



## INFORME

### TAREA No. 1: PRELIMINARES

NICOLÁS DANIEL PERALTA SOPÓ  
DAVID RICARDO BERNAL

ANÁLISIS DE ALGORITMOS

INGENIERÍA DE SISTEMAS  
FACULTAD DE INGENIERÍA

BOGOTÁ D.C

1. Implemente el algoritmo para sumar dos números representados como listas de dígitos, en un método sumar que retorne el resultado.

Hecho. Ver código en la carpeta src.

2. Utilice el algoritmo del paso 1 para escribir un algoritmo que, dados dos números representados como listas de dígitos, los multiplique.

Hecho. Ver código en la carpeta src.

3. Analice ambos algoritmos temporal y espacialmente en el peor caso. Cerciórese de incluir todo el procedimiento en su informe.

Algoritmo de suma:

- Tamaño de la Entrada =  $n + m$ ; donde  $n = \text{numero1.length}$  &  $m = \text{numero2.length}$ .
- Operación Básica: '+' (Operación de Suma o Adición)
- $T(n) = k$ ; donde  $k = \max(n, m)$  que corresponde al número de veces  $k$  que se ejecuta el bucle (while para este caso).
- $S(n) = 7 + (n+1)$

Variable	Tamaño
i	1
j	1
carry	1
resul_tam	1
sumando1	1
sumando2	1
suma	1
resul	$n+1$

Algoritmo de Multiplicar\*:

(\*Supuesto: El número de cifras del multiplicando es mayor o igual al del multiplicador)

- Tamaño de la Entrada =  $n + m$ ; donde  $n = \text{numero1.length}$  &  $m = \text{numero2.length}$
- Operación Básica: + (Ya que este suma el multiplicando por el número de veces del dígito que corresponde al multiplicador[i] )

- $T(n) = n$
- $S(n) = 4 + (n+1) + (k+1)$

Variable	Tamaño
i	1
j	1
posición	1
dígito	1
semiProducto	n+1
producto	k+1

4. Mida el tiempo de ejecución de su algoritmo de multiplicación para instancias de tamaño 10, 100 y 1.000. Registre en el informe los resultados, junto con la descripción de la máquina utilizada.

Las mediciones del tiempo de ejecución correspondiente al algoritmo de multiplicar se realizan en una máquina que cuenta con las siguientes especificaciones:

- Procesador: Intel Core i5-4210U – 4 CPU's @ 1.70GHz
- Memoria: DDR3 8GB @ 800MHz
- Sistema Operativo: Microsoft Windows 10 Professional x64
- Máquina virtual: Java SE Runtime Environment 8 (v1.8.0\_211) 64-Bit

*Nota: Los datos anteriores fueron obtenidos con CPU-Z (freeware software).*

Debido a que el tiempo de ejecución del algoritmo no es constante, es decir que este tiempo varía cada vez que se ejecuta, aunque los datos de entrada no varíen. Es por ello que se registra el 5 distintos tiempos de ejecución por cada cambio del tamaño de las instancias que recibe el algoritmo, dichos se promedian para obtener un tiempo de ejecución promedio aproximado.

Dichos cálculos se realizarán usando el nanosegundo como unidad de tiempo debido a que en algunas ocasiones no se puede registrar el tiempo en milisegundos ya que su valor es inferior.

En nanosegundos:

- Para instancias de tamaño 10:

<terminated> SumarMultiplicar [Java Application] C:\Program Files\Java\jre1.8.0\_211\bin\javaw.exe (8/09/2020 01:06:54 PM – 01:06:54 PM)  
Instancia de tamaño 10  
El tiempo de ejecucion de la multiplicacion en nanosegundos: 235700.0

<terminated> SumarMultiplicar [Java Application] C:\Program Files\Java\jre1.8.0\_211\bin\javaw.exe (8/09/2020 01:07:49 PM – 01:07:49 PM)  
Instancia de tamaño 10  
El tiempo de ejecucion de la multiplicacion en nanosegundos: 199100.0

