3)  
points(Alista$Muestras,Alista$Buscar1,col="black", pch = 2, cex = 0.3)  
legend( x="topleft",  
        legend=c("Mejor caso", "Caso promedio","Peor caso"),  
        col=c("red","blue","black"), lwd=1, lty=c(NA, NA, NA),  
        pch=c(4,1, 2) )
```


Arreglo→Lista: Buscar



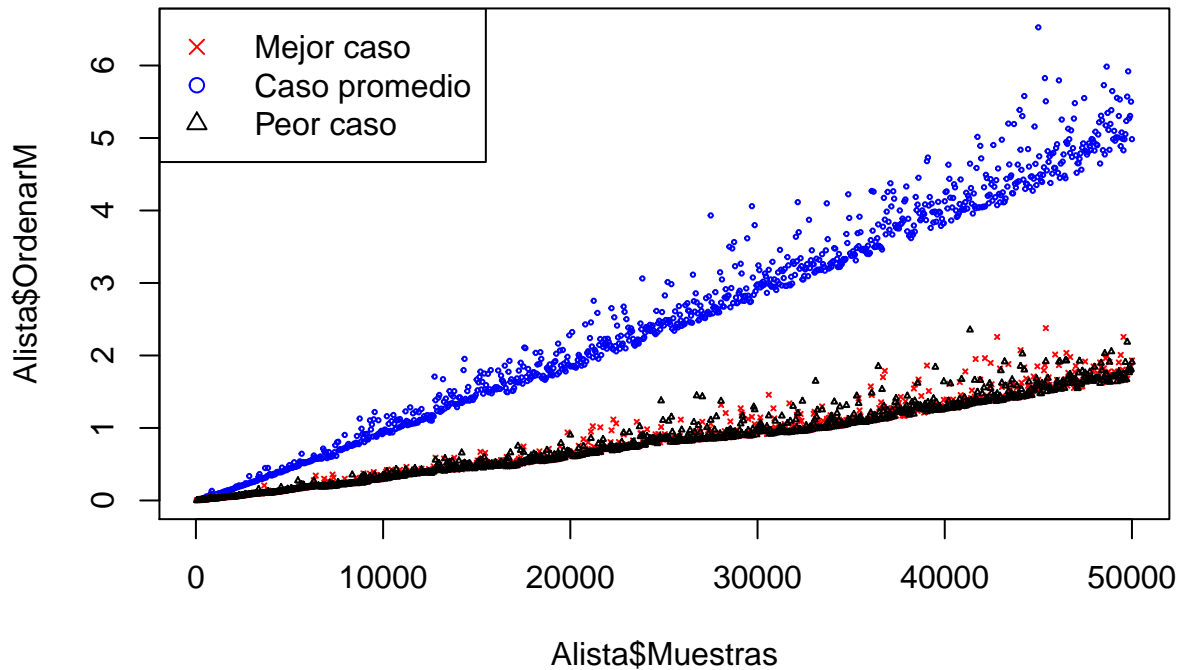
```
#-----  
plot(Alista$Muestras,Alista$Eliminar0,col="red", xlab = "Muestras",  
      ylab = "Tiempo (ms)", pch = 4, cex = 0.3)  
title("Arreglo->Lista: Eliminar")  
points(Alista$Muestras,Alista$EliminarM,col="blue", pch = 1, cex = 0.3)  
points(Alista$Muestras,Alista$Eliminar1,col="black", pch = 2, cex = 0.3)  
legend( x="topleft",  
        legend=c("Mejor caso", "Caso promedio","Peor caso"),  
        col=c("red","blue","black"), lwd=1, lty=c(NA, NA, NA),  
        pch=c(4,1, 2) )
```

Arreglo->Lista: Eliminar



```
#-----  
plot(Alista$Muestras,Alista$OrdenarM,col="blue", pch = 1, cex = 0.3)  
title("Arreglo->Lista: Ordenar")  
points(Alista$Muestras,Alista$Ordenar0,col="red", xlab = "Muestras",  
       ylab = "Tiempo (ms)", pch = 4, cex = 0.3)  
points(Alista$Muestras,Alista$Ordenar1,col="black", pch = 2, cex = 0.3)  
legend( x="topleft",  
       legend=c("Mejor caso", "Caso promedio","Peor caso"),  
       col=c("red","blue","black"), lwd=1, lty=c(NA, NA, NA),  
       pch=c(4,1, 2) )
```

Arreglo->Lista: Ordenar



Al igual que la cola implementada a partir de arreglos/vectores, se puede observar que el tiempo de inserción, cuando se requiere el redimensionamiento del arreglo, es relativamente alto si se compara con los demás casos.

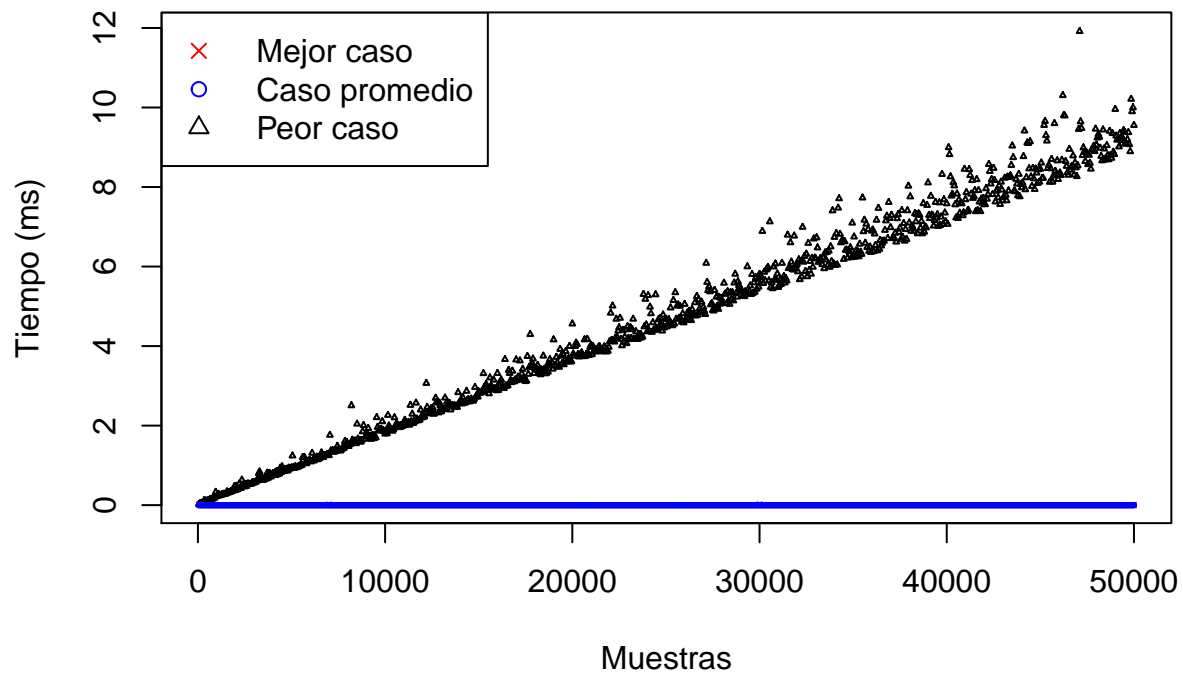
En el caso del caso promedio del ordenamiento (blue points), se puede observar que es mayor al de los demás casos. Esto debido a que se tomó como caso promedio al vector del dataset en su estado natural, es decir, aleatorio. Y la naturaleza del ordenamiento usado no se ve beneficiada de esto.

Pilas basadas en arreglos:

