

Práctica 2

Divide y Vencerás

El elemento en su posición – Grupo 5

Integrantes:

- Guillermo Gómez Trenado
- Miguel Ángel Rispal Martínez
- Juan Carlos Ruiz García
- Ignacio Irurita Contreras
- Joaquín Fernández León
- Daniel García Martos

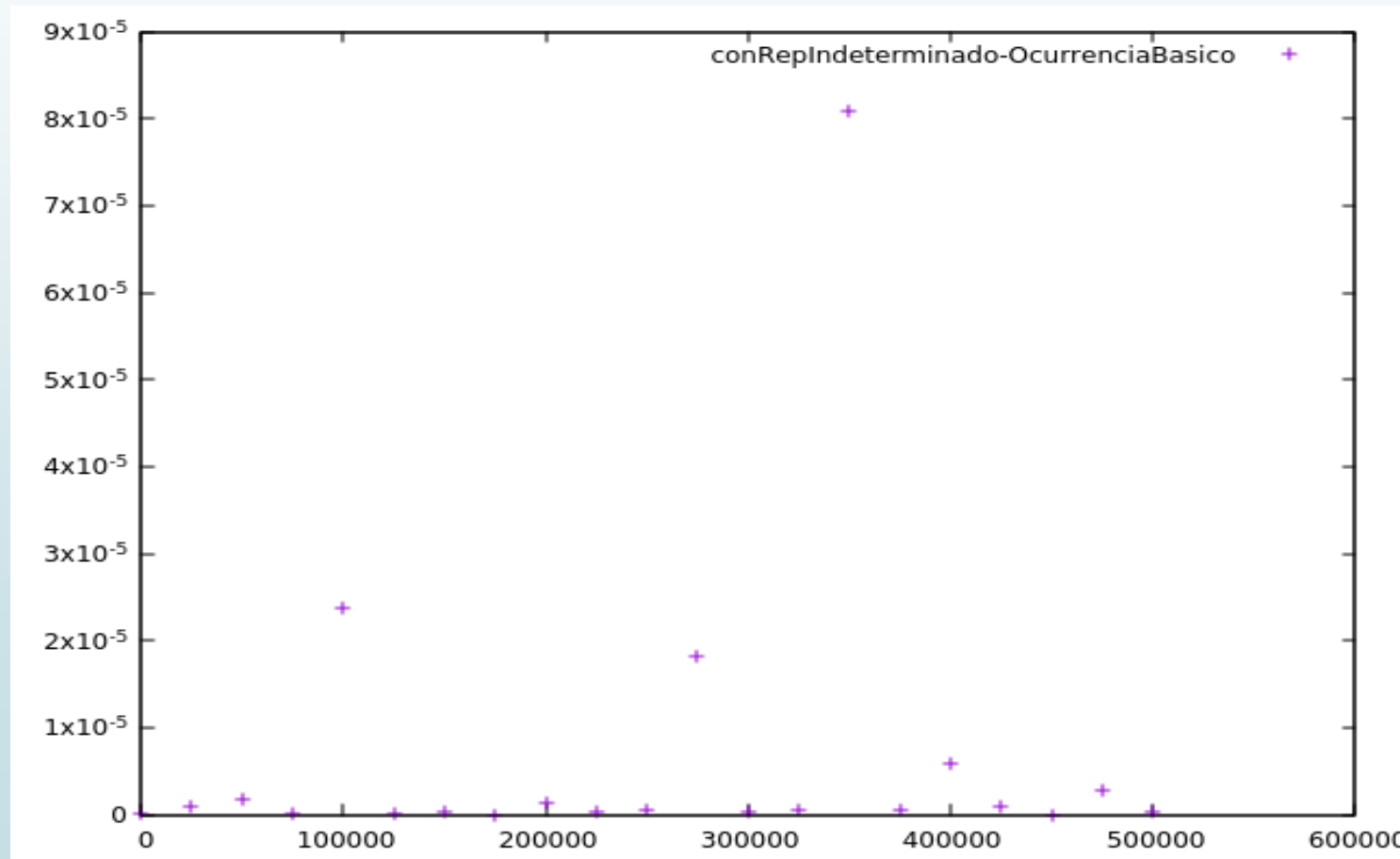


3. Algoritmo obvio

- El funcionamiento de este algoritmo es muy básico. Recibe como parámetros el **vector**, la posición de **inicio** y la posición de **fin** y simplemente realiza una **búsqueda secuencial** en dicho vector **desde ini hasta fin** para ver si en algún caso **mi_vector[i] == i**. En caso de acierto devuelve el valor de **i** y en otro caso **-1**.
- La eficiencia de este algoritmo es **$O(n)$** .