<terminated> SumarMultiplicar [Java Application] C:\Program Files\Java\jre1.8.0\_211\bin\javaw.exe (8/09/2020 01:08:18 PM – 01:08:18 PM)  
Instancia de tamaño 10  
El tiempo de ejecucion de la multiplicacion en nanosegundos: 206900.0

<terminated> SumarMultiplicar [Java Application] C:\Program Files\Java\jre1.8.0\_211\bin\javaw.exe (8/09/2020 01:08:47 PM – 01:08:47 PM)  
Instancia de tamaño 10  
El tiempo de ejecucion de la multiplicacion en nanosegundos: 205300.0

<terminated> SumarMultiplicar [Java Application] C:\Program Files\Java\jre1.8.0\_211\bin\javaw.exe (8/09/2020 01:09:35 PM – 01:09:35 PM)  
Instancia de tamaño 10  
El tiempo de ejecucion de la multiplicacion en nanosegundos: 211400.0

• Para instancias de tamaño 100:

<terminated> SumarMultiplicar [Java Application] C:\Program Files\Java\jre1.8.0\_211\bin\javaw.exe (8/09/2020 11:40:27 AM – 11:40:27 AM)  
Instancia de tamaño 100  
El tiempo de ejecucion de la multiplicacion en nanosegundos: 5469500.0

<terminated> SumarMultiplicar [Java Application] C:\Program Files\Java\jre1.8.0\_211\bin\javaw.exe (8/09/2020 11:43:26 AM – 11:43:26 AM)  
Instancia de tamaño 100  
El tiempo de ejecucion de la multiplicacion en nanosegundos: 5268600.0

<terminated> SumarMultiplicar [Java Application] C:\Program Files\Java\jre1.8.0\_211\bin\javaw.exe (8/09/2020 11:44:44 AM – 11:44:44 AM)  
Instancia de tamaño 100  
El tiempo de ejecucion de la multiplicacion en nanosegundos: 5039300.0

<terminated> SumarMultiplicar [Java Application] C:\Program Files\Java\jre1.8.0\_211\bin\javaw.exe (8/09/2020 11:46:02 AM – 11:46:02 AM)  
Instancia de tamaño 100  
El tiempo de ejecucion de la multiplicacion en nanosegundos: 6200200.0

<terminated> SumarMultiplicar [Java Application] C:\Program Files\Java\jre1.8.0\_211\bin\javaw.exe (8/09/2020 11:46:29 AM – 11:46:29 AM)  
Instancia de tamaño 100  
El tiempo de ejecucion de la multiplicacion en nanosegundos: 5322800.0

• Para instancias de tamaño 1000:

<terminated> SumarMultiplicar [Java Application] C:\Program Files\Java\jre1.8.0\_211\bin\javaw.exe (8/09/2020 11:47:53 AM – 11:47:53 AM)  
Instancia de tamaño 1000  
El tiempo de ejecucion de la multiplicacion en nanosegundos: 1.275565E8

<terminated> SumarMultiplicar [Java Application] C:\Program Files\Java\jre1.8.0\_211\bin\javaw.exe (8/09/2020 11:50:04 AM – 11:50:04 AM)  
Instancia de tamaño 1000  
El tiempo de ejecucion de la multiplicacion en nanosegundos: 1.33266E8

<terminated> SumarMultiplicar [Java Application] C:\Program Files\Java\jre1.8.0\_211\bin\javaw.exe (8/09/2020 11:50:52 AM – 11:50:52 AM)  
Instancia de tamaño 1000  
El tiempo de ejecucion de la multiplicacion en nanosegundos: 1.608997E8

<terminated> SumarMultiplicar [Java Application] C:\Program Files\Java\jre1.8.0\_211\bin\javaw.exe (8/09/2020 11:51:33 AM – 11:51:33 AM)  
Instancia de tamaño 1000  
El tiempo de ejecucion de la multiplicacion en nanosegundos: 1.299059E8

