**Práctica 3: Algoritmo Clásico para el Producto de Polinomios**

1. ***Arquitectura.***

El análisis se desarrolla bajo un procesador con arquitectura Intel 64, implementación de Intel de la arquitectura x86-64 (la versión de 64-bits del conjunto de instrucciones de la arquitectura x86). Dicho procesador cuenta con doble núcleo y 4 subprocesos, una velocidad de reloj de 1,7 GHz y frecuencia máxima de 2,6Ghz.

Memoria Caché, Memoria RAM, Intel i5, …

1. ***Pruebas.***

El algoritmo clásico quedó:

Polinomio aux(this->gr\_ + p.gr\_);

aux.nulo();

for (int i = 0; i < aux.getTam(); i++)

for (int j = 0; j <= i && j < this->getTam(); j++) {

if ((i - j) < p.getTam())

aux.coef\_[i] += this->coef\_[j] \* p.coef\_[i – j]

}

}

Observando el algoritmo, tenemos que: aux.nulo() es de orden O(n), que el primer for es de orden O(2n + 1) y el segundo for es de orden O(n/2). Por lo que, el orden del algoritmo es O(n2) = O(n + (2n+1)n/2).

En la parte práctica hemos implementado una clase Tests que saca por fichero o pantalla los resultados del producto de polinomios. Para ello, se introduce el grado máximo a evaluar y el nº de pruebas que quieres que se realice para cada grado. De esta manera, se calcularía una media de los tiempos en milisegundos. Los tiempos obtenidos se muestran a continuación:

|  |  |
| --- | --- |
| ***Grado*** | ***Milisegundos*** |
| 0 | 0 |
| 100 | 2 |
| 200 | 6 |
| 300 | 13 |
| 400 | 22 |
| 500 | 35 |
| 600 | 51 |
| 700 | 69 |
| 800 | 90 |
| 900 | 115 |
| 1000 | 96 |
| 1100 | 113 |
| 1200 | 138 |
| 1300 | 164 |
| 1400 | 185 |
| 1500 | 249 |
| 1600 | 256 |
| 1700 | 274 |
| 1800 | 320 |
| 1900 | 353 |
| 2000 | 392 |

Ahora con ayuda del Excel y la tabla de valores anteriores, se ha obtenido una gráfica. Luego, se le ha aplicado una línea de tendencia polinómica de orden 2 que se muestra en color rojo.

Observando la gráfica, se ve claramente que datos teóricos y prácticos se asemejan.