

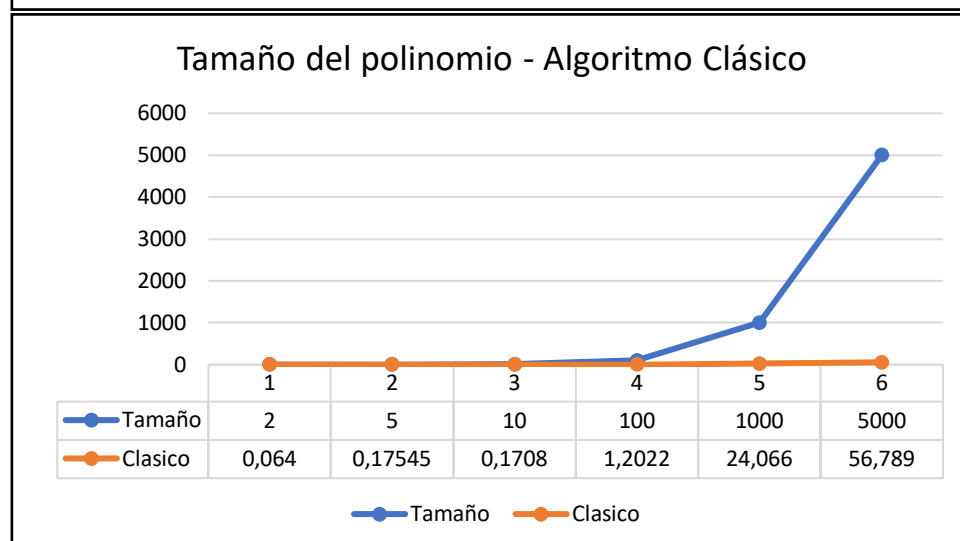
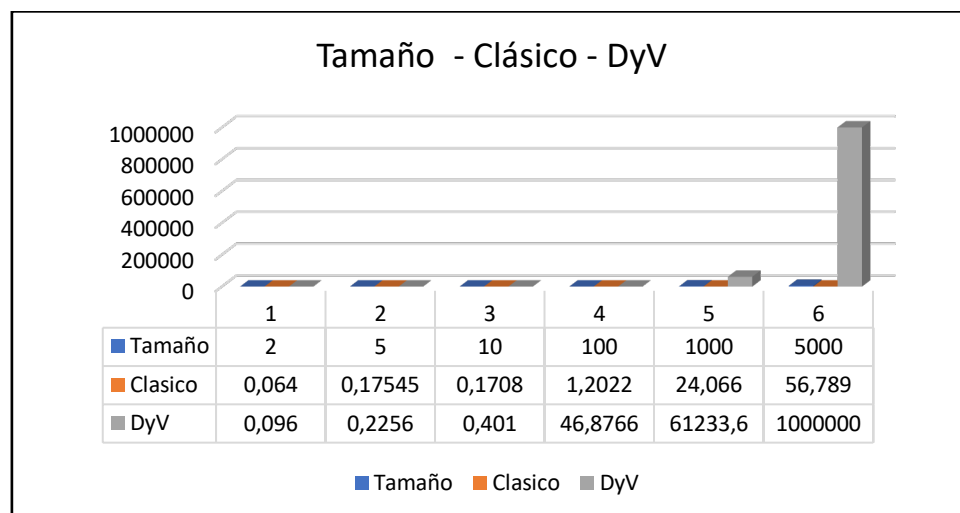
# Informe Diseño y Análisis de Algoritmos.

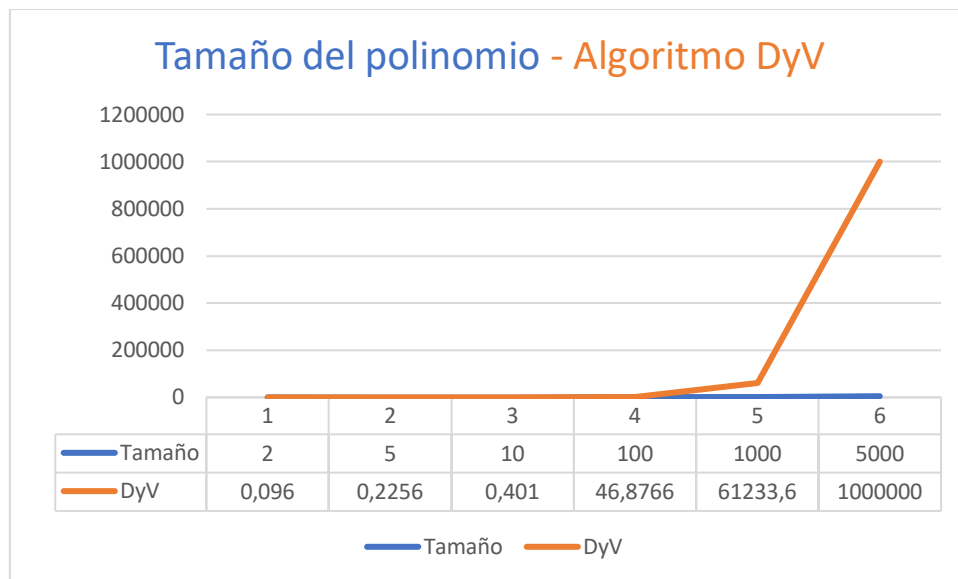
## Multiplicación de polinomios

En este informe analizaremos el tiempo que se tarda en multiplicar dos polinomios del mismo tamaño. Las pruebas se realizaron sobre una maquina cuya arquitectura consta de:

- Procesador: Intel Core i9-9900k 3.6 Ghz 8 cores 16 threads
- Memoria Ram: 16Gb

| Polinomio | 2     | 5      | 10     | 100     | 1000    | 5000 |
|-----------|-------|--------|--------|---------|---------|------|
| Clásico   | 0.064 | 0.1745 | 0.1708 | 1.2022  | 24.066  | -    |
| DyV       | 0.096 | 0.2256 | 0.401  | 46.8766 | 61233.6 | -    |





Como se puede apreciar, en esta ocasión se comprueba que el algoritmo clásico tarda menos tiempo que el algoritmo de DyV. El principal fallo se encuentra en la implementación del algoritmo, ya que para cada paso de la formula, se aplican numerosos pasos intermedios, para preparar los polinomios. Es decir, en pasos intermedios, podemos obtener un polinomio de la forma  $(3x^2 + 10)$ , entonces si el grado del polinomio es 3, significando que faltan los monomios  $(0x^2, 0x)$  por lo que gran cantidad de pasos intermedios y que aumentan la complejidad y el tiempo de resolución se los llevan estos métodos intermedios. Otro método que hay que aplicar es la rotación de polinomios, cuando en la formula se multiplica el polinomio por  $x$ , implicando tener que rodar todos los exponentes un grado arriba.

Además parte del problema de la implementación del programa viene por haber almacenado el vector en orden inverso a las posiciones del vector, es decir que el elemento de mayor grado se almacena en la primera posición del vector, para así llegar hasta la última posición donde se almacenaría el monomio de grado 0 siendo este un simple coeficiente