```
Apila<-read.csv("~/Proyecto 1/Apila_t.txt")
plot(Apila$Muestras,Apila$Insertar1,col="black", pch = 2, cex = 0.3,
     xlab = "Muestras", ylab = "Tiempo (ms)")
title("Arreglo->Pila: Insertar")
points(Apila$Muestras,Apila$Insertar0,col="red", pch = 4, cex = 0.3)
points(Apila$Muestras,Apila$InsertarM,col="blue", pch = 1, cex = 0.3)

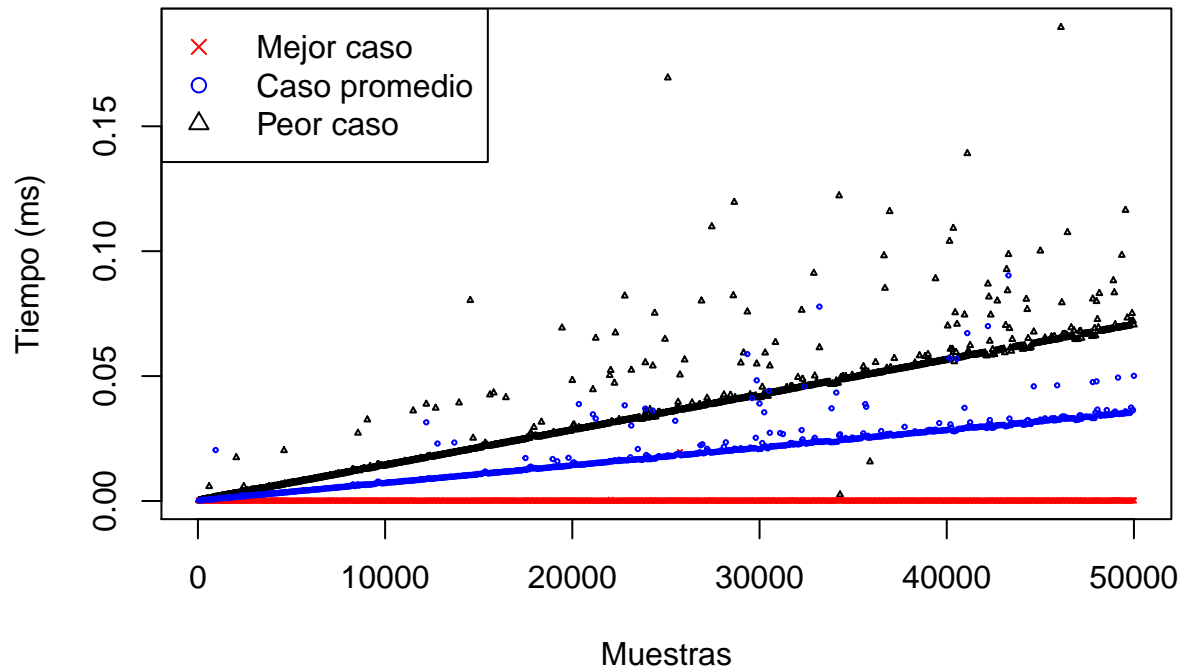
legend( x="topleft",
       legend=c("Mejor caso", "Caso promedio","Peor caso"),
       col=c("red","blue","black"), lwd=1, lty=c(NA, NA, NA),
       pch=c(4,1, 2) )
```

Arreglo→Pila: Insertar



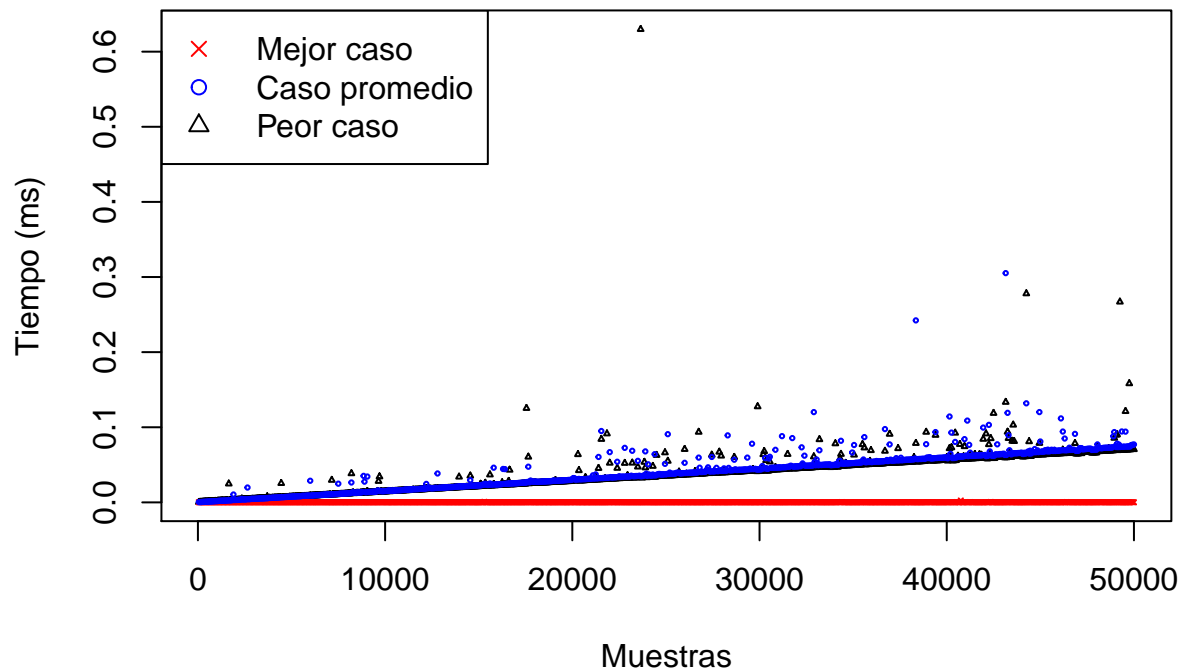
```
#-----  
plot(Apila$Muestras,Apila$Buscar1,col="black", pch = 2, cex = 0.3,  
     xlab = "Muestras", ylab = "Tiempo (ms)")  
  
