



**UNIVERSIDAD  
SERGIO ARBOLEDA**

ESCUELA DE CIENCIAS EXACTAS E INGENIERÍA

COMPUTACIÓN PARALELA Y DISTRIBUIDA

Profesor:

John Corredor Franco

Presentan:

Roxanyffer Andreina Velasco Contreras

## MEDIDAS DE RENDIMIENTO

### Introducción

Se realizan las prácticas y las pruebas de los programas en relación al problema Planeta Órbita, para realizar la toma de rendimiento correspondiente. Este ejercicio tiene como finalidad realizar un análisis práctico e interactivo para diferenciar los procesos y el avance en rendimiento que representa y significa en concreto uno frente al otro, tomando como referencia significativa el hecho de que se realizará una comparación entre los rendimientos de los programas Cython y Python. Si bien comúnmente Python es el lenguaje más popular actualmente entre los dos, cabe aclarar que Cython es un lenguaje de programación que combina Python con el sistema de tipos estáticos de C y C++, lo que sin duda alguna es una unificación de las mejores características de estos grandes e increíbles lenguajes de programación.

Para presentar las medidas de rendimiento en primera instancia se generan un total de 8 iteraciones al programa, para Python y Cython respectivamente. A continuación se logrará realizar una visualización de los resultados de dicha práctica para ambos lenguajes de programación, con los resultados descritos en la siguiente tabla:

Python	Cython
1,60582	1,102748
1,597000	1,101458
1,591760	1,108395
1,593630	1,099695
1,586130	1,102034
1,590050	1,100285
1,589150	1,100347
1,590330	1,105538

*Imagen 1. Tabla de datos*

### Desarrollo

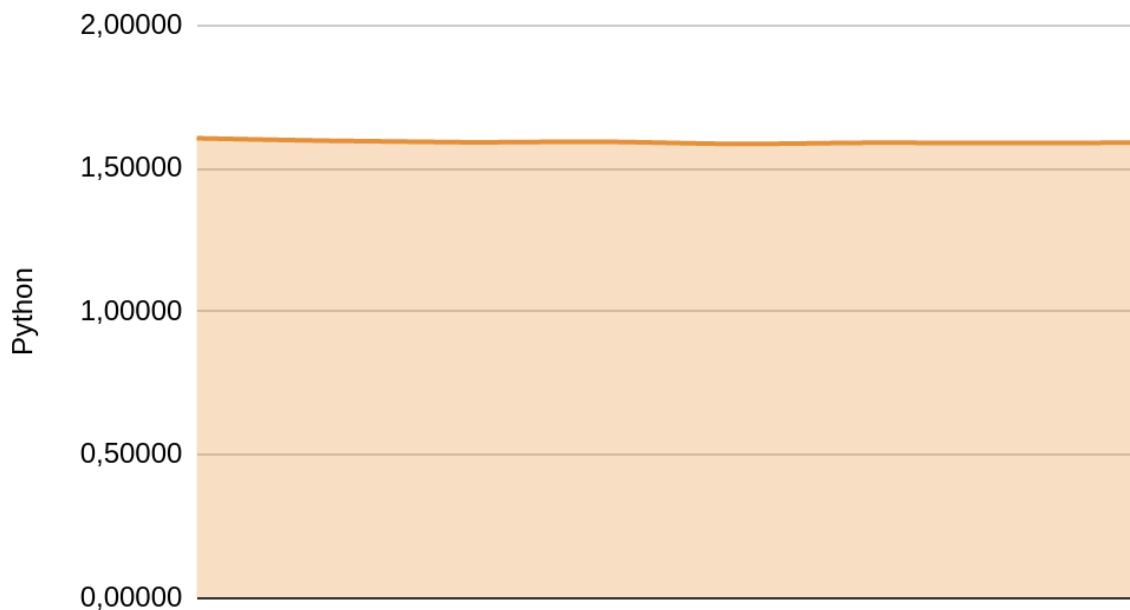
- **Resultados Python**

En cuanto a los resultados hallados en las 8 iteraciones para el lenguaje de programación Python, se observa un rendimiento formidable, en comparación a lo que pudo haber significado una comparación por ejemplo con el lenguaje de programación C, incluso es

probable que se presenten resultados mejores a los que se habrían obtenido para el lenguaje de programación C++.

Como se puede observar en la gráfica a continuación (*ver Imagen 2*) y de la mano con los datos en la tabla anterior, los resultados de las iteraciones para la toma del rendimiento de Python se encuentran en un rango entre 1.58s y 1.60s.