<terminated> SumarMultiplicar [Java Application] C:\Program Files\Java\jre1.8.0\_211\bin\javaw.exe (8/09/2020 11:52:38 AM – 11:52:38 AM)  
Instancia de tamaño 1000  
El tiempo de ejecucion de la multiplicacion en nanosegundos: 1.370306E8

5. Basándose en los resultados de los puntos 3 y 4, elabore una predicción del tiempo de ejecución de una instancia de tamaño 2.000. Incluya todos los detalles de cómo llegó al valor predicho en el informe.

En la predicción se utilizaron los valores de los tiempos de ejecución obtenidos en cada uno de los casos (instancias de tamaño 10, 100 y 1000) para realizar un análisis de regresión en cual determinar una relación entre las variables las cuales en este caso son: el tiempo de ejecución del algoritmo (variable dependiente) y el tamaño de la instancia (variable independiente).

Para dicho proceso, se tabularon los datos obtenidos en Excel (ver archivo adjunto) y posteriormente se realizó una gráfica de dispersión en cual se puedan visualizar la relación entre ellos. Usando esta gráfica y las herramientas que ofrece Excel, se realizó una regresión potencial para determinar una función que relacione las dos variables y así proyectar el comportamiento de la variable dependiente (el tiempo de ejecución) en distintos escenarios:

$$y = 8329,5x^{1,4067}$$

Con esta relación, podemos realizar la predicción del tiempo de ejecución para un escenario en cual el tamaño de la instancia (variable X) es de 2000:

$$y = 8329,5(2000)^{1,4067}$$

$$y = 3,666 \times 10^8 ns$$

Como se puede observar, el tiempo de ejecución teórico aproximado para el algoritmo de multiplicación con una instancia de tamaño 2000 es de  $3,666 \times 10^8$  nanosegundos.

6. Realice la medición en la misma máquina utilizada para el punto 4.

En nanosegundos:

```
<terminated> SumarMultiplicar [Java Application] C:\Program Files\Java\jre1.8.0_211\bin\javaw.exe (8/09/2020 10:01:20 AM - 10:01:21 AM)
Instancia de tamaño 2000
El tiempo de ejecucion de la multiplicacion en nanosegundos: 3.509797E8
```

7. Redacte un buen par de párrafos de conclusiones en los cuales reflexione sobre los resultados y cómo llegó a ellos.

En la predicción del tiempo de ejecución del algoritmo para una instancia de tamaño 2000, se calculó a partir de una regresión el valor teórico del tiempo ejecución para dicha instancia y lograr así un valor aproximado al valor experimental obtenido mediante una función en el código. Sin embargo, el error porcentual entre dichos valores fue mayor al 100% lo cual indica que pudo haber sido influenciado por el número de muestras tomadas en cada uno de los casos y el modelo matemático. Una crítica común a este tipo de modelos de predicción matemática es que no son óptimos, ya que se genera una

falsa correlación entre los datos si el número de muestras tomadas es bajo. Esto ocasiona que el modelo pueda generalizar o predecir los datos de forma incorrecta por carecer de suficientes muestras iniciales, este fenómeno o efecto es conocido en otras áreas como **underfitting**.

El término underfitting, es perteneciente al campo del Machine Learning, donde se describe como la existencia de un exceso de generalización de un modelo en cuestión, lo cual conlleva a un modelo simplista que no tiene la capacidad de capturar anomalías, particularidades o dispersiones.

Aunque se aumentó el número de muestras tomadas para instancia, la disminución gradual del error porcentual era baja por lo que una forma de mitigar esto sería tomar muestras del tiempo ejecución para instancias que varían desde 1 hasta 2000 que es la variable de estudio. Al aumentar el número de muestras para el modelo matemático, se podría generar un mejor ajuste de la regresión y por ende un valor del tiempo de ejecución con mayor exactitud.