Complejidad de los ejercicios del Taller 5

1. Suma de los elementos de un arreglo

1.1 Copiar el código en Word

```
public static int[] insertionSort (int[] array){
      for(int a=0;a<array.length;a++){
         int menor= a;
         for(int b=a+1;b<array.length;b++){
            if(array[b]< array[menor]){
               menor= b;
            }
         }
         int aux= array[a];
         array[a]= array[menor];
         array[menor]= aux;
      }
      return array;
   }
```

1.2 Identificar quién es el tamaño del problema (llamado también "n")

El tamaño del problema son los elementos que me falta por organizar en el arreglo.

1.3 Etiquetar cuánto se demora cada línea

```
public static int[] insertionSort (int[] array){
      for(int a=0;a<array.length;a++){ // c_1+c_2(n+1)+c_3
         int menor= a; // c_4n
         for(int b=a+1;b<array.length;b++){ //c_5+c_6(n(n+1))/2+c_7
            if(array[b]< array[menor]){ //c_8*(n(n+1))/2
               menor= b; //c_9*(n(n+1))/2
            }
         }
         int aux= array[a]; //c_10*(n(n+1))/2
         array[a]= array[menor]; //c_11*(n(n+1))/2
         array[menor]= aux; {//c_12*(n(n+1))/2
      }
      return array; //c_13
   }
```

1.4 Resolver la ecuación con Wolfram Alpha

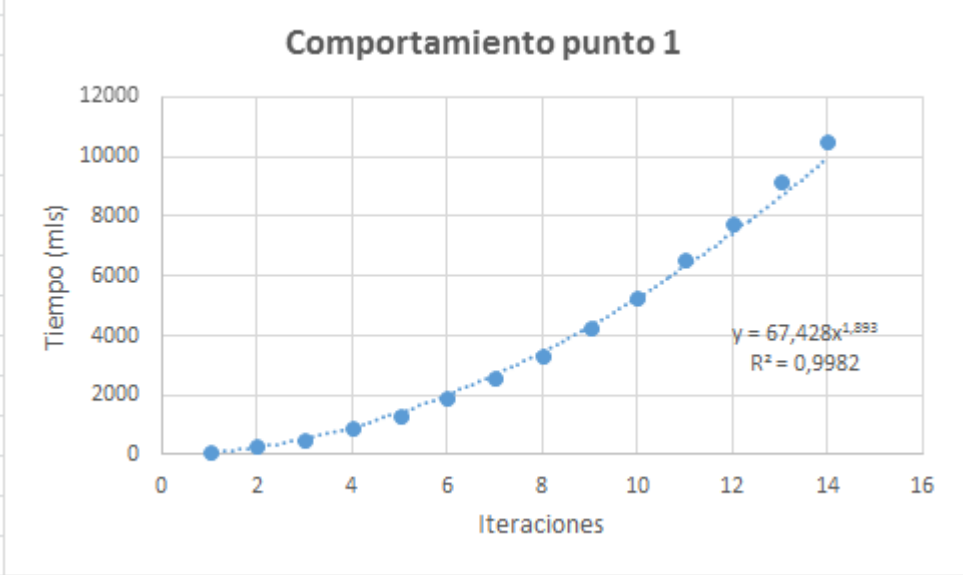
T(n) = c1+c2(n+1) + c3+ c5+c6*(n(n+1))/2 +c7+c8*(n(n+1))/2+ c9*(n(n+1))/2+ c10*(n(n+1))/2+ c11*(n(n+1))/2+ c12*(n(n+1))/2+c13

T(n) es O (c6*n^2 +n) Regla de la suma

T(n) es O (n^ 2) Regla del producto

1.5 Explicar en palabras

La complejidad asintótica (es decir, para valores grandes de n) para el peor de los casos (es decir, en el que el algoritmo hace más operaciones) para el algoritmo de organizar los elementos de un arreglo usando ciclos for es O(n^2). A continuación veremos la gráfica que ilustra dicho resultado.



2. Sumar los números de un arreglo

2.1 Copiar el código en Word

```
public static int suma(int[] a){
    int suma = 0; // c_1
    for(int i = 0; i < a.length; i++) // c2 + sum c3, i=0 to n
        suma += a[i];  //sum c4, i=0 to n-1
    return suma; //c5
 }
```

2.2 Identificar quién es el tamaño del problema (llamado también “n”)

El tamaño del problema es el número de elementos del arreglo.