title("Arreglo->Pila: Buscar")  
points(Apila$Muestras,Apila$Buscar0,col="red", pch = 4, cex = 0.3)  
points(Apila$Muestras,Apila$BuscarM,col="blue", pch = 1, cex = 0.3)  
  
legend( x="topleft",  
       legend=c("Mejor caso", "Caso promedio","Peor caso"),  
       col=c("red","blue","black"), lwd=1, lty=c(NA, NA, NA),  
       pch=c(4,1, 2) )
```

Arreglo->Pila: Buscar



```
#-----  
plot(Apila$Muestras,Apila$Eliminar1,col="black", pch = 2, cex = 0.3,  
      xlab = "Muestras", ylab = "Tiempo (ms)")  
points(Apila$Muestras,Apila$Eliminar0,col="red", pch = 4, cex = 0.3)  
title("Arreglo->Pila: Eliminar")  
points(Apila$Muestras,Apila$EliminarM,col="blue", pch = 1, cex = 0.3)  
  
legend( x="topleft",  
        legend=c("Mejor caso", "Caso promedio","Peor caso"),  
        col=c("red","blue","black"), lwd=1, lty=c(NA, NA, NA),  
        pch=c(4,1, 2) )
```

Arreglo→Pila: Eliminar



Al igual que toda estructura basada en arreglos/vectores, el tiempo de inserción cuando se redimensiona el vector interno es relativamente alto. Se pueden observar valores atípicos en la operación de Eliminar. Esto puede ser debido a la latencia de la memoria, subprocessos, etc.

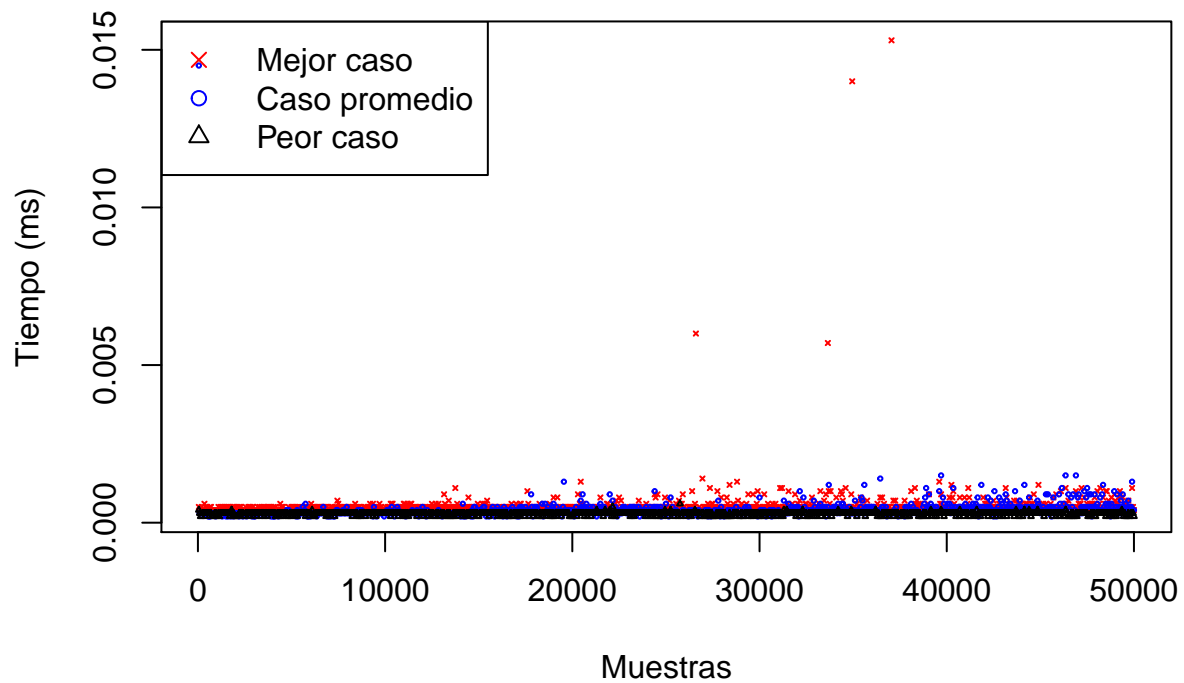
A continuación se presentan las mismas estructuras pero implementadas con **Nodos**:

Cola Basada en Nodos:

```
Ncola<-read.csv("~/Proyecto 1/Ncola_t.txt")