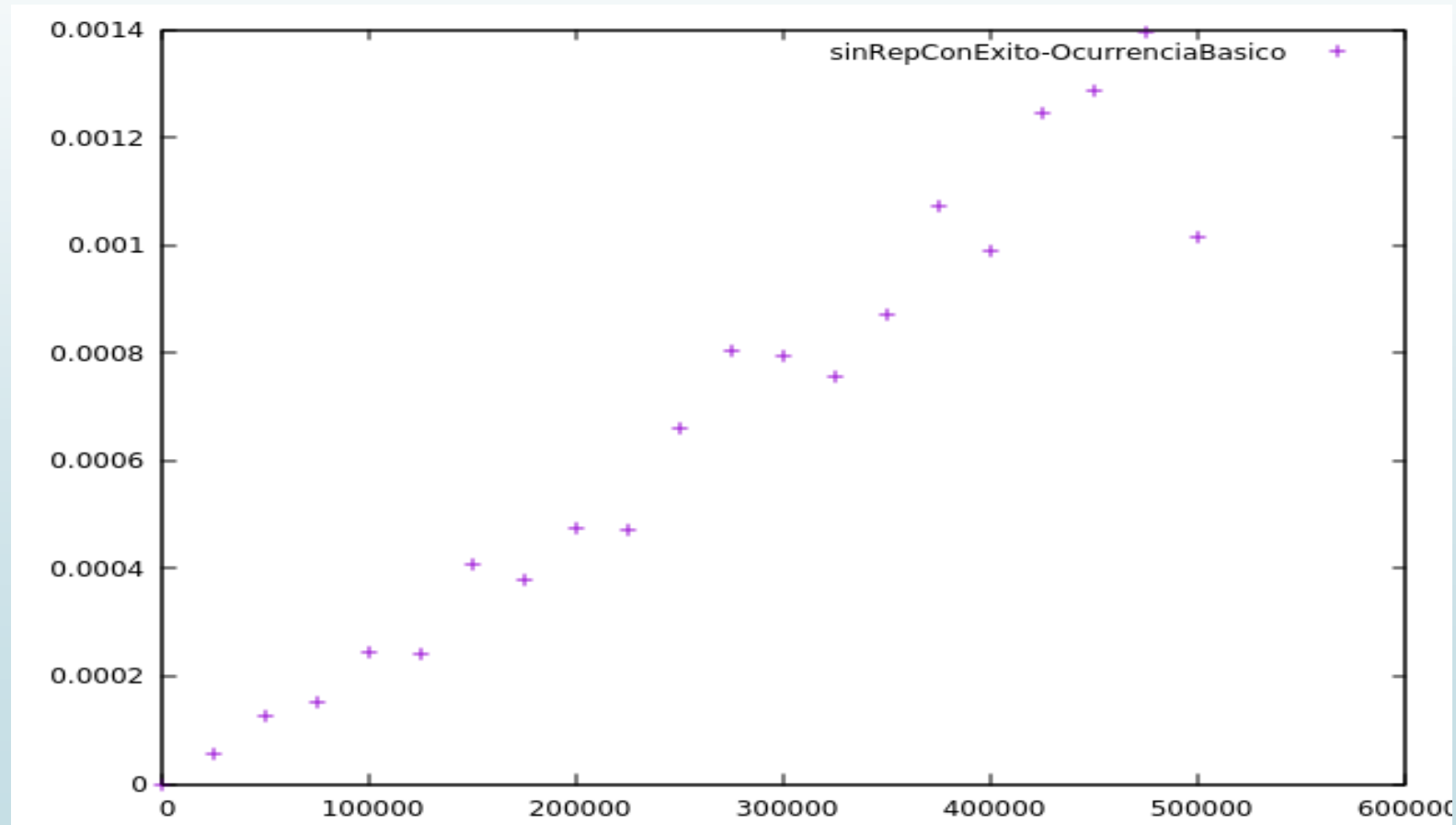
3.1. Eficiencia Empírica

Con elementos repetidos y ocurrencia indeterminada



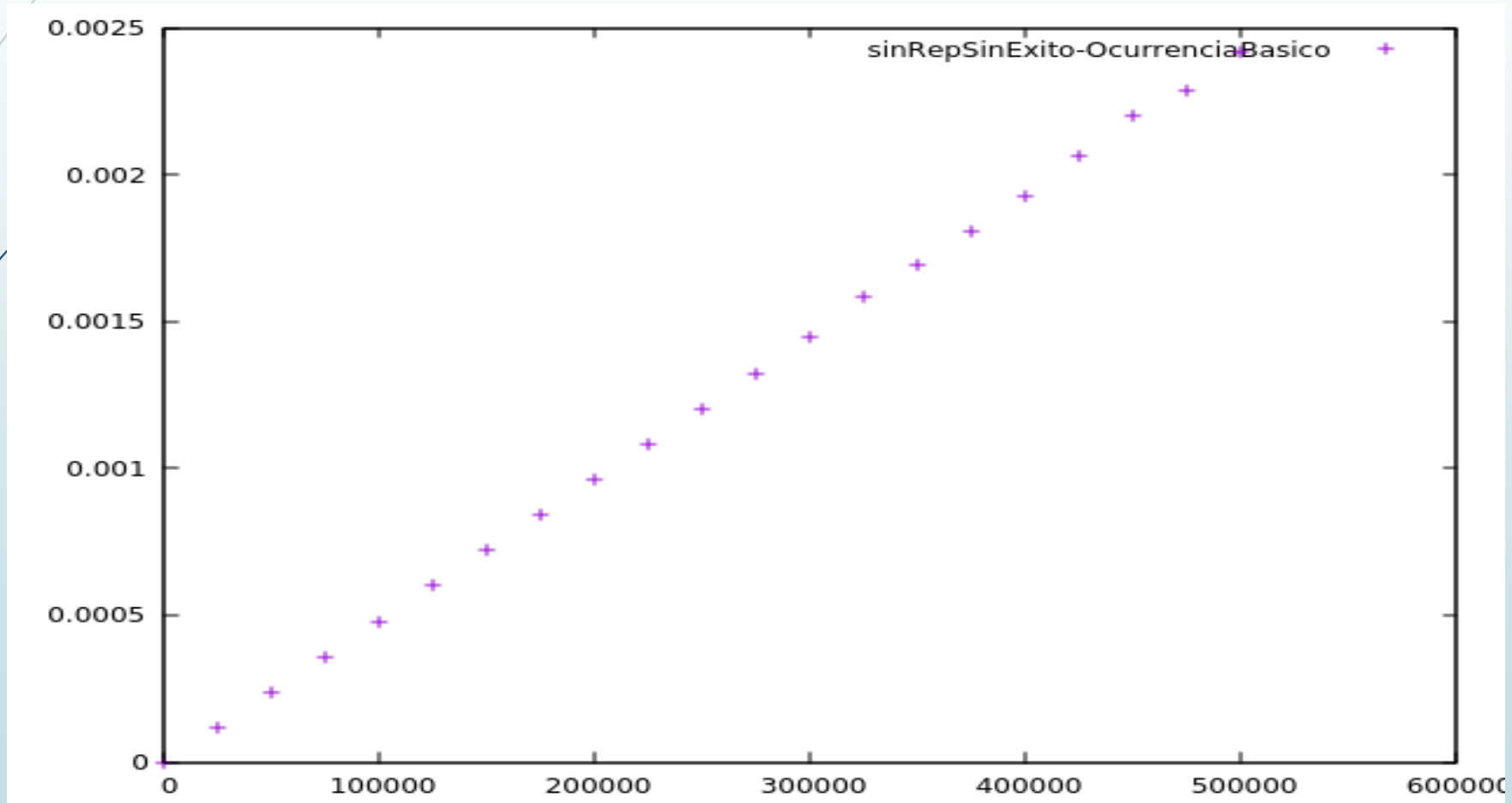
3.1. Eficiencia Empírica

Sin elementos repetidos y éxito en ocurrencia



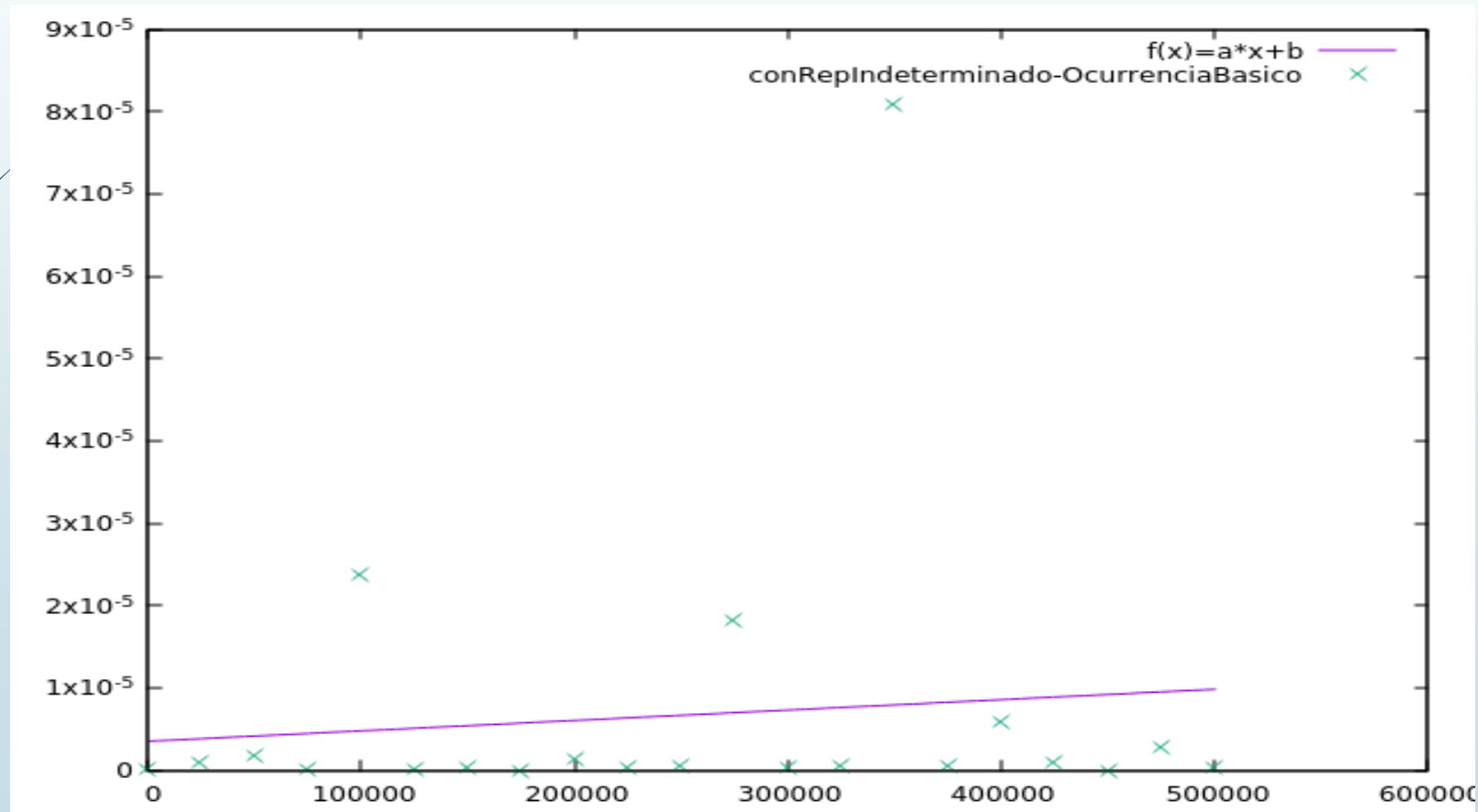
