Análisis de Eficiencia de Algoritmos

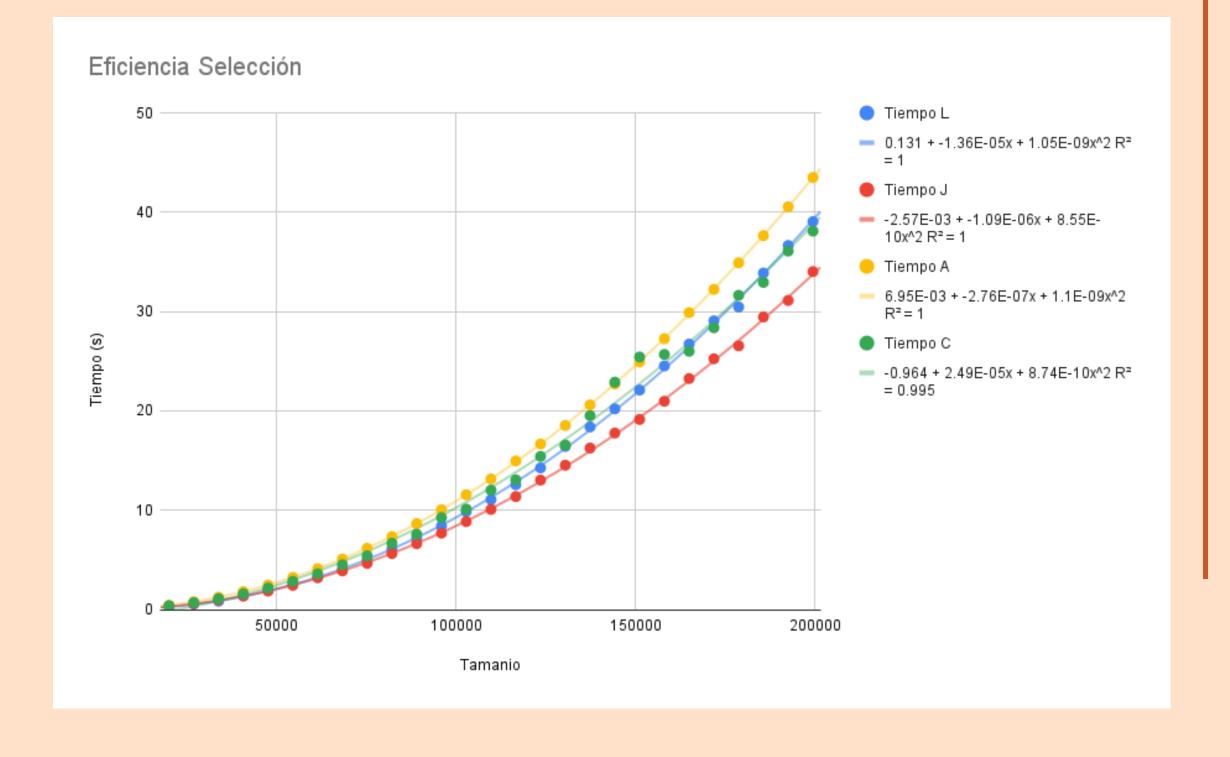
JOAQUÍN ARCILA PÉREZ LAURA LÁZARO SORALUCE CRISTÓBAL MERINO SÁEZ ÁLVARO MOLINA ÁLVAREZ

ÍNDICE

- I. Introducción
- II. Desarrollo
 - 1. Algoritmo de selección
 - 2. Algoritmo de inserción
 - 3. Algoritmo de heapsort
 - 4. Algoritmo de quicksort
 - 5. Algoritmo de Floyd
 - 6. Algoritmo de Hanoi
- III. Comparaciones
- IV. Conclusión