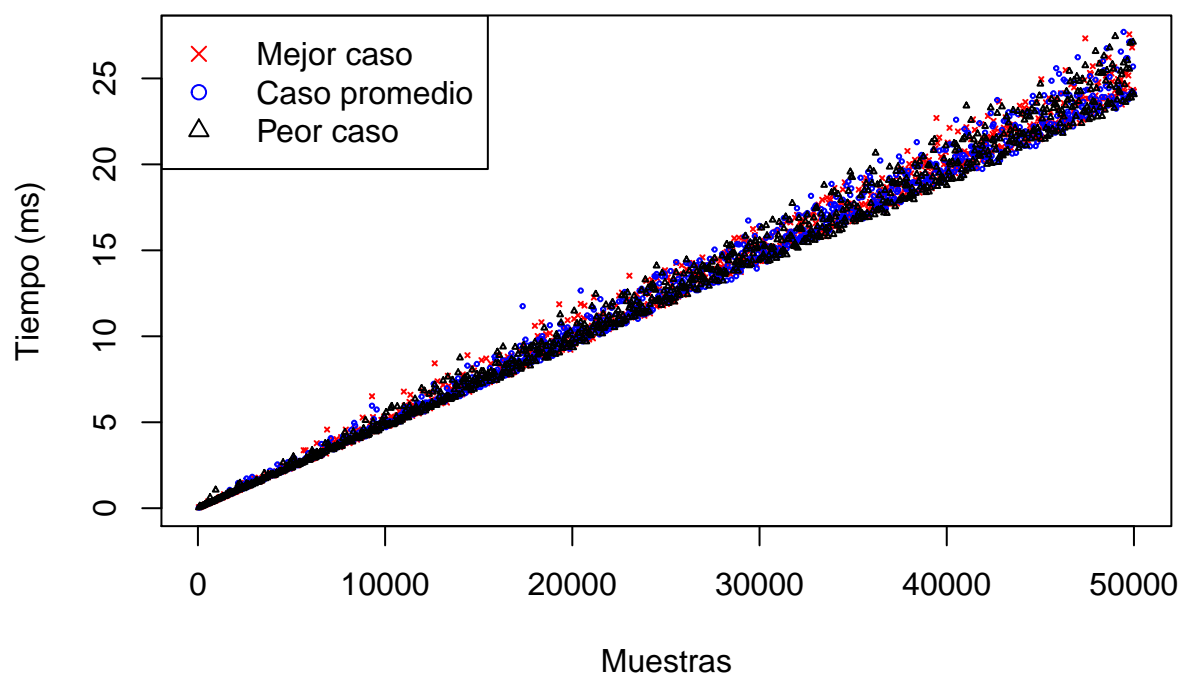
plot(Ncola$Muestras,Ncola$Insertar0,col="red", xlab = "Muestras",
     ylab = "Tiempo (ms)", pch = 4, cex = 0.3)
title("Nodo->Cola: Insertar")
points(Ncola$Muestras,Ncola$InsertarM,col="blue", pch = 1, cex = 0.3)
points(Ncola$Muestras,Ncola$Insertar1,col="black", pch = 2, cex = 0.3)
legend( x="topleft",
       legend=c("Mejor caso", "Caso promedio","Peor caso"),
       col=c("red","blue","black"), lwd=1, lty=c(NA, NA, NA),
       pch=c(4,1, 2) )
```

Nodo->Cola: Insertar



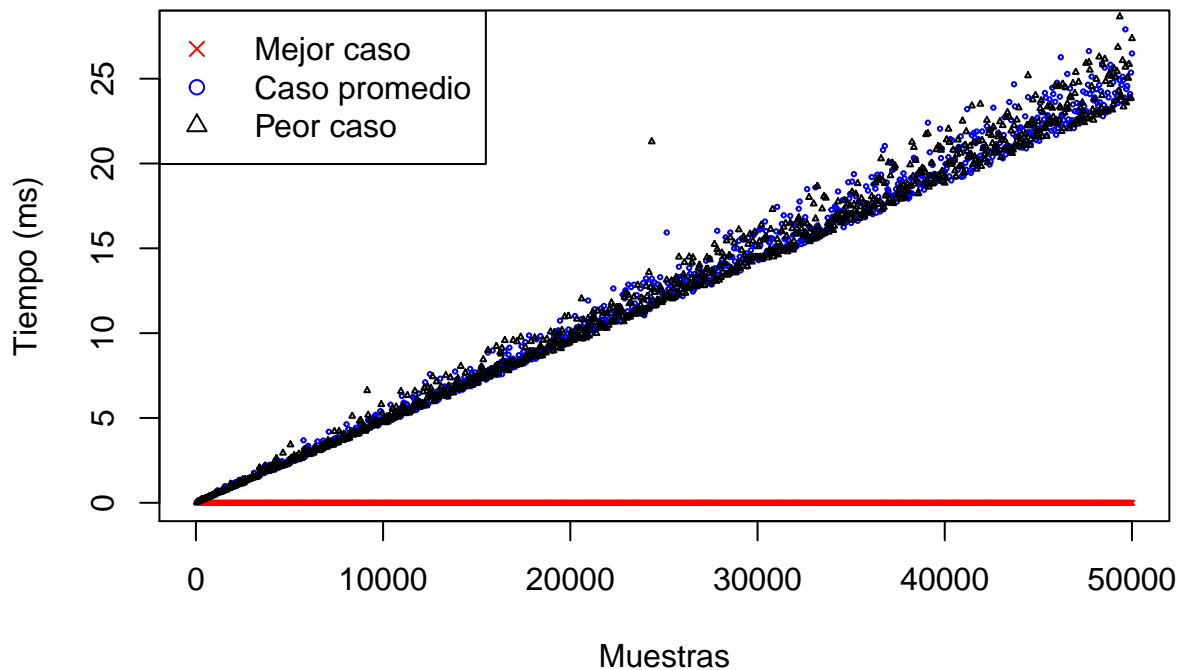
```
#-----  
plot(Ncola$Muestras,Ncola$Buscar0,col="red", xlab = "Muestras",  
      ylab = "Tiempo (ms)", pch = 4, cex = 0.3)  
title("Nodo->Cola: Buscar")  
points(Ncola$Muestras,Ncola$BuscarM,col="blue", pch = 1, cex = 0.3)  
points(Ncola$Muestras,Ncola$Buscar1,col="black", pch = 2, cex = 0.3)  
legend( x="topleft",  
        legend=c("Mejor caso", "Caso promedio","Peor caso"),  
        col=c("red","blue","black"), lwd=1, lty=c(NA, NA, NA),  
        pch=c(4,1, 2) )
```

Nodo→Cola: Buscar



