Prueba de rendimiento de la prueba TRACLUS:

Para esta prueba fracciones el código en partes, una por función creada por el código, hicimos 5 iteraciones variando la cantidad de filas que se usarían para ejecutar el algoritmo, empezando por 100 y incrementado 100 en cada una.

Las funcione que supusieron una carga mayor a un segundo en ejecución fueron:

* **Función 1 - partition():** esta función es perfectamente despreciable porque aunque si que parece costar algo no asciendo de los 10 segundos en ninguna de las ejecuciones.
* **Función 2 - get\_distance\_matrix():** la más costosa de las funciones, esta crea una matriz con las distancias de cada segmento a todos los otros y tiene una complejidad estimada de O(n^2) lo cual la hace una función muy costosos de escalar. Al momento de optimizar el código los mayores esfuerzos deben estar destinadas a esta función.
* **Función 3 - Ejecución librería sklearn:** Lo segundo más costoso de la ejecución, esto solo consiste en la llamada a funciones programadas en otra librería que posiblemente no puedan ser optimizadas de forma sencilla.
* **Función 4 - get\_representative\_trajectory():** aunque tome algunos minutos de la ejecución esta función es poco exigente respecto a las otras dos vistas anteriormente, además que representa todas las trayectorias de los clusters incluyendo al cluster -1 de descarte que tiende a tener el 60% de los segmentos y que posiblemente en analisis final no se utilice, por tanto se podría reducir fácilmente su tiempo de ejecución.

El eje vertical es segundos, el eje horizontal son número de trayectorias.