3.1. Eficiencia Empírica

Sin elementos repetidos y sin éxito en ocurrencia



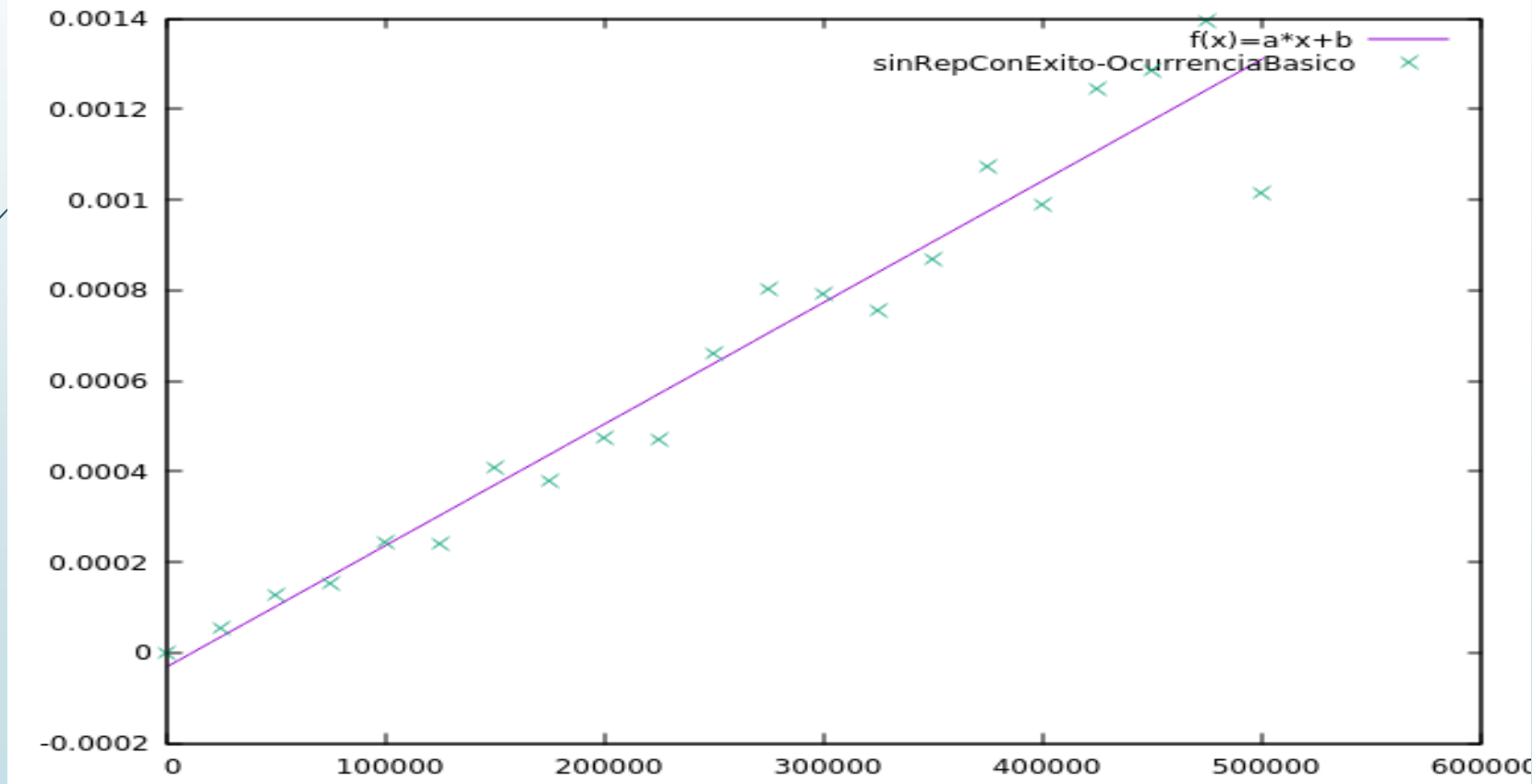
3.2. Eficiencia Híbrida (Ajuste lineal)

Con elementos repetidos y ocurrencia indeterminada



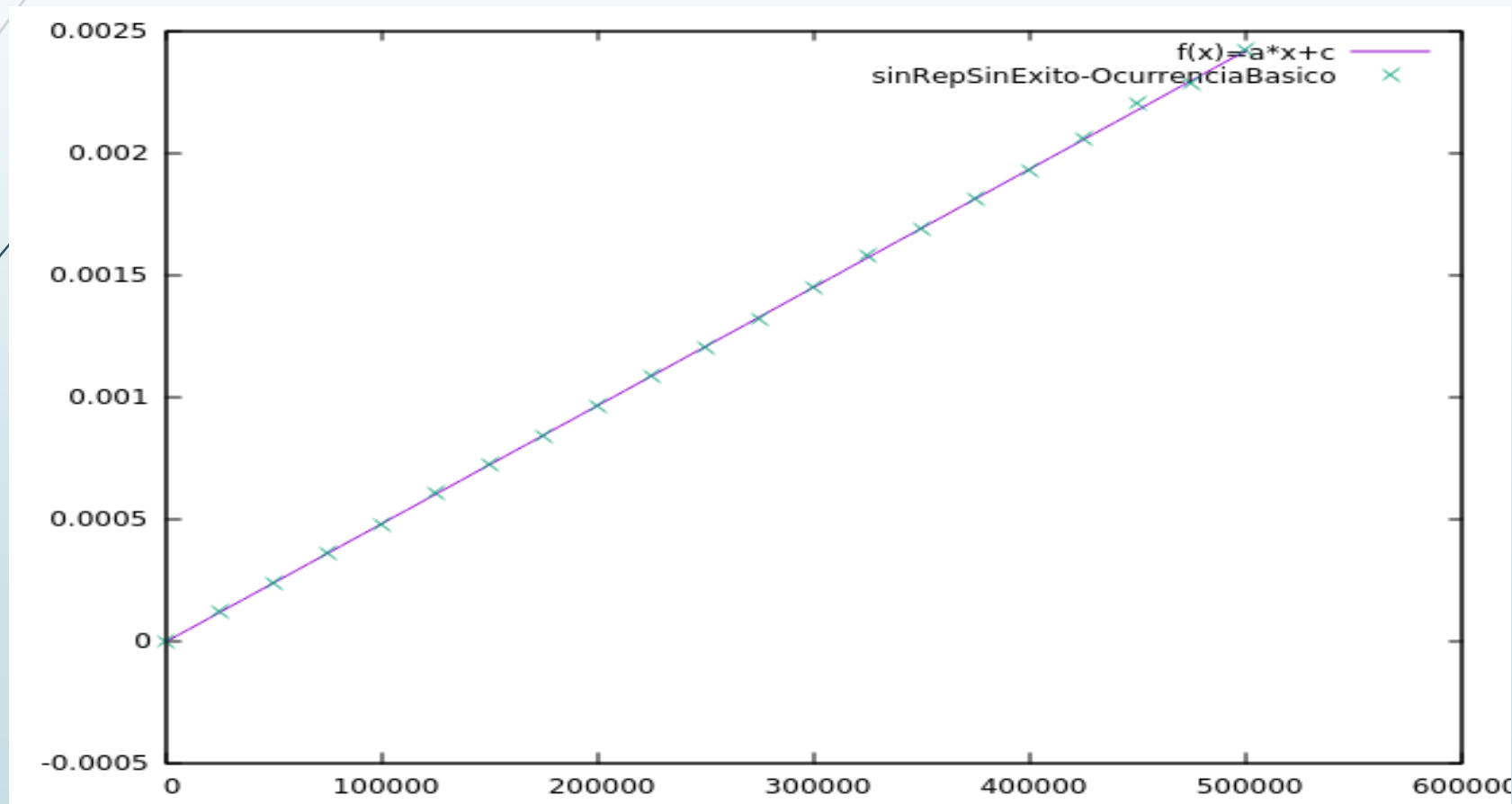
3.2. Eficiencia Híbrida (Ajuste lineal)