```
#-----  
plot(Ncola$Muestras,Ncola$EliminarM,col="blue", pch = 1, cex = 0.3,  
      xlab = "Muestras", ylab = "Tiempo (ms)")  
title("Nodo->Cola: Eliminar")  
points(Ncola$Muestras,Ncola$Eliminar0,col="red", pch = 4, cex = 0.3)  
points(Ncola$Muestras,Ncola$Eliminar1,col="black", pch = 2, cex = 0.3)  
legend( x="topleft",  
        legend=c("Mejor caso", "Caso promedio","Peor caso"),  
        col=c("red","blue","black"), lwd=1, lty=c(NA, NA, NA),  
        pch=c(4,1, 2) )
```


Nodo->Cola: Eliminar

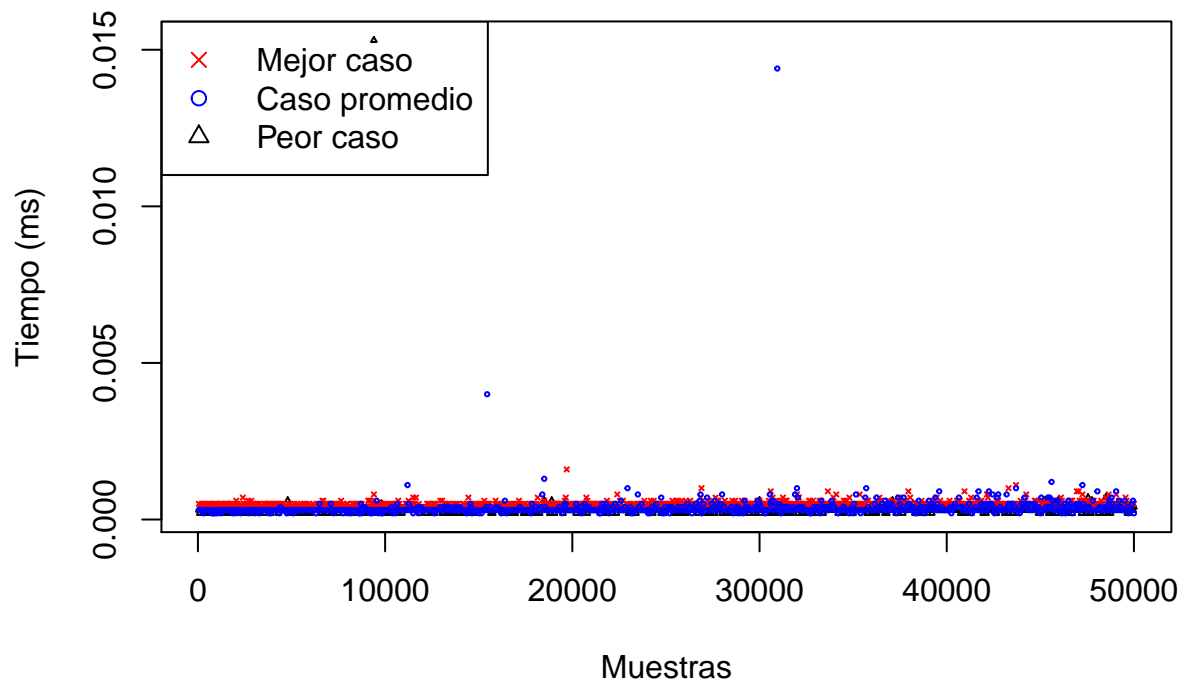


Al ser una estructura de memoria dinámica, los tiempos de inserción no varían significativamente por la cantidad de elementos que contiene ésta. Sin embargo, dada la naturaleza de las Colas, se debe destruir la estructura para navegar dentro de ella, por lo tanto, para buscar un valor o eliminar uno que no se encuentre al final de esta, se crea un vector auxiliar, que contenga todos los datos menos el eliminado, y se crea una nueva cola basada en nodos, reinicializándola e insertando todos los valores de la anterior.

Lista Enlazada (Lista basada en Nodos):

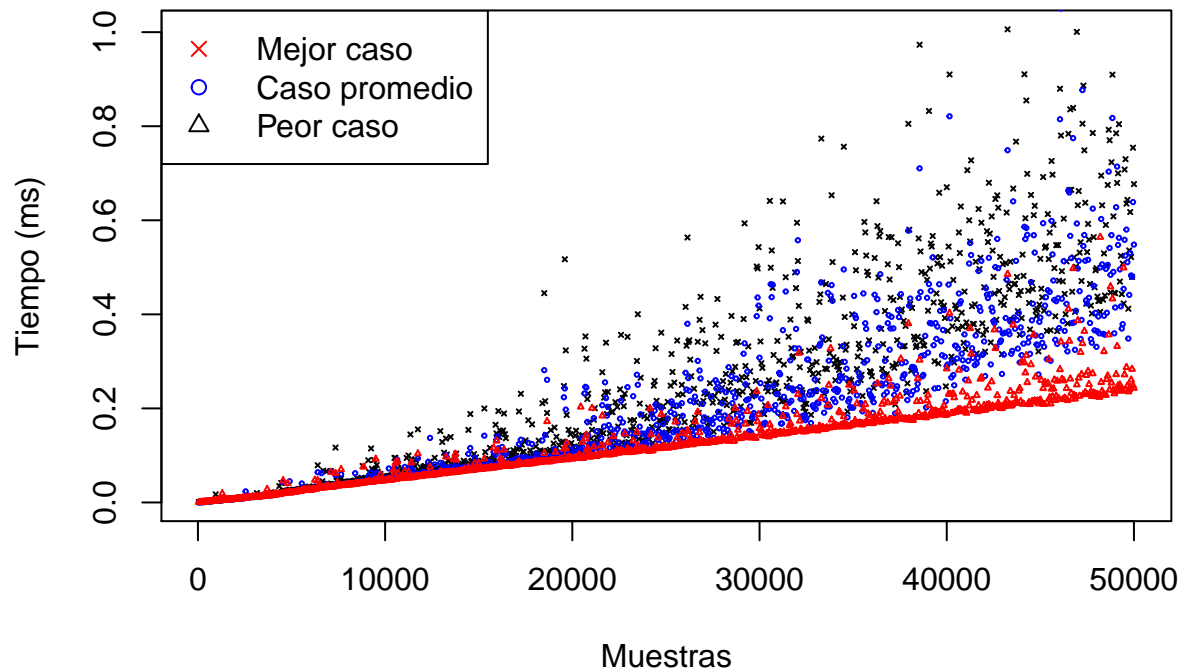
```
Nlista<-read.csv("~/Proyecto 1/Nlista_t.txt")
plot(Nlista$Muestras,Nlista$Insertar1,col="black", pch = 2, cex = 0.3,
     xlab = "Muestras", ylab = "Tiempo (ms)")
title("Nodo->Lista: Insertar")
points(Nlista$Muestras,Nlista$Insertar0,col="red", pch = 4, cex = 0.3)
points(Nlista$Muestras,Nlista$InsertarM,col="blue", pch = 1, cex = 0.3)
View(Nlista)
legend( x="topleft",
       legend=c("Mejor caso", "Caso promedio","Peor caso"),
       col=c("red","blue","black"), lwd=1, lty=c(NA, NA, NA),
       pch=c(4,1, 2) )
```

Nodo->Lista: Insertar



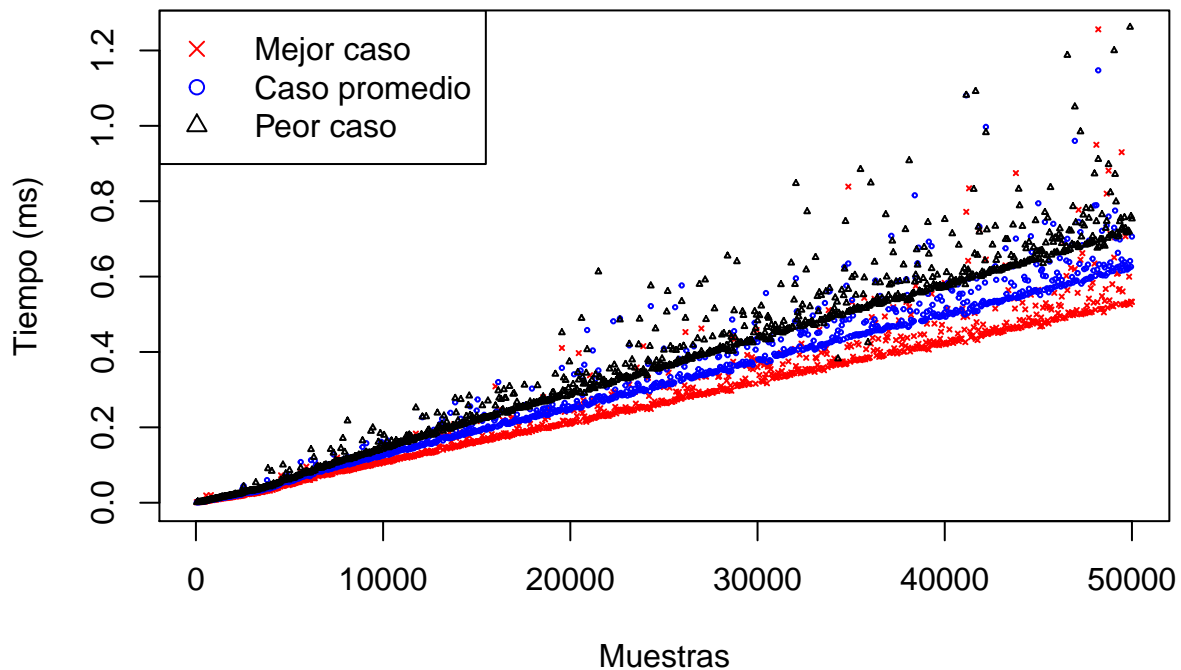
```
#-----  
plot(Nlista$Muestras,Nlista$Buscar0,col="black", xlab = "Muestras",  
      ylab = "Tiempo (ms)", pch = 4, cex = 0.3)  
title("Nodo->Lista: Buscar")  
points(Nlista$Muestras,Nlista$BuscarM,col="blue", pch = 1, cex = 0.3)  
points(Nlista$Muestras,Nlista$BuscarI,col="red", pch = 2, cex = 0.3)  
legend( x="topleft",  
        legend=c("Mejor caso", "Caso promedio","Peor caso"),  
        col=c("red","blue","black"), lwd=1, lty=c(NA, NA, NA),  
        pch=c(4,1, 2) )
```

Nodo→Lista: Buscar