## Toma de rendimiento en función del tiempo - Python



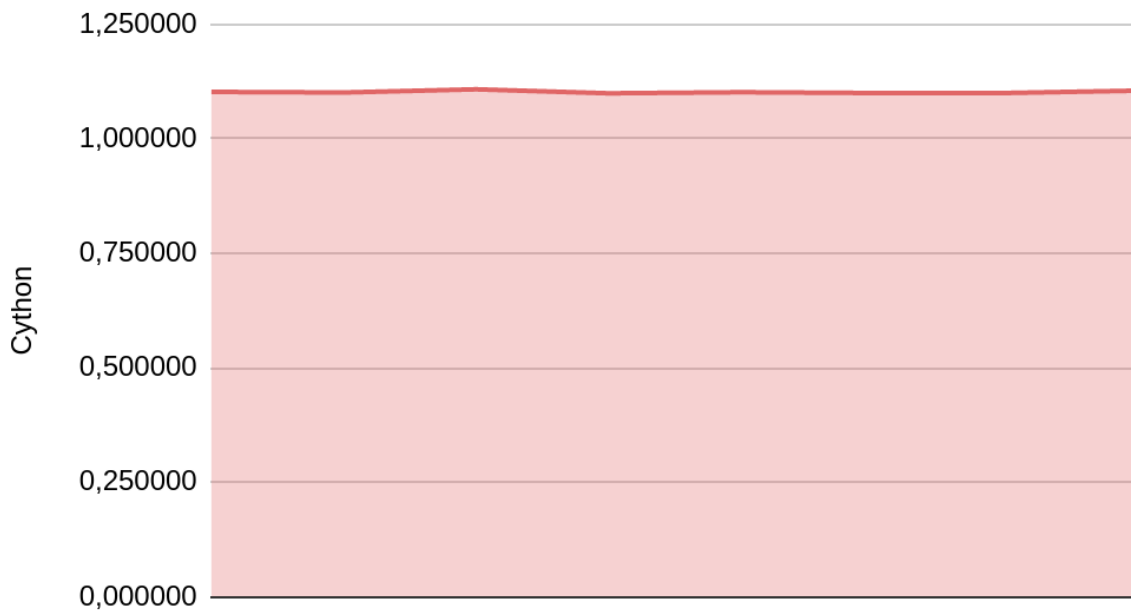
*Imagen 2. Gráfica de toma de rendimiento en función del tiempo - Python*

- **Resultados Cython**

En cuanto a los resultados hallados en las 8 iteraciones para el lenguaje de programación Cython, se observa un rendimiento significativo en gran manera, en comparación a lo percibido con el lenguaje de programación Python, demostrando en forma visual lo que se describe y la teoría expuesta en la introducción. Este programa reúne algunas de las mejores características de los lenguajes de programación Python, C y C++.

Como se puede observar en la gráfica a continuación (*ver Imagen 3*) y de la mano con los datos en la tabla anterior, los resultados de las iteraciones para la toma del rendimiento de Cython se encuentran en un rango entre 1.09s y 1.10s, lo que representa de forma sustancial una mejora en el rendimiento en comparación a los resultados de la toma de rendimiento presentada en Python.

## Toma de rendimiento en función del tiempo - Cython



*Imagen 3. Gráfica de toma de rendimiento en función del tiempo - Cython*

### Conclusión

Teniendo en cuenta el análisis realizado, de la mano con los procesos y el comportamiento que se visualiza en las gráficas, para los resultados de la toma de rendimiento en el problema de Planeta Órbita en los lenguajes de programación Python y Cython respectivamente, se logra comprender y analizar que al generar los procesos correspondientes, claro está, de la mano con el uso e implementación de buenas prácticas de programación a nivel algorítmico, se logra generar una interpretación basada fundamentalmente en el hecho de que Cython, como se ha mencionado anteriormente, es una fusión de muchas de las mejores características de Python, C y C++. Para quien conoce a nivel teórico y fundamentalmente práctico los lenguajes de programación mencionados, antes del diseño de Cython, habrán logrado generar una postura de reconocimiento y apreciación ante este, debido a que es un lenguaje de programación para simplificar la escritura de módulos de extensión para Python en C y C++. Cython posee una sintaxis bastante similar a Python pero con algunos agregados, como generar llamados de funciones de C y C++ y brindando la posibilidad de emplear distintos tipos estáticos en las variables.

Según este preámbulo teórico y los resultados obtenidos en la práctica es acertado afirmar que Cython contribuye significativamente al rendimiento de un equipo, lo que para cualquier profesional de la computación representa una característica relevante debido a que no es necesario realizar un sacrificio tan grande del rendimiento, sino que se obtienen resultados óptimos en los procesos realizados sin comprometer unidades de rendimiento y recursos

adicionales en la práctica. Siempre recomendando realizar una captura de datos que no se vea perjudicada por procesos web adicionales que generen basura en los datos recopilados.

### **Referencias bibliográficas**

- Correoso, K. (2015, 9 marzo). *C elemental, querido Cython* – Pybonacci.  
<https://pybonacci.org/2015/03/09/c-elemental-querido-cython/>