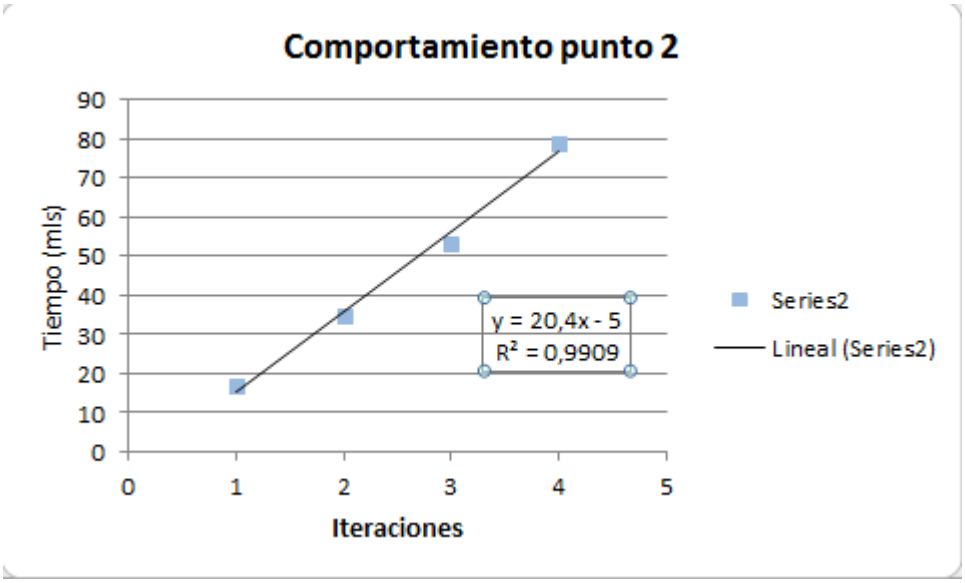
2.3 Etiquetar cuánto se demora cada línea

```
public static int suma(int[] a){
    int suma = 0; // c_1
    for(int i = 0; i < a.length; i++) // c2 + sum c3, i=0 to n
        suma += a[i];  //sum c4, i=0 to n-1
    return suma; //c5
 }
```

2.4 Resolver la ecuación con Wolfram Alpha

T(n) = c1 + c2 + c3(n+1) + c4n + c5
T(n) es O(c3n) Regla de la suma
T(n) es O(n) Regla del producto

2.7 Explicar en palabras
La complejidad asintótica (es decir, para valores grandes de n) para el peor de los casos (es decir, en el que el algoritmo hace más operaciones) para el algoritmo que usa ciclo for para sumar los elementos de un arreglo es O (n). A continuación veremos la gráfica que ilustra dicho resultado.



3. Tablas de multiplicar
3.1 Copiar el código en Word

```
public static void mul (int num){
    for(int i=1;i<=num;i++){
        for(int j=1;j<=num;j++){
            //System.out.println(i +" * "+ j + " = "+ i*j);
        }
    }
}
```

3.2 Identificar quién es el tamaño del problema (llamado también "n")

El tamaño del problema es n^2, siendo n el número digitado
3.3 Etiquetar cuánto se demora cada línea

```
public static void mul (int num){
    for(int i=1;i<=num;i++){ //c_1+ sum c_2, i=0 to n+1
        for(int j=1;j<=num;j++){ //c_4+ sum c_5, i=0 to n+1
```

```
            //System.out.println(i +" * "+ j + " = "+ i*j);//c_6
        }
    }
}
```

3.4 Resolver la ecuación con Wolfram Alpha

T(n) = c_2*(n+2) +c_1+ c_3*(n+2) +c_4

T(n) = O(c_2*(n+2) +c_1+ c_3*(n+2) +c_4), por definición de O
T(n)= O(n*n), regla de la suma
T(n)= O(n^2)

3.7 Explicar en palabras

La complejidad asintótica (es decir, para valores grandes de n) para el peor de los casos (es decir, en el que el algoritmo hace más operaciones) para el algoritmo que calcula las tablas de multiplicar haciendo uso del ciclo for es O(n^2). A continuación veremos la gráfica que ilustra dicho resultado.