Sin elementos repetidos y éxito en ocurrencia



3.2. Eficiencia Híbrida (Ajuste lineal)

Sin elementos repetidos y sin éxito en ocurrencia



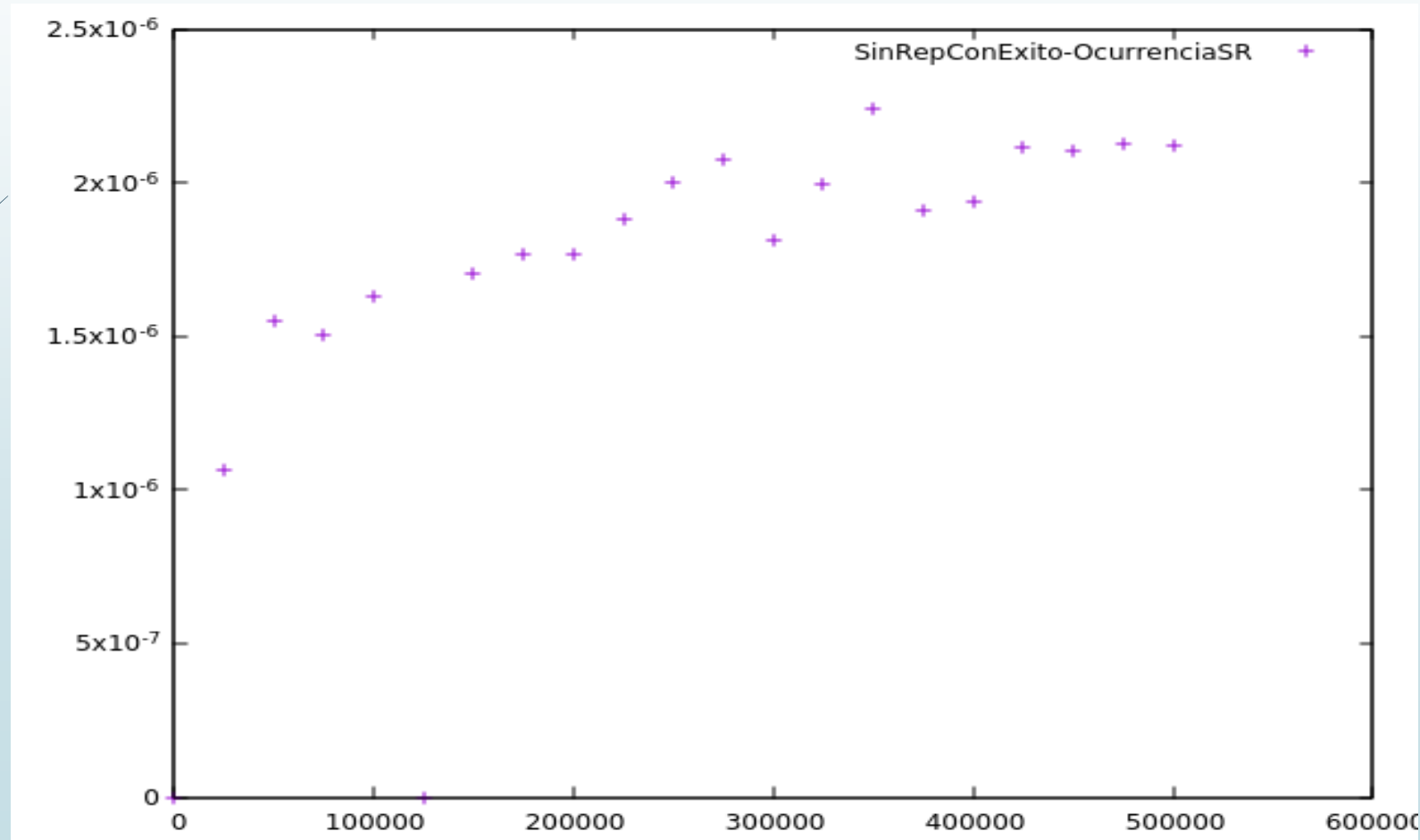
4. Algoritmo “Divide y vencerás sin repetidos”

- Este algoritmo basa su funcionamiento en la *técnica divide y vencerás*. En nuestro caso, dividiremos el vector en **dos mitades** de forma **recursiva**, buscando en todo momento que la posición del **medio** de nuestro vector sea igual a el valor de **v[medio]** devolviendo el valor **medio** en este caso o, en caso contrario pudiendo llegar al **caso base** en el que **ini=fin**, momento en el que sabremos que no existe ninguna ocurrencia.
- La eficiencia de este algoritmo es **$O(\log n)$** .

Mostramos las distintas gráficas obtenidas de las diferentes ejecuciones de los 2 vectores en este algoritmo (solo 2 vectores debido a que este algoritmo no funciona con un vector con repetidos).

