TP2 Complejidad Computacional

**Conclusiones Polinomios**:

En base a los test realizados, podemos observar como los métodos que tardan más tiempo son; **Multiplicaciones Sucesivas**, **Recursiva**, **Recursiva Par** y **Pow**. Y entre estos, el que tarda más tiempo de todos es el de ***Recursiva Par*** seguido por ***Multiplicaciones Sucesivas.***

Los demás métodos (**Prog. Dinamica**, **Mejorada**, **Pow**, **Horner**) oscilan en un tiempo de rendimiento similar.

También se probaron otros lotes. Y se pudo comprobar, que los métodos ***Recursiva*** y ***Recursiva*** Par, no llegan a procesar polinomios de grado igual o mayor a 10.000 en cambio los demás métodos si pueden. También para polinomios de grado 1.000.000 el método ***Multiplicaciones Sucesivas*** ya no sirve porque tarda demasiado (varios minutos). Los lotes de prueba usados indican que los métodos restantes (***Prog. dinámica***, ***Mejorada***, ***Pow*** y ***Horner***) andan bien incluso con polinomios de grado 10.000.000 tardando todos más o menos, alrededor de los 2 seg.

**Conclusiones Binomios:**

En base a los test realizados, podemos observar que el método que más tiempo demora es **obtenerTerminosTarta,** dicho método dispara su tiempo de ejecución a partir del grado 30 y obtiene los valores de los términos hasta el grado 70 aproximadamente.

Podemos observar que el tiempo de ejecución del método **resolverBinomioEstatico** se dispara conforme se aumenta el grado del binomio.

A diferencia de su versión estática, el método **resolverBinomio** presenta un incremento en el tiempo de ejecución notablemente menor. Este método resuelve binomios de distintos grados con un aumento mínimo en el tiempo de ejecución.

Al utilizar los métodos de polinomios para resolver los binomios de Newton, vemos que los resultados son parecidos, siendo **evaluarProgDinamica** y **evaluarMejorada** los métodos más eficientes a la hora de resolverlos.

**Observaciones finales:**

**EvaluarProgDinamica** y **evaluarMejorada** son los métodos más optimizados, ya que utilizan la programación dinámica, y por esto tienen una menor complejidad computacional. Es por esto que se ve una diferencia notable en el tiempo de ejecución de las pruebas. En cambio métodos como  **evaluarRecursiva** que no usan la programación dinámica, y en cambio usan la recursividad, tienen una mayor complejidad computacional debido a eso.

Por último, es notable la diferencia en tiempo de ejecución entre un algoritmo de complejidad computacional N a comparación de uno de N^2, por ejemplo en el caso de **evaluarMSucesivas** (complejidad N^2), es notable como escala la diferencia en el tiempo de ejecución dependiendo del grado del polinomio, a diferencia de **evaluarProgDinamica** (complejidad N), donde apenas se nota la diferencia.

Adjuntos se encuentran las tablas de tiempos de ejecución donde se ven las gráficas que ayudan a notar la diferencia entre cada método.