INTRODUCCIÓN

DESARROLLO Algoritmo de selección



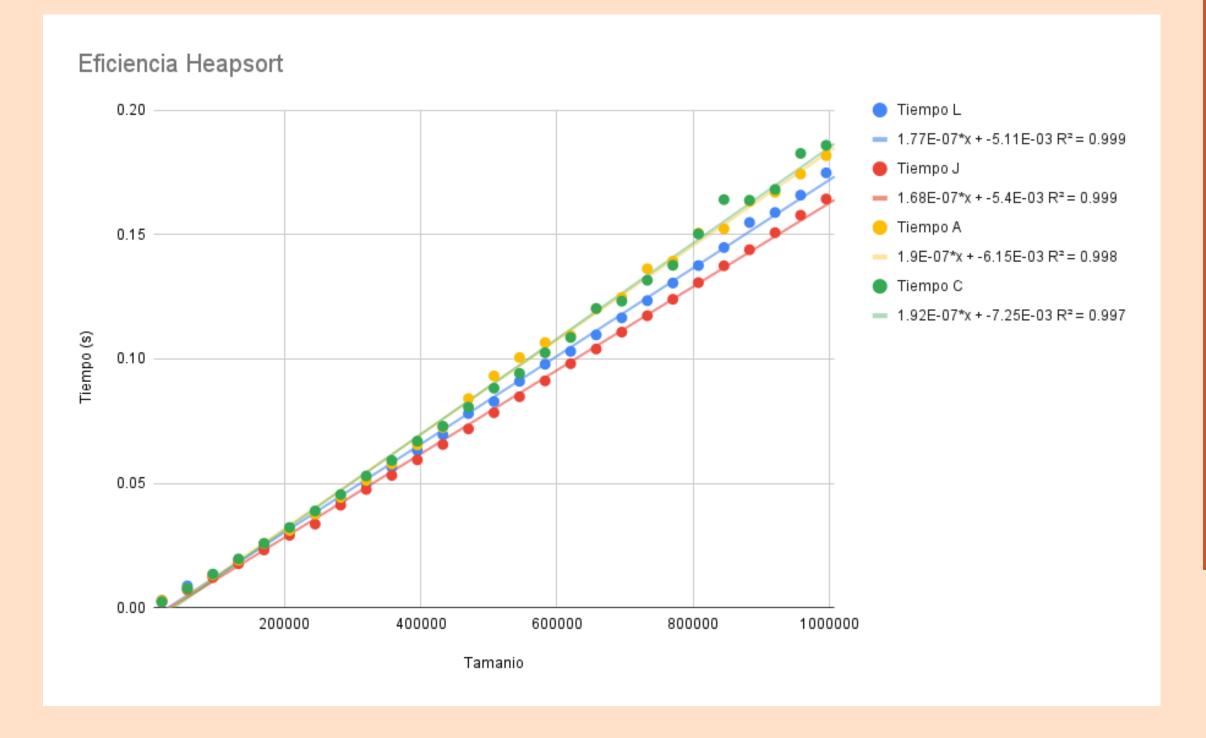
```
void seleccion(int T[], int num_elem)
seleccion_lims(T, 0, num_elem);
static void seleccion_lims(int T[], int inicial, int
final)
int i, j, indice_menor;
int menor, aux;
for (i = inicial; i < final - 1; i++) {
indice_menor = i;
menor = T[i];
for (j = i; j < final; j++)
if (T[j] < menor) {</pre>
indice_menor = j;
menor = T[j];
aux = T[i];
T[i] = T[indice_menor];
T[indice_menor] = aux;
```

DESARROLLO Algoritmo de inserción

Eficiencia Inserción 40 🔵 Tiempo L 0.576 + -2.06E-05x + 9.06E-10x^2 R² Tiempo J 0.151 + -5.51E-06x + 7.32E-10x^2 R² 30 Tiempo A 0.0169 + -6.04E-07x + 7.99E-10x^2 R² Tiempo C 20 -0.0482 + 1.47E-06x + 7.95E-10x^2 R² = 0.99910 100000 150000 200000 Tamanio

```
inline static void insercion(int T[], int
num_elem)
insercion_lims(T, 0, num_elem);
static void insercion_lims(int T[], int inicial, int
final)
int i, j;
int aux;
for (i = inicial + 1; i < final; i++) {
j = i;
while ((T[j] < T[j-1]) \&\& (j > 0)) {
aux = T[j];
T[j] = T[j-1];
T[j-1] = aux;
```

DESARROLLO Algoritmo de heapsort



```
static void heapsort(int T[], int num_elem)
int i;
for (i = num_elem/2; i >= 0; i--)
reajustar(T, num_elem, i);
for (i = num_elem - 1; i >= 1; i--)
int aux = T[0];
T[O] = T[i];
T[i] = aux;
reajustar(T, i, 0);
static void reajustar(int T[], int num_elem, int
int j;
int v;
v = T[k];
bool esAPO = false;
while ((k < num_elem/2) && !esAPO)
j = k + k + 1;
if ((j < (num_elem - 1)) && (T[j] < T[j+1]))
j++;
if (\lor > = T[i])
esAPO = true;
T[k] = T[j];
k = j;
T[k] = v;
```

DESARROLLO Algoritmo de quicksort

Eficiencia Quicksort 0.15 Tiempo L 1.18E-07*x + -2.95E-03 R² = 0.999 Tiempo J 1.08E-07*x + -2.23E-03 R² = 0.999 Tiempo A 1.33E-07*x + -2.89E-03 R² = 0.999 Tiempo C 0.10 1.22E-07*x + -2.9E-03 R² = 0.999 0.05 200000 400000 600000 800000 1000000

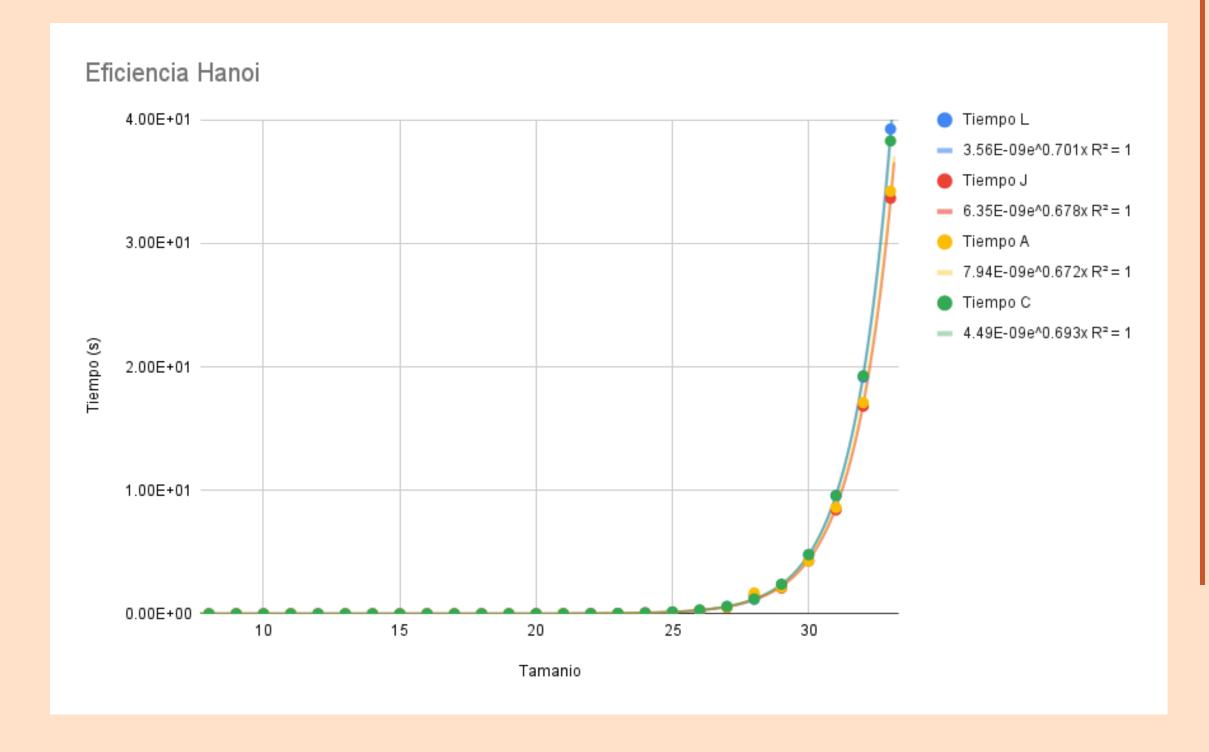
```
inline void quicksort(int T[], int num_elem)
{ quicksort_lims(T, 0, num_elem); }
static void quicksort_lims(int T[], int inicial, int final)
int k;
if (final - inicial < UMBRAL_QS) {</pre>
insercion_lims(T, inicial, final);
} else {
dividir_qs(T, inicial, final, k);
quicksort_lims(T, inicial, k);
quicksort_lims(T, k + 1, final); };
static void dividir_qs(int T[], int inicial, int final, int & pp)
int pivote, aux;
int k=inicial, l=final;
pivote = T[inicial];
do k++; while ((T[k] \le pivote) && (k \le final-1));
do l--; while (T[l] > pivote);
while (k < l) {
aux = T[k];
T[k] = T[l];
T[l] = aux;
do k++; while (T[k] <= pivote);
do l--; while (T[l] > pivote); };
aux = T[inicial];
T[inicial] = T[l];
T[l] = aux;
pp = l;
```

DESARROLLO Algoritmo de Floyd

Eficiencia FLoyd 20 0.0988 + -7.24E-04x + 1.21E-06x^2 + 4.23E-09x^3 R2 = 1 Tiempo J 0.038 + -3.12E-04x + 6.54E-07x^2 + 3.18 $E-09x^3 R^2 = 1$ 15 Tiempo A 1.06 + -6.5E-03x + 9.61E-06x^2 R² = 0.996 Tiempo C 1.24 + -7.44E-03x + 1.05E-05x^2 R² = 0.996 250 500 1000 1250 1500 Tamanio

```
void Floyd(int **M, int dim)
{
  for (int k = 0; k < dim; k++)
  for (int i = 0; i < dim;i++)
  for (int j = 0; j < dim;j++)
  {
  int sum = M[i][k] + M[k][j];
  M[i][j] = (M[i][j] > sum) ? sum : M[i][j];
  }
}
```

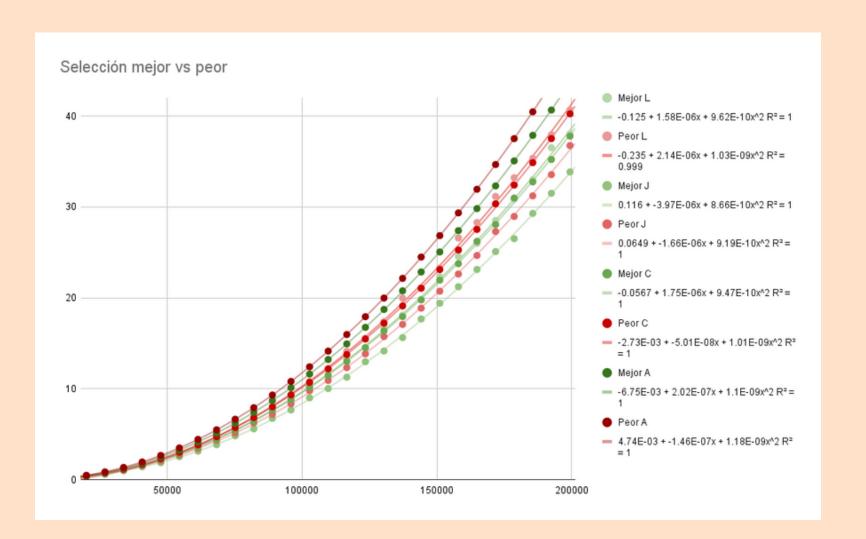
DESARROLLO Algoritmo de Hanoi



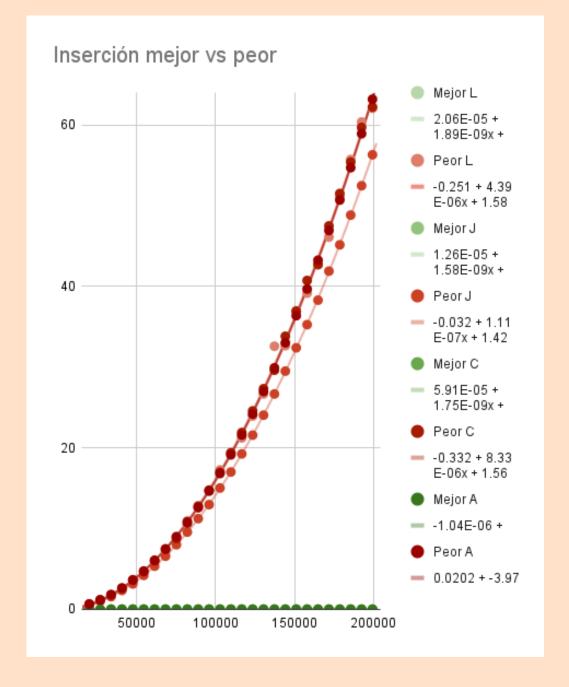
```
void hanoi (int M, int i, int j)
{
  if (M > 0)
  {
    hanoi(M-1, i, 6-i-j);
    //cout << i << " -> " << j << endl;
    hanoi (M-1, 6-i-j, j);
  }
}</pre>
```

COMPARACIONES Mejor vs Peor Caso

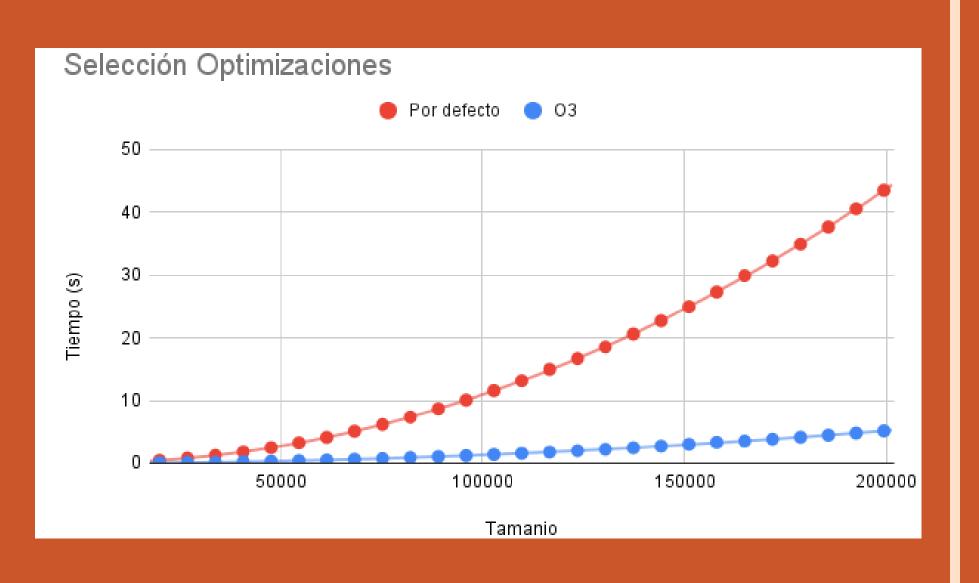
Selección:

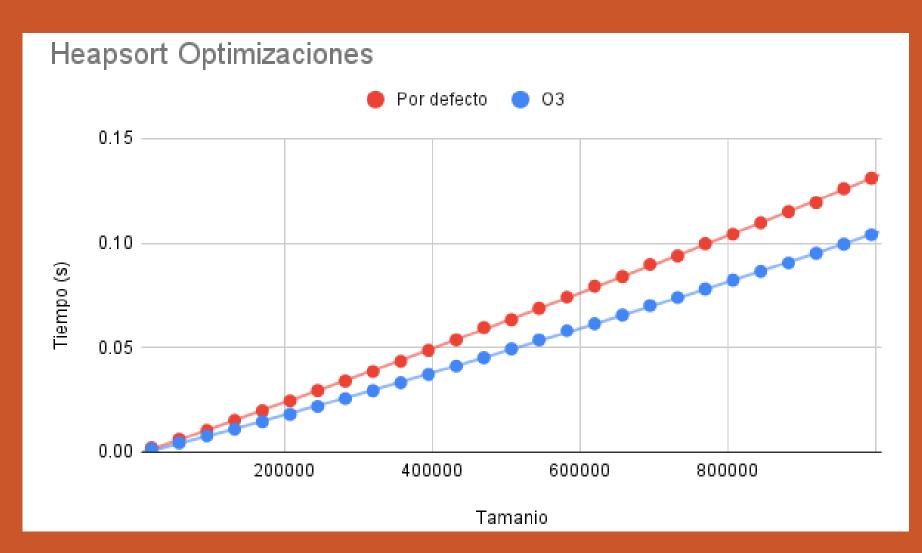


Inserción:

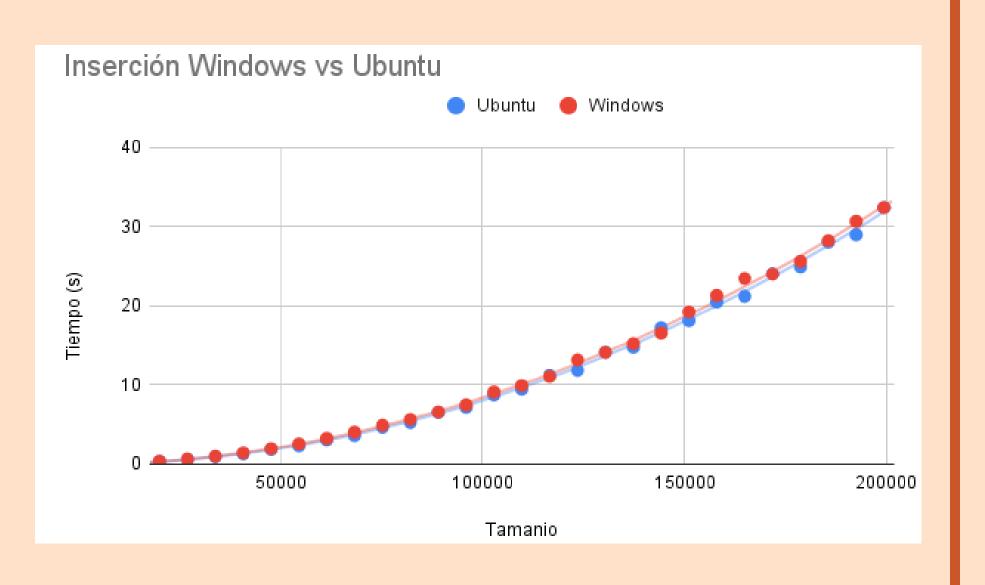


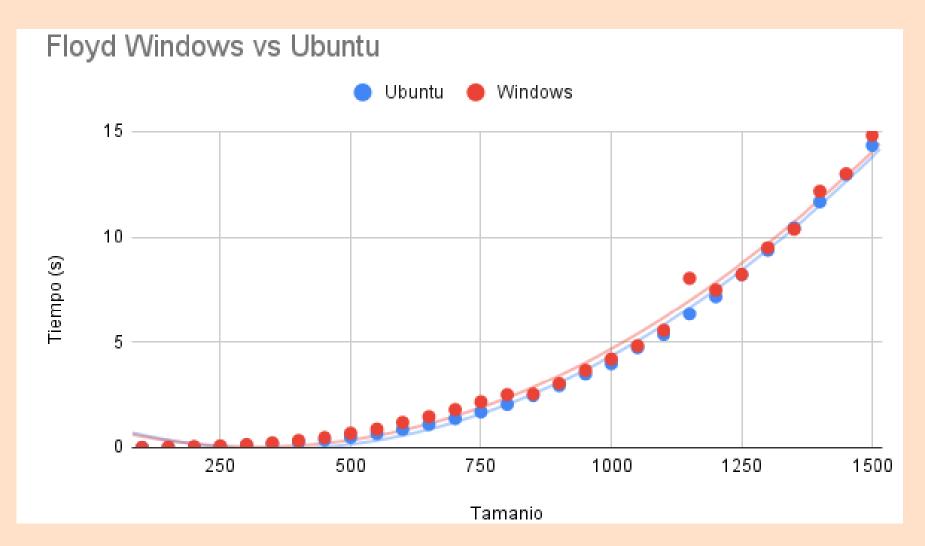
COMPARACIONES Optimizaciones



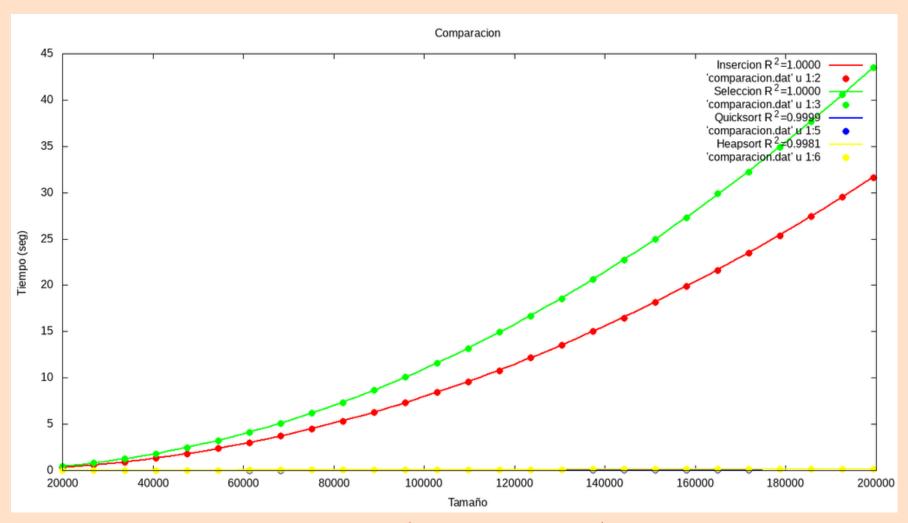


COMPARACIONES Windows vs Ubuntu

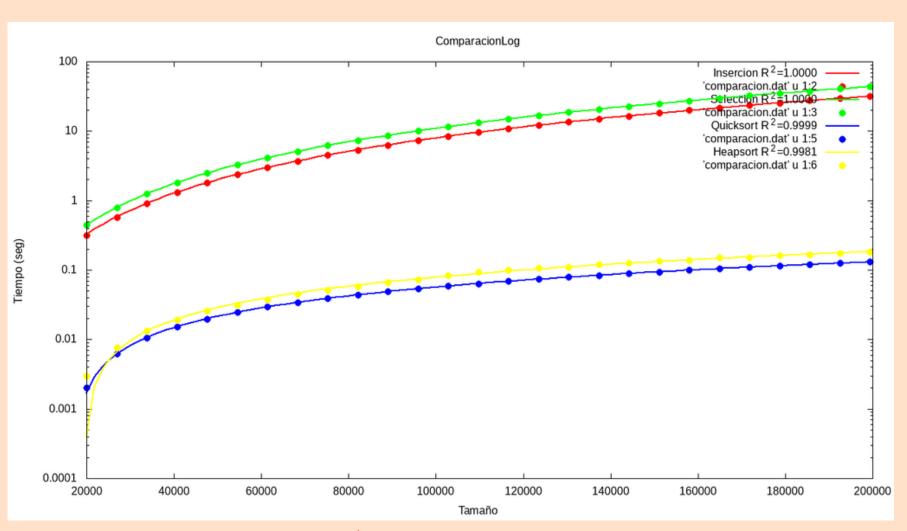




COMPARACIONES Algoritmos



Escala Normal



Escala Logarítmica

CONCLUSIÓN

- -Mayor importancia del orden de eficiencia que de otros factores de ejecución
- -Poca diferencia entre la ejecución en sistemas operativos distintos
- -Pequeña diferencia entre el peor y el mejor caso de selección
- -Gran diferencia entre el peor y el mejor caso de inserción

MUCHAS GRACIAS