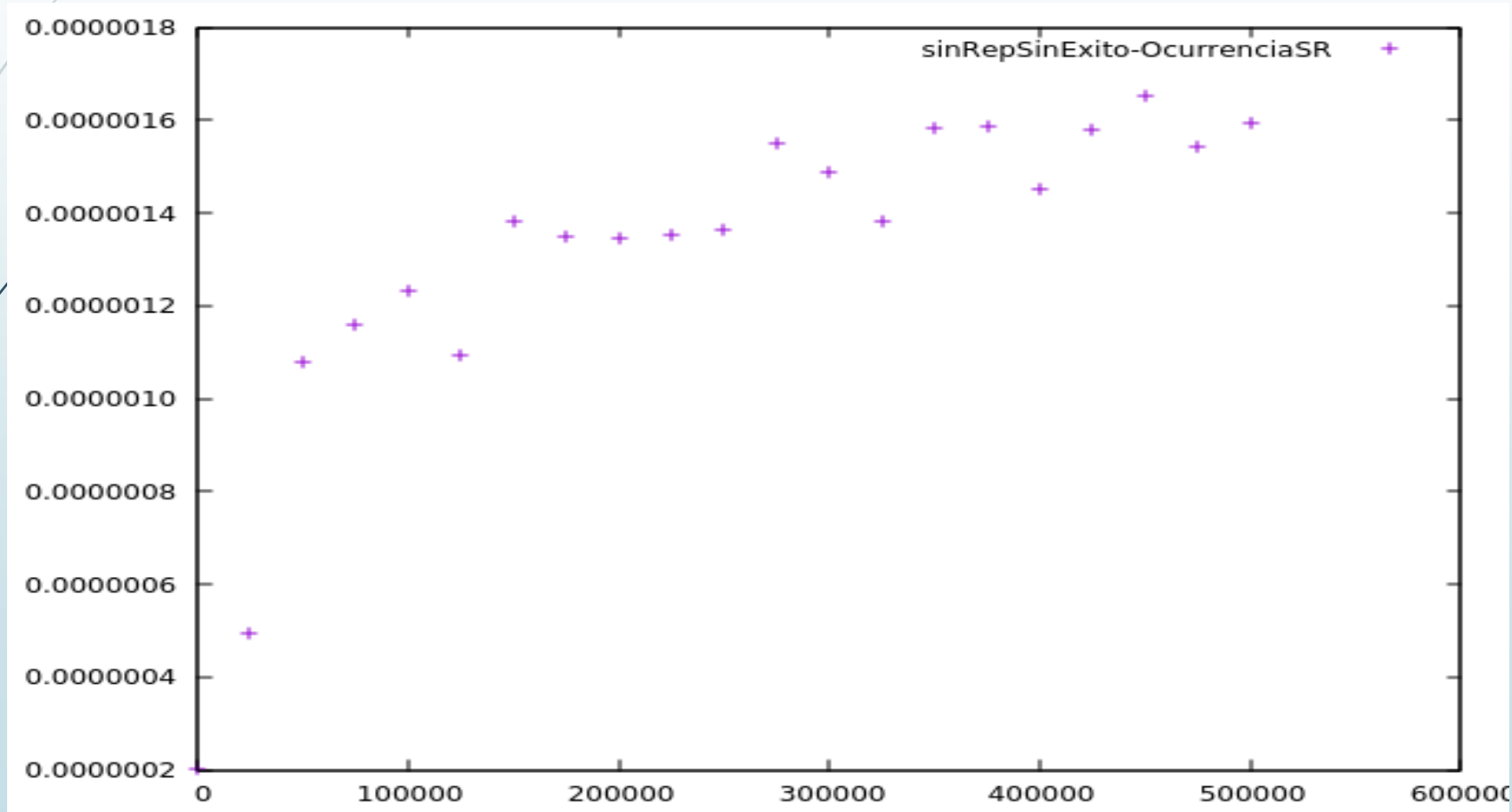
4.1. Eficiencia Empírica

Sin elementos repetidos y éxito en ocurrencia



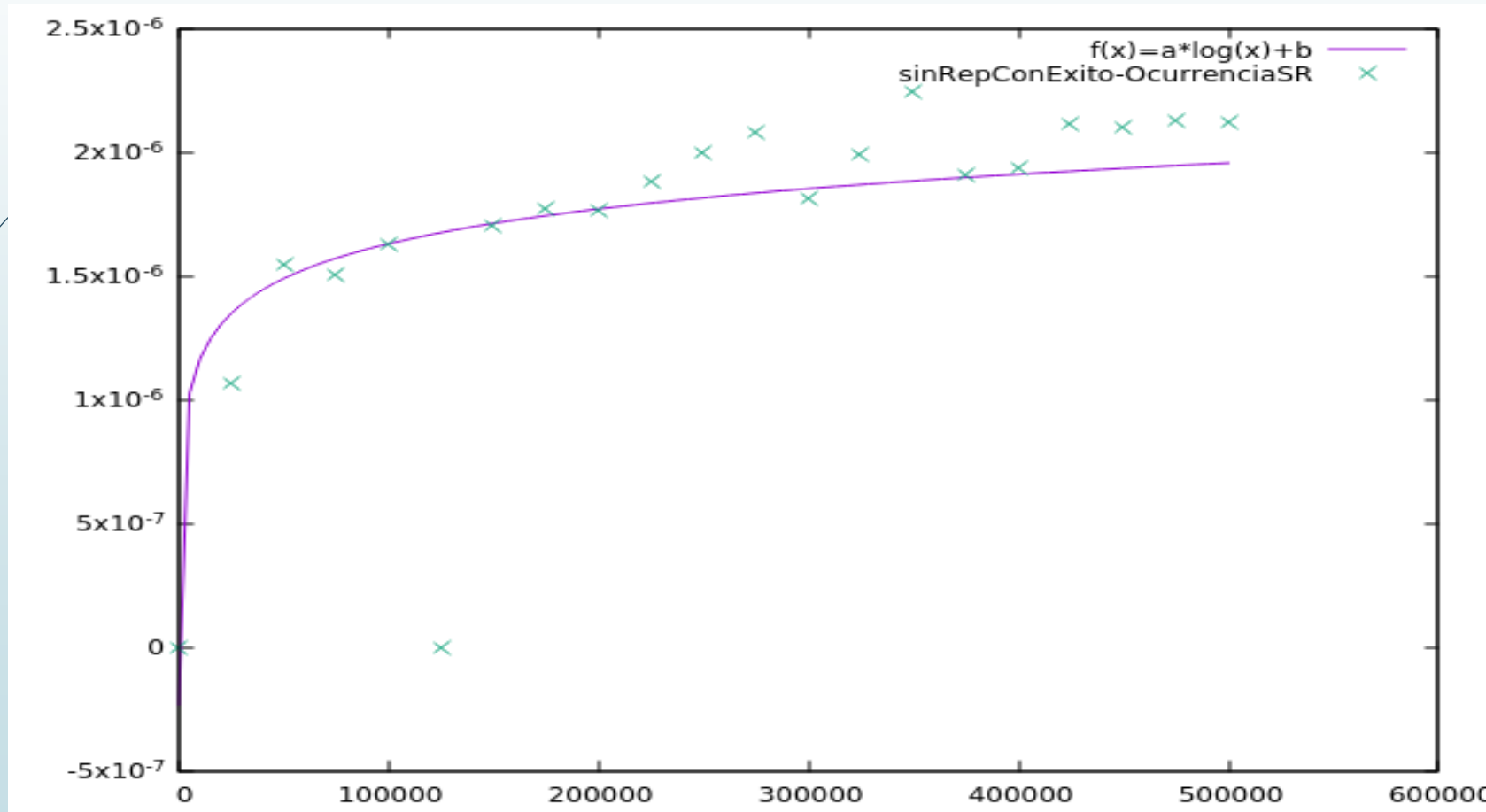
4.1. Eficiencia Empírica

Sin elementos repetidos y sin éxito en ocurrencia



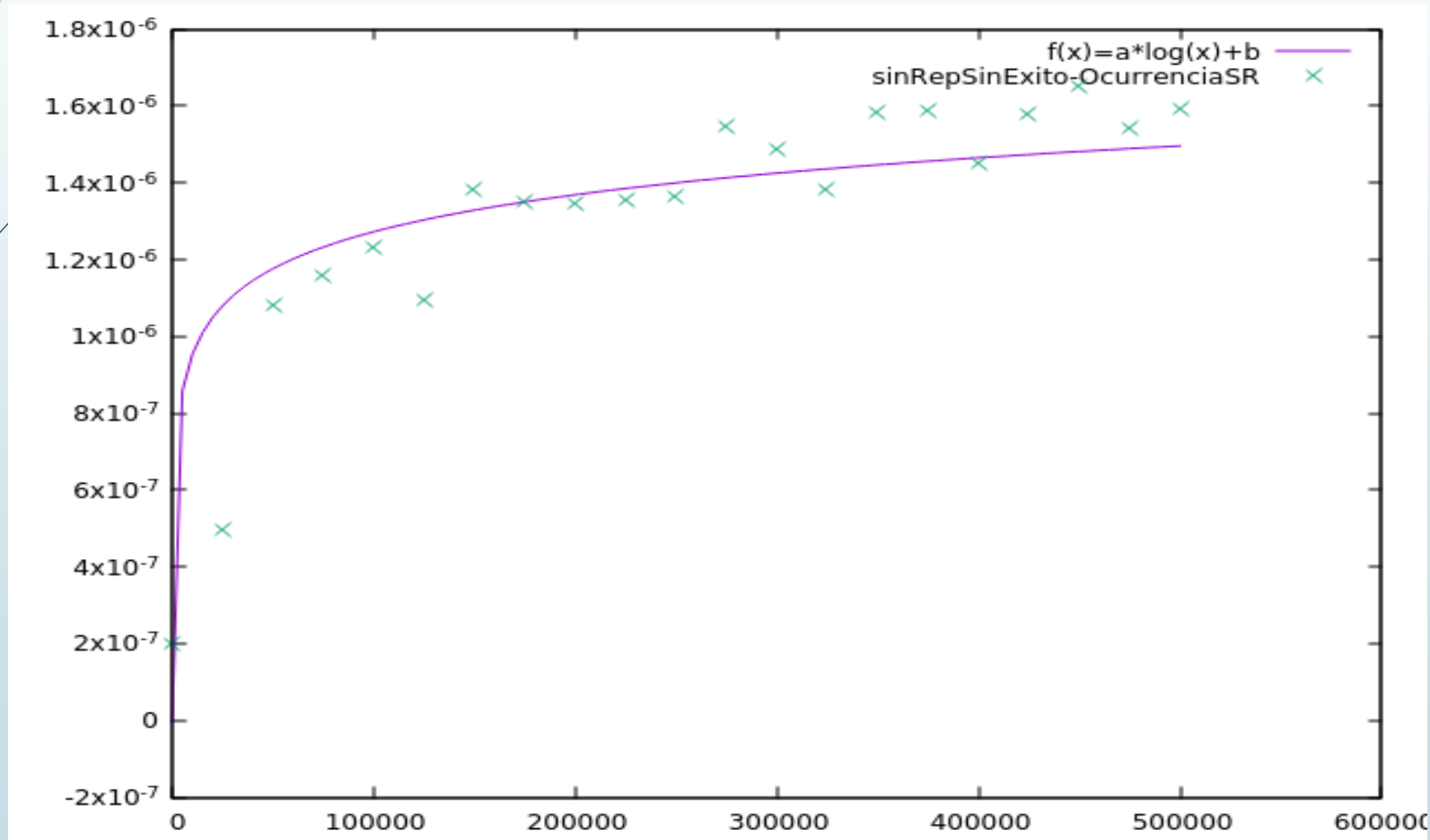
4.2. Eficiencia Híbrida (Ajuste logarítmico)

Sin elementos repetidos y con éxito en ocurrencia



4.2. Eficiencia Híbrida (Ajuste logarítmico)

Sin elementos repetidos y sin éxito en ocurrencia



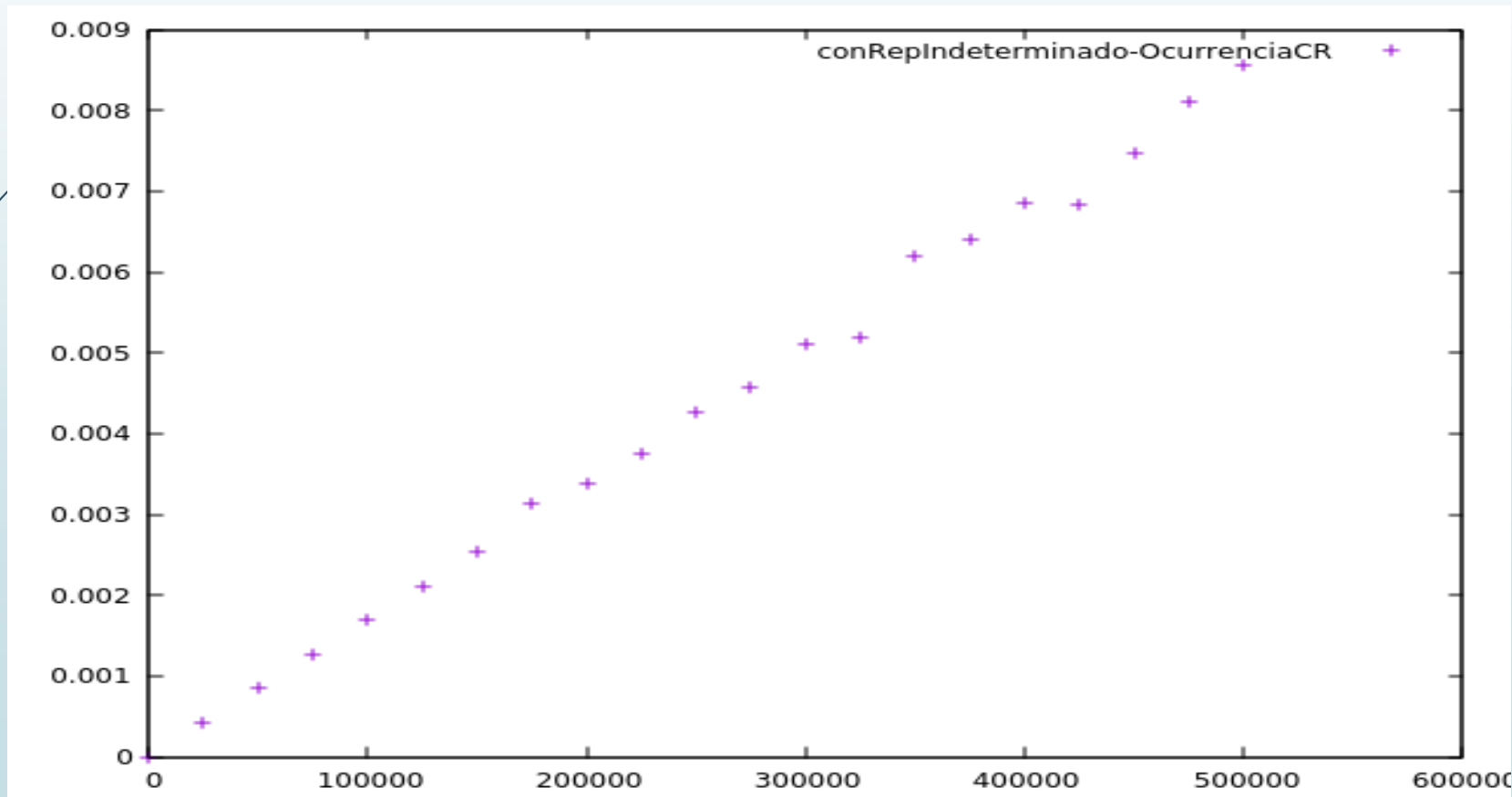
5. Algoritmo “Divide y Vencerás con repetidos”

- Este algoritmo al igual que el anterior también se basa en la *técnica divide y vencerás*. Su funcionamiento es muy similar al anterior, salvando la excepción de que ahora podemos **tener elementos repetidos** en el vector. Ahora lo que hacemos es **no descartar** una de las dos mitades y buscar en ambas, ya que al haber elementos repetidos **no podemos asegurarnos de en qué lado es posible que exista una ocurrencia**.
- Como dividimos el vector en **dos mitades** de forma **recursiva** pero **no descartamos ninguna mitad** al final acabamos recorriendo el vector completamente, por lo que la eficiencia de este algoritmo es **$O(n)$** .

Mostramos las distintas gráficas obtenidas de las diferentes ejecuciones de los 2 vectores en este algoritmo.

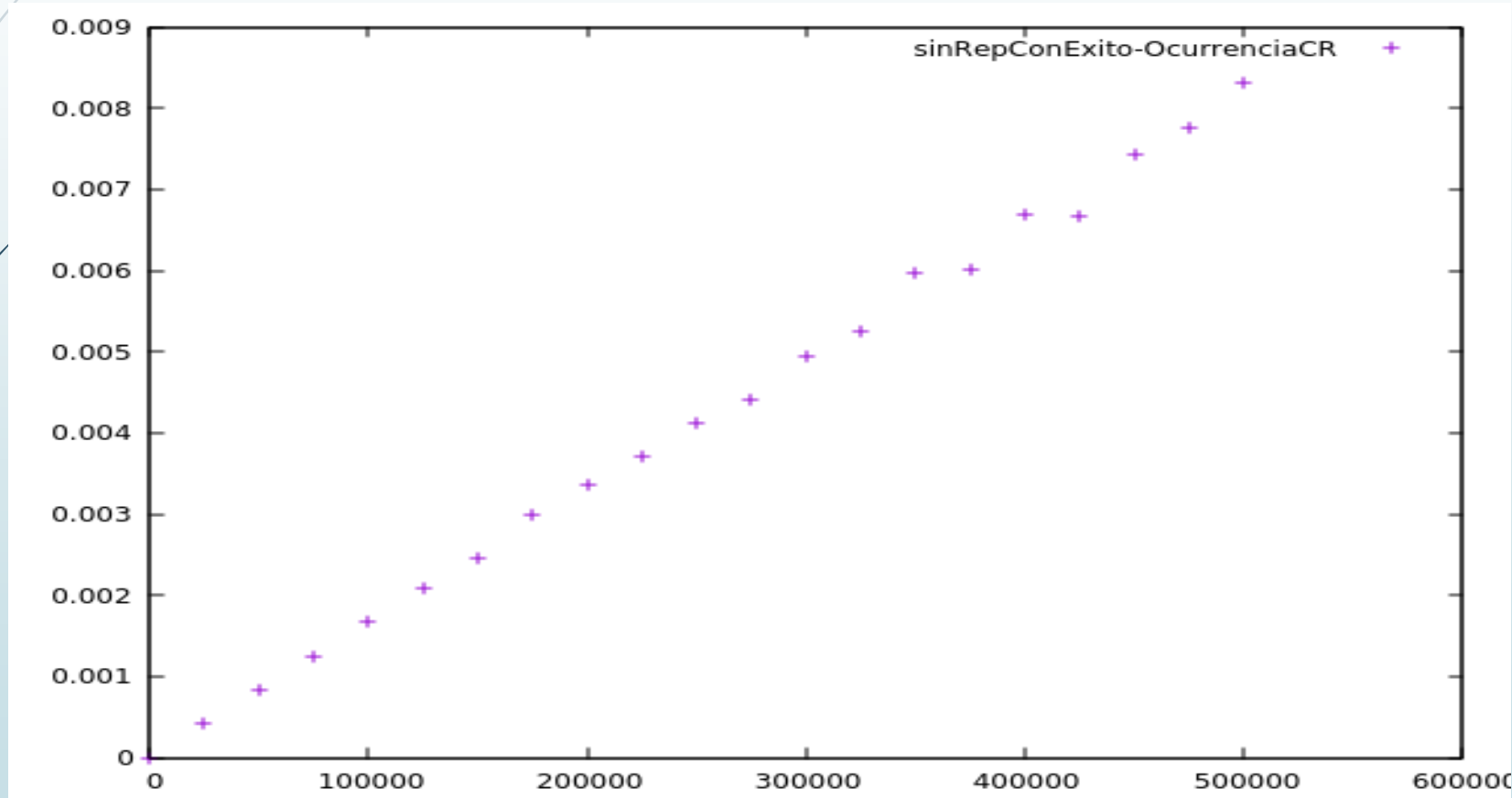
5.1. Eficiencia Empírica

Con elementos repetidos y ocurrencia indeterminada



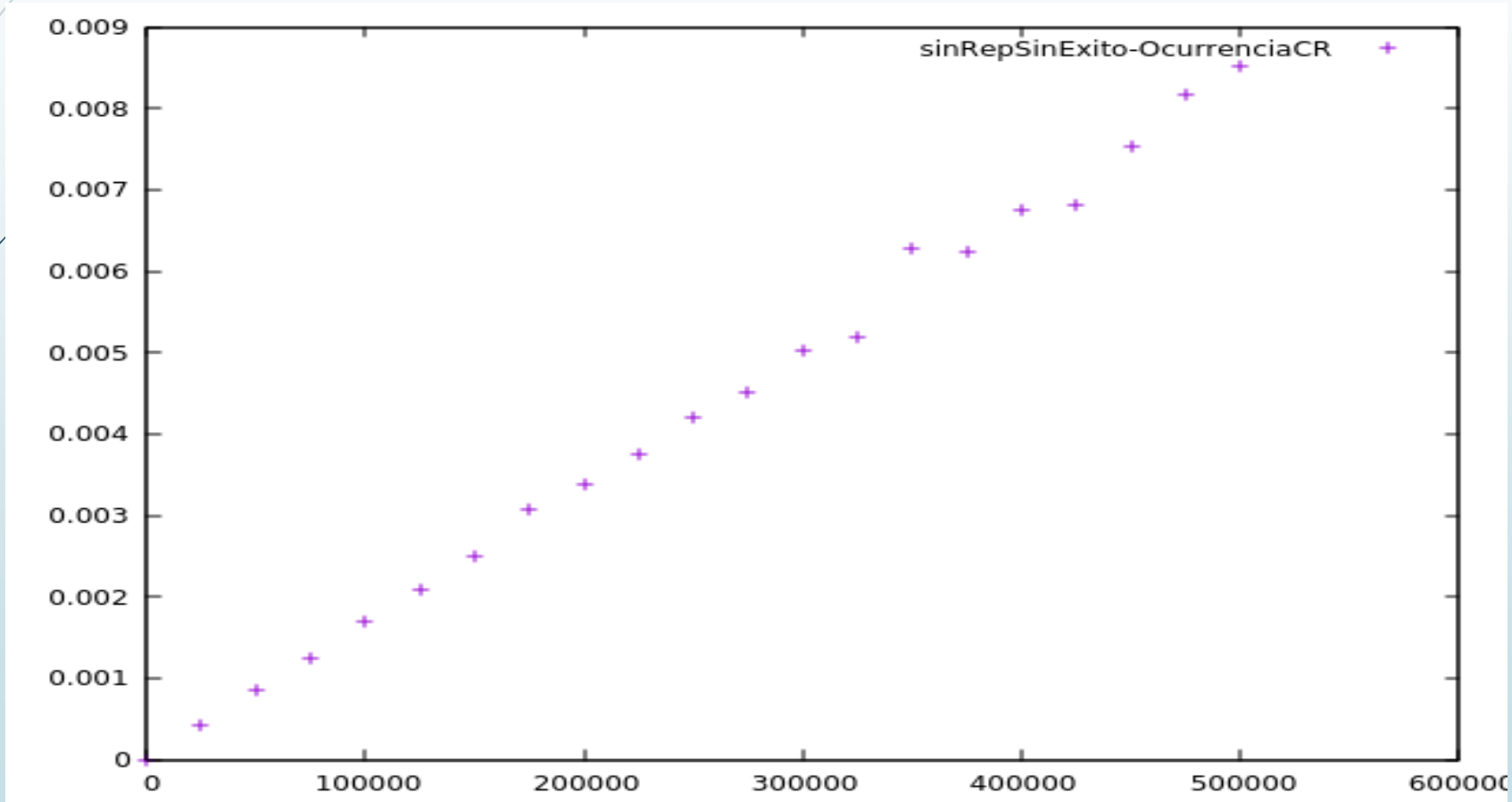
5.1. Eficiencia Empírica

Sin elementos repetidos y con éxito en ocurrencia



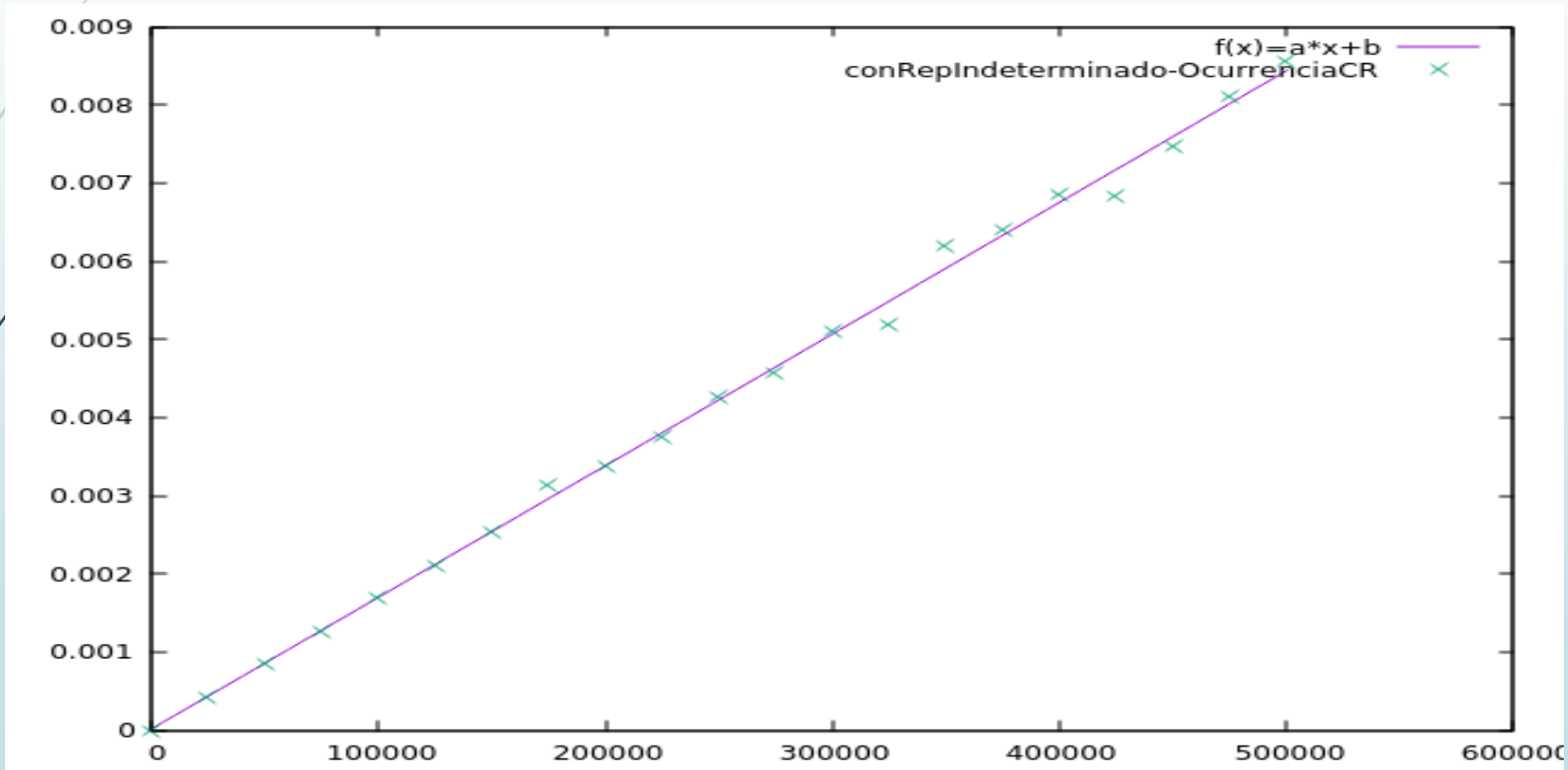
5.1. Eficiencia Empírica

Sin elementos repetidos y sin éxito en ocurrencia



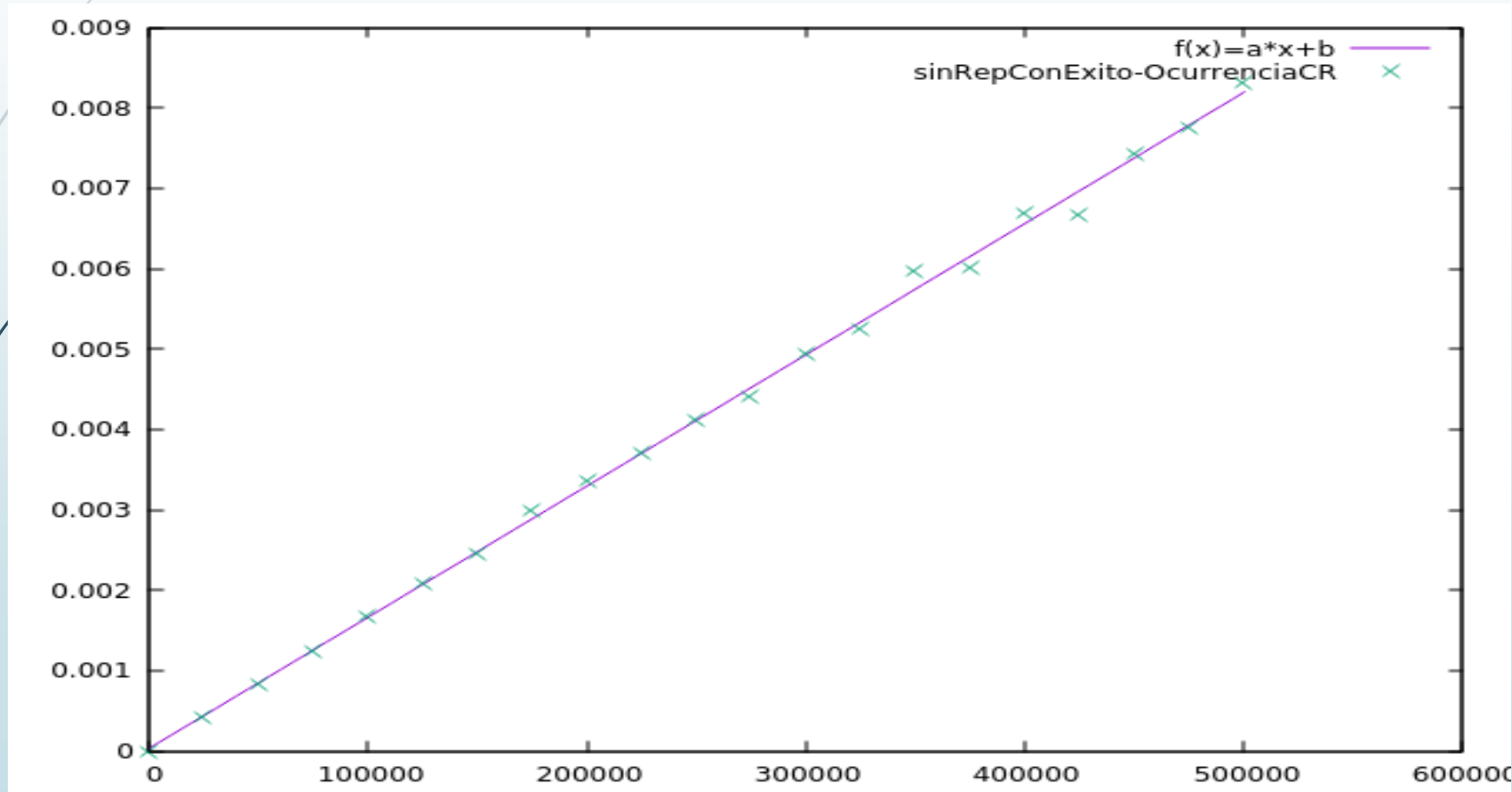
5.1. Eficiencia Híbrida (Ajuste lineal)

Con elementos repetidos y ocurrencia indeterminada



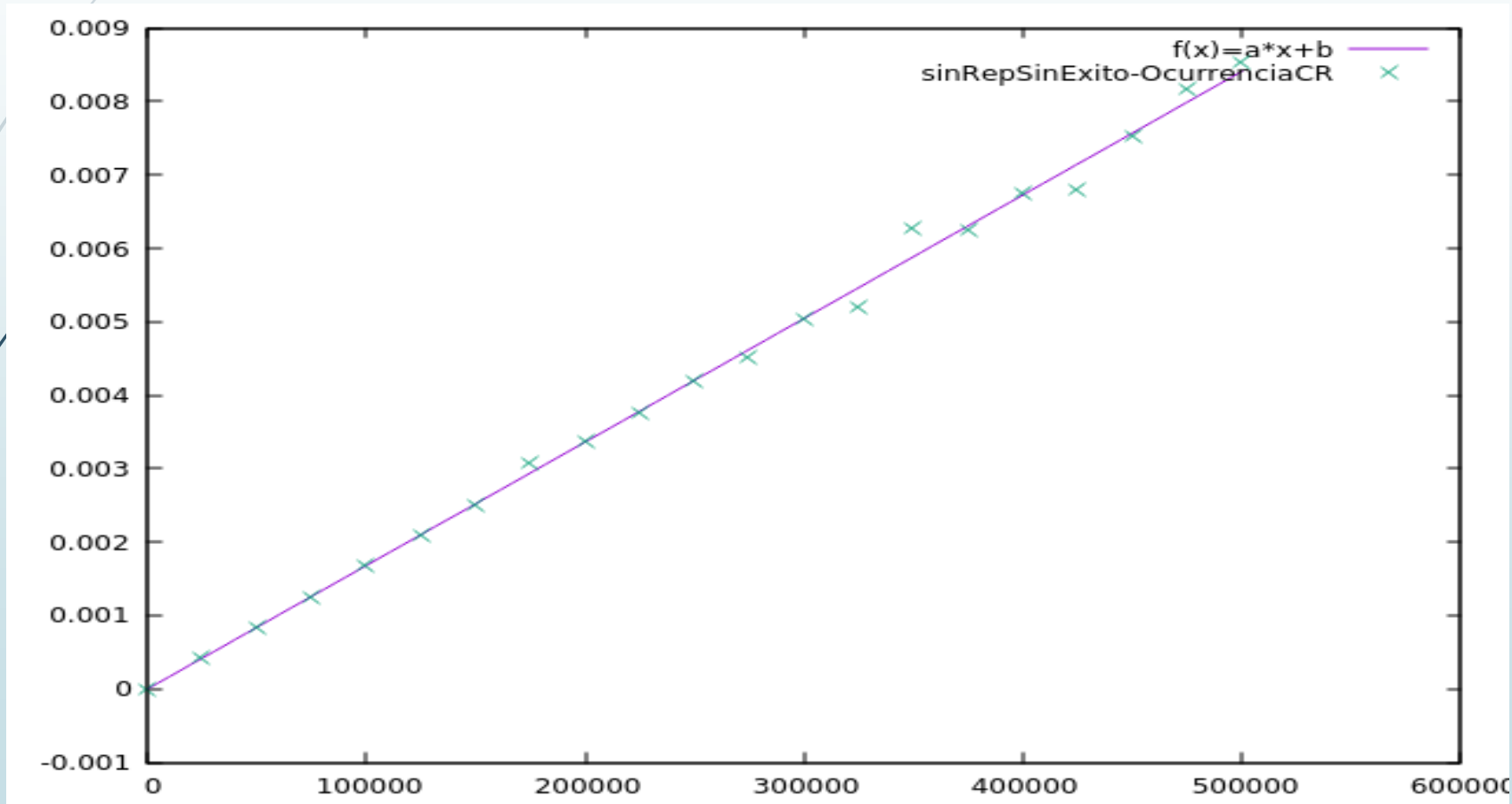
5.1. Eficiencia Híbrida (Ajuste lineal)

Sin elementos repetidos y con éxito en ocurrencia



5.1. Eficiencia Híbrida (Ajuste lineal)

Sin elementos repetidos y sin éxito en ocurrencia





6. Conclusión

Tras realizar la eficiencia empírica e híbrida de los **algoritmos básico y “divide y vencerás con repetidos”**, observando sus gráficas y los tiempos obtenidos hemos llegado a la conclusión de que este es un claro ejemplo en el que **no interesa aplicar la técnica de *divide y vencerás*** ya que los tiempos obtenidos **no mejoran en nada** a los del algoritmo “obvio”, y es absurdo molestarse en crear un algoritmo más complejo teniendo uno más simple que resulta **igual de eficiente**.