```
#-----  
plot(Nlista$Muestras,Nlista$Eliminar0,col="red", xlab = "Muestras",  
      ylab = "Tiempo (ms)", pch = 4, cex = 0.3)  
title("Nodo->Lista: Eliminar")  
points(Nlista$Muestras,Nlista$EliminarM,col="blue", pch = 1, cex = 0.3)  
points(Nlista$Muestras,Nlista$EliminarI,col="black", pch = 2, cex = 0.3)  
legend( x="topleft",  
        legend=c("Mejor caso", "Caso promedio","Peor caso"),  
        col=c("red","blue","black"), lwd=1, lty=c(NA, NA, NA),  
        pch=c(4,1, 2) )
```

Nodo→Lista: Eliminar



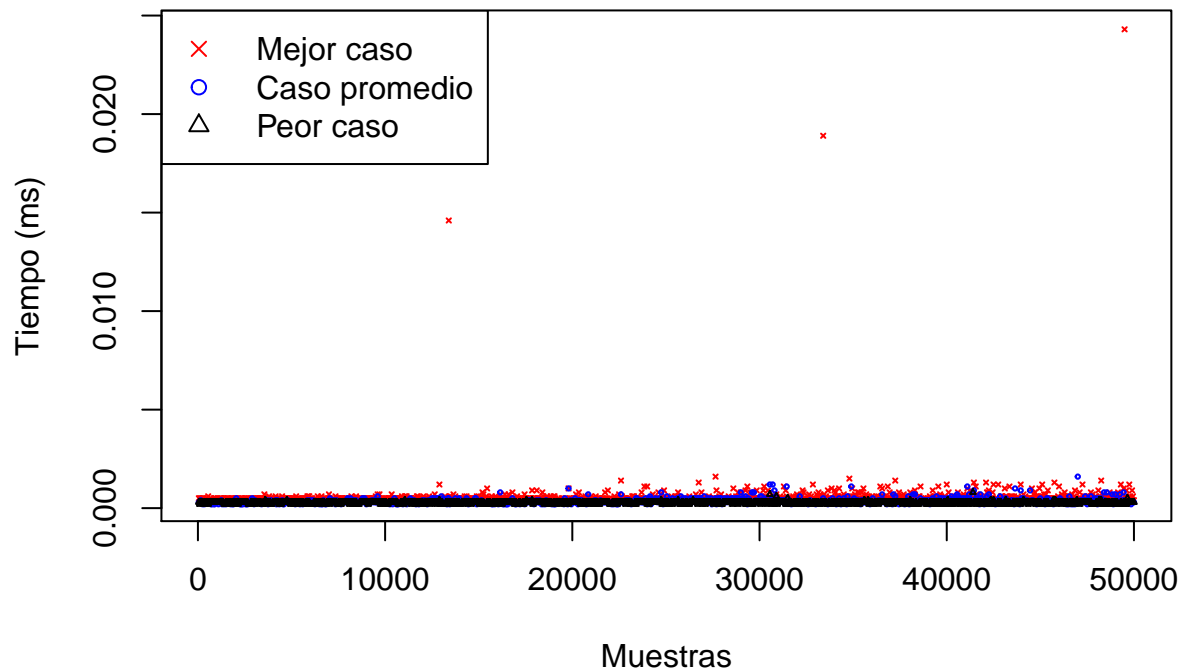
Al igual que las demás estructuras basadas en nodos, el tiempo de inserción es prácticamente independiente con respecto a la cantidad de elementos que contiene la estructura. En el caso de la búsqueda, se observa una dispersión importante en todos los casos menos el mejor.

Lista Enlazada (Pila basada en Nodos:

```
Npila<-read.csv("~/Proyecto 1/Npila_t.txt")

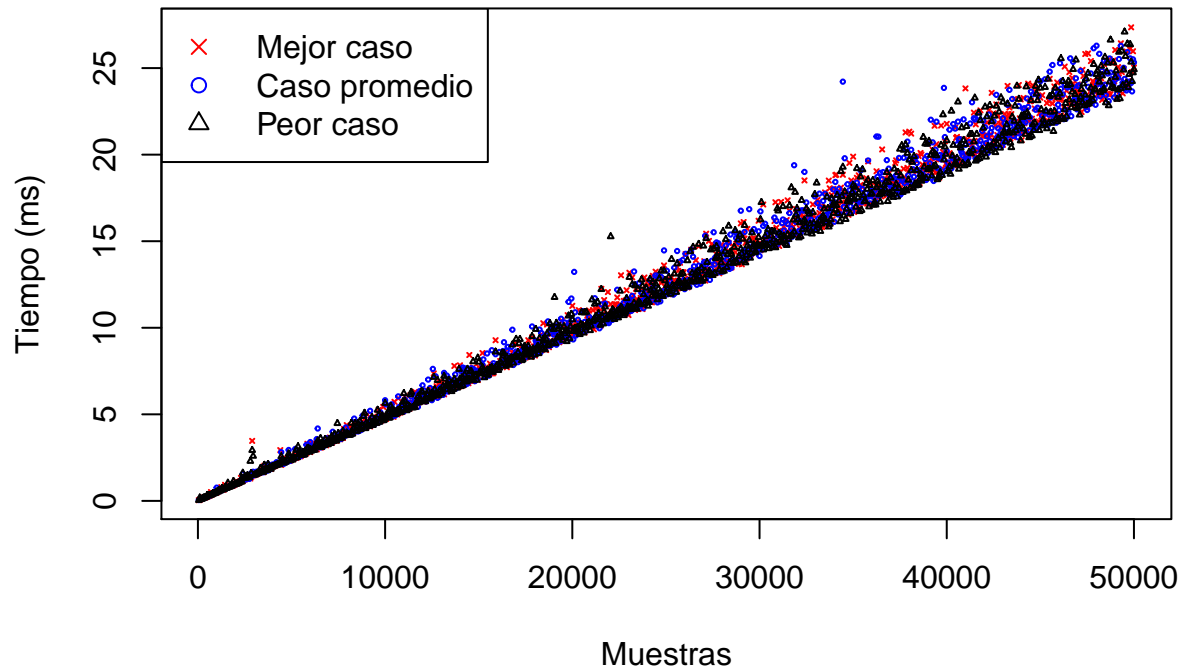
plot(Npila$Muestras,Npila$Insertar0,col="red", xlab = "Muestras",
     ylab = "Tiempo (ms)", pch = 4, cex = 0.3)
title("Nodo→Pila: Insertar")
points(Npila$Muestras,Npila$InsertarM,col="blue", pch = 1, cex = 0.3)
points(Npila$Muestras,Npila$Insertar1,col="black", pch = 2, cex = 0.3)
legend( x="topleft",
       legend=c("Mejor caso", "Caso promedio","Peor caso"),
       col=c("red","blue","black"), lwd=1, lty=c(NA, NA, NA),
       pch=c(4,1, 2) )
```

Nodo→Pila: Insertar



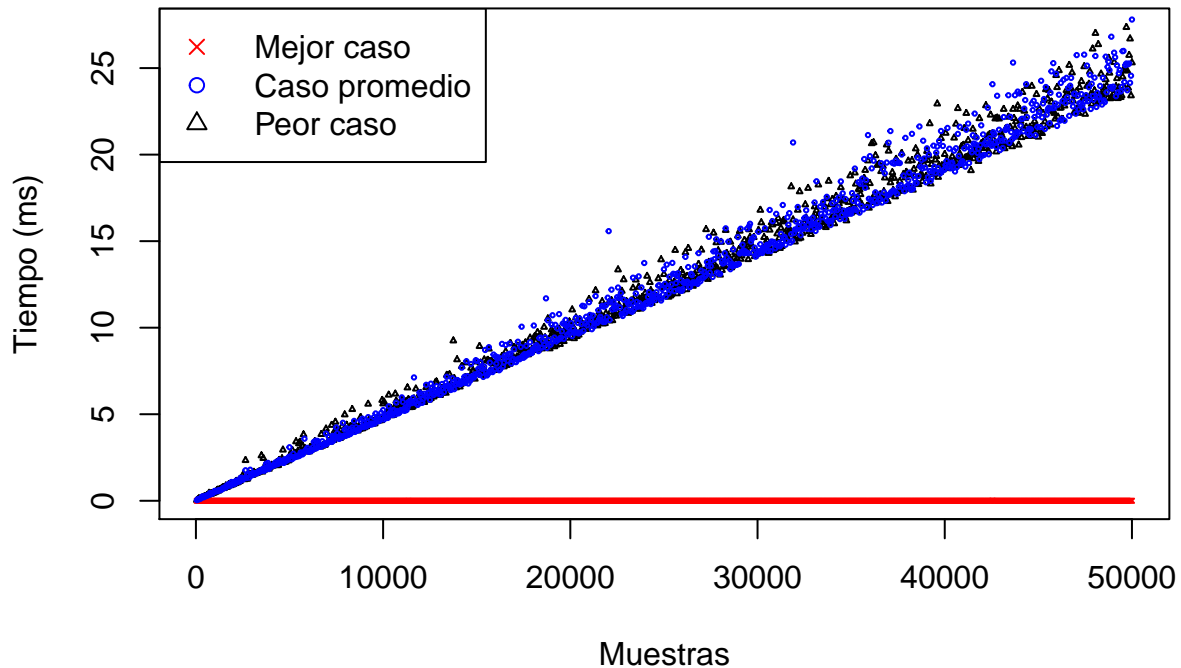
```
#-----  
plot(Npila$Muestras,Npila$Buscar0,col="red", xlab = "Muestras",  
      ylab = "Tiempo (ms)", pch = 4, cex = 0.3)  
title("Nodo->Pila: Buscar")  
points(Npila$Muestras,Npila$BuscarM,col="blue", pch = 1, cex = 0.3)  
points(Npila$Muestras,Npila$Buscar1,col="black", pch = 2, cex = 0.3)  
legend( x="topleft",  
        legend=c("Mejor caso", "Caso promedio","Peor caso"),  
        col=c("red","blue","black"), lwd=1, lty=c(NA, NA, NA),  
        pch=c(4,1, 2) )
```

Nodo->Pila: Buscar



