

**UNIVERSIDAD AUTÓNOMA DE BAJA  