[[1]](#footnote-0)

Estructuras de Datos y Algoritmos Avanzados

**Laboratorio 1**

Danilo Bonometti

# Algoritmos

En este laboratorio se implementaron los algoritmos de búsqueda lineal, binaria y doblada y se evaluó su rendimiento para distinta cantidades de datos. Las implementaciones se encuentran adjuntas al boletín.

## Búsqueda Lineal

Para el peor caso de complejidad temporal se tiene que el elemento a buscar es el último elemento del arreglo, por lo tanto tiene una complejidad . Para el análisis adaptativo podemos ver que como el algoritmo busca elementos uno a uno linealmente la complejidad estará determinada por la posición del elemento a buscar, por lo que si se tiene el elemento en la posición , se tiene complejidad de .

## Búsqueda Binaria

La búsqueda binaria requiere de tener los elementos ordenados. Como esta búsqueda comienza desde la mediana y luego acota el espacio de búsqueda a la mitad en cada iteración, en el peor caso deberá hacer búsquedas hasta encontrar el elemento.

## Búsqueda Doblada

Esta búsqueda considera dos etapas. La primera hace una búsqueda exponencial con potencias de 2 y compara el el valor del elemento en el que se encuentra con el elemento a buscar, si la búsqueda se pasa del valor entonces continua con una búsqueda binaria acotada por los limites de la anterior potencia de 2 y la siguiente o el limite del arreglo. La complejidad de la primera etapa depende de la posición del elemento a buscar por lo que es de , luego la etapa de búsqueda binaria tiene una complejidad de . Finalmente la complejidad en términos de está acotada por .

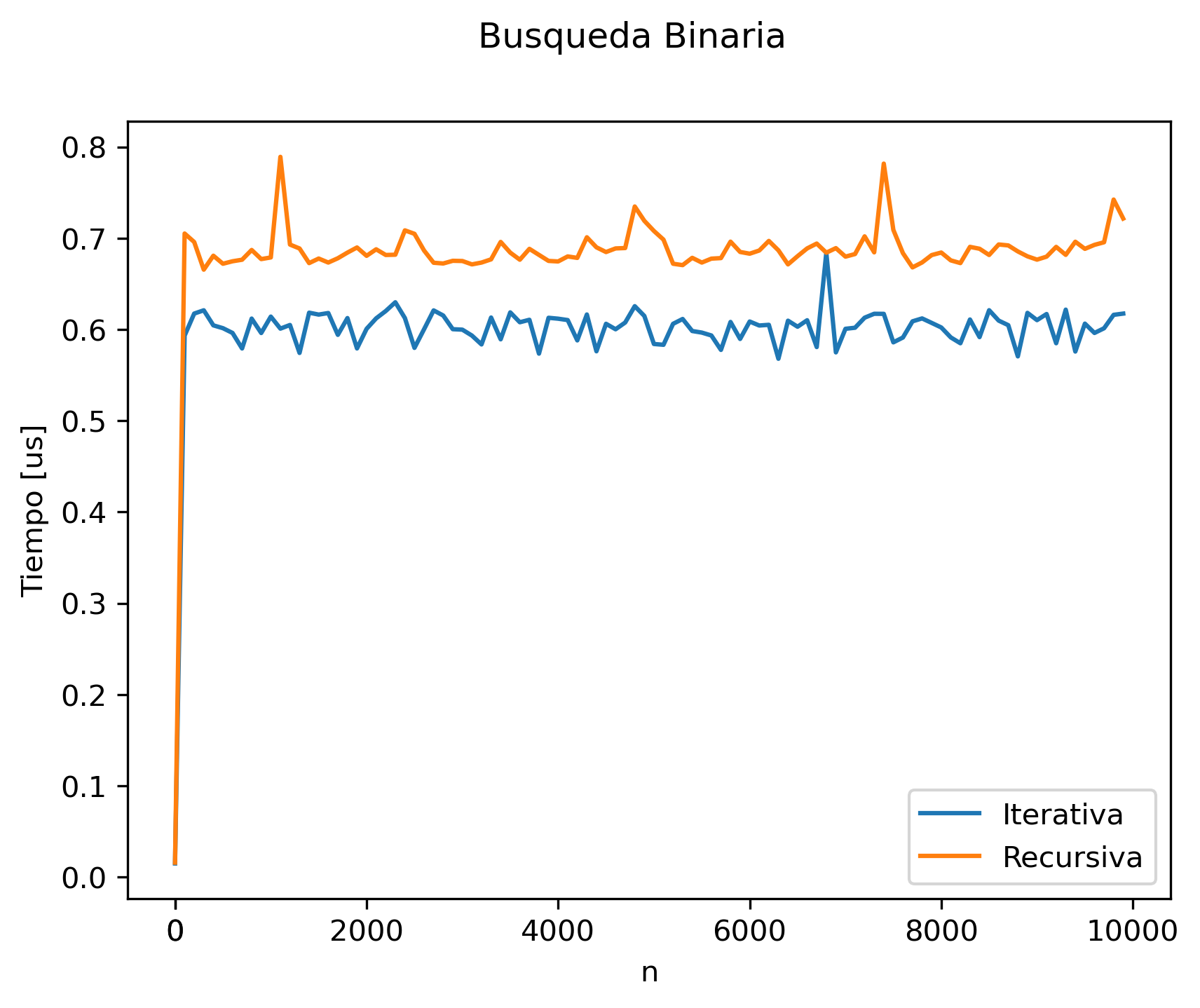
Para el peor caso se tiene que la primera etapa sobrepasa el arreglo por lo que tendrá una complejidad de .

# Resultados

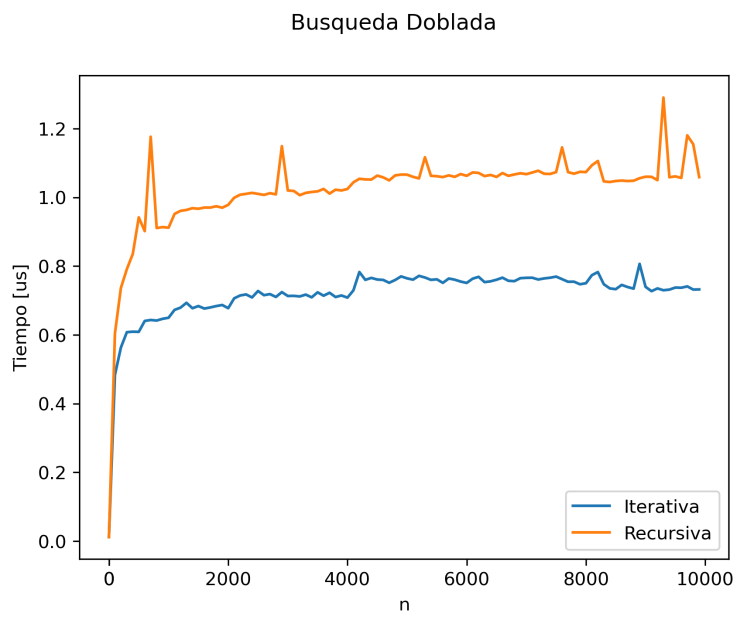
Se hicieron pruebas de 100 a 10000 elementos. Debido a que las búsquedas binaria y doblada se ejecutan demasiado rápido en las pruebas de búsqueda incremental para medirlas sin ruido, se tomaron ventanas de 1000 muestras y se ponderaron sus tiempos de búsqueda.

# busqueda_aleatoria

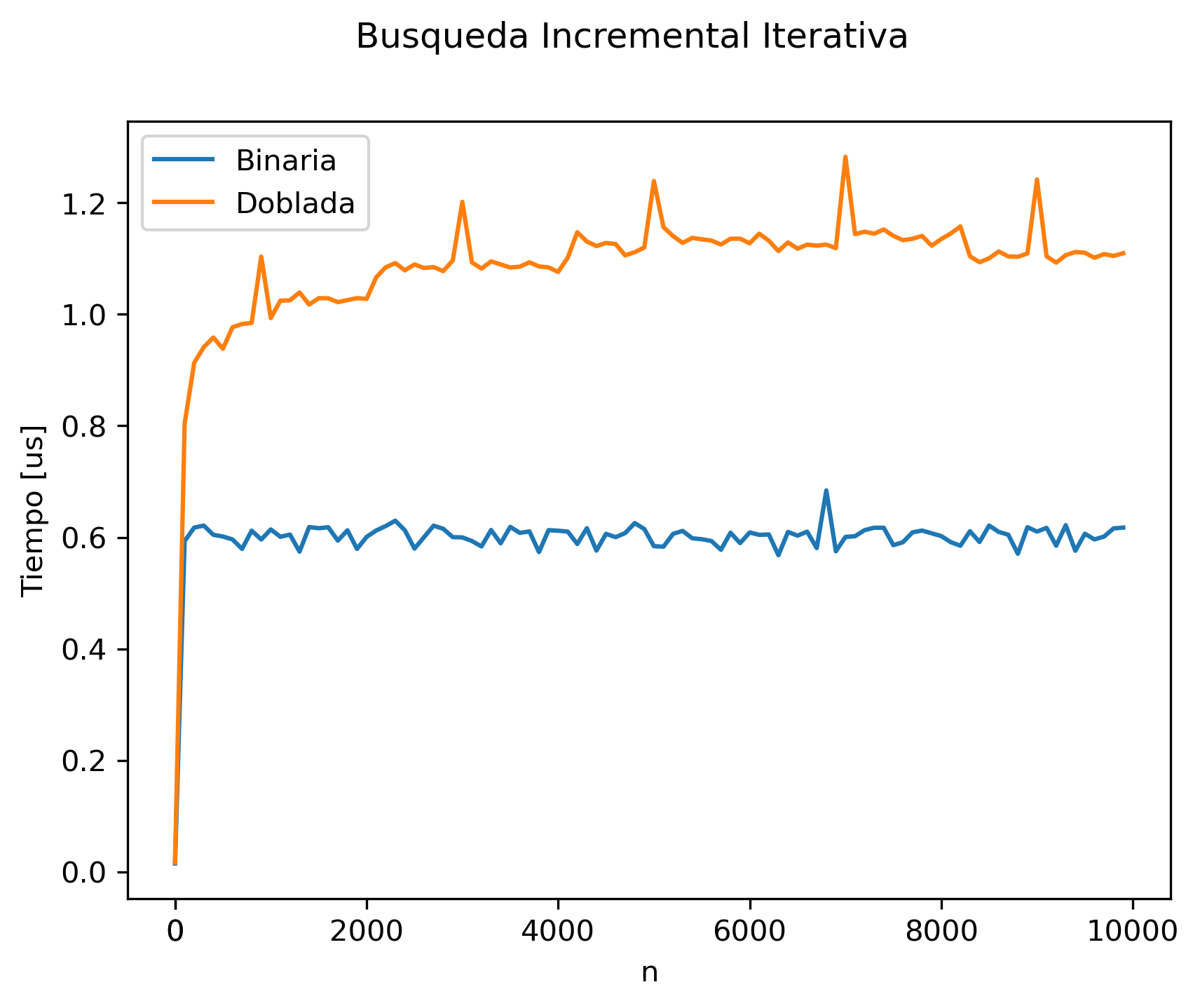
1. Búsqueda Aleatoria de las tres implementaciones.



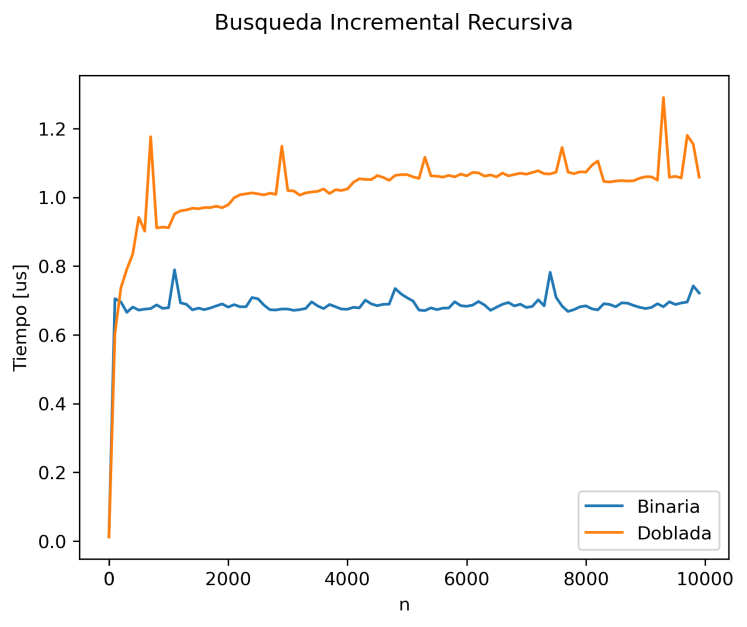
1. Búsqueda incremental binaria iterativa vs recursiva.



1. Búsqueda incremental doblada iterativa vs recursiva.



1. Búsqueda incremental iterativa binaria vs doblada.



1. Búsqueda incremental recursiva binaria vs doblada.

# Conclusiones

De la figura 1 se puede observar la relación entre las tres búsquedas y como crece el tiempo de ejecución para cada una. Se consideró una cantidad de 100 elementos, ya que para una cantidad mayor la curva lineal opaca fácilmente las curvas de las otras búsquedas. También se puede observar de a pesar de la ventaja adaptativa de la búsqueda doblada, se mantiene por sobre la búsqueda binaria. Esto también se refleja en el resto de las imágenes. Si bien, la búsqueda doblada debería tener mejor caso para algún elemento cercano al comienzo, las constantes asociadas a esta impiden que esa ventaja se vea reflejada en los resultados, ya que las búsquedas son demasiado rápidas para medirlas de forma fidedigna a su rendimiento teórico.

# Anexos

Los códigos del laboratorio pueden encontrarse en: <https://github.com/chromosome/laboratorio1>

1. [↑](#footnote-ref-0)