```
#-----  
plot(Npila$Muestras,Npila$Eliminar1,col="black", pch = 2, cex = 0.3,  
      xlab = "Muestras", ylab = "Tiempo (ms)")  
title("Nodo->Pila: Eliminar")  
points(Npila$Muestras,Npila$Eliminar0,col="red", pch = 4, cex = 0.3)  
points(Npila$Muestras,Npila$EliminarM,col="blue", pch = 1, cex = 0.3)  
  
legend( x="topleft",  
        legend=c("Mejor caso", "Caso promedio","Peor caso"),  
        col=c("red","blue","black"), lwd=1, lty=c(NA, NA, NA),  
        pch=c(4,1, 2) )
```

Nodo→Pila: Eliminar



Se observa un comportamiento similar al de las colas basadas en Nodos. En las 3 operaciones probadas.

Conclusiones

- Las ventajas de las estructuras implementadas a base de arreglos es que se puede acceder a elementos aleatorios dentro de ésta de una forma rápida, complejidad $O(n)$, donde "n" es el índice del elemento.
- Tanto las pilas y colas ofrecen gran rapidez para acceder a su primer elemento insertado (en el caso de las colas) y a su último elemento insertado (en el caso de las pilas). Independientemente de su tamaño, complejidad $O(n)$.
- Implementar estructuras basadas en nodos permiten que el tiempo de inserción sea mínimo, diminuto en comparación con sus contrapartes de arreglos/vectores.
- El acceso a datos aleatorios dentro de un arreglo permite que las operaciones de búsqueda, eliminación y ordenamiento sean mucho más eficientes. Sin embargo, las colas y pilas perderían su significado, convirtiéndose en simples "arreglos"/"vectores".
- Es inviable un eficiente ordenamiento en estructuras basadas en nodos. A menos que se creen 2 punteros a nodos auxiliares para que se puedan hacer swap entre sus valores.
- Para cantidades de datos pequeñas (<1000000), convendría trabajar con vectores, ya que se puede declarar un vector mucho más grande de la cantidad de elementos que este esperase. Obteniendo una máxima complejidad de $O(n)$ en cualquier operación.

Observaciones

- Al inicio, se generó un dataset con números pseudo-aleatorios truncados a 50.000, pero se detectaron varios valores repetidos, lo que potencialmente reduciría la validez del estudio.
- Se halló una diferencia en el tiempo de ejecución (del código completo) bastante grande entre los integrantes del grupo:
 - i7 10750h (6 núcleos físicos, 12 lógicos @ 4.3 Ghz) + 16 Gb ram. 3 minutos.
 - i7 8xxxh (6 núcleos físicos, 12 lógicos @ 4.3 Ghz) + 16 Gb ram. >5 minutos.

Cabe resaltar que, en la primera PC, se usó Visual Studio 2019. En cambio, en la segunda se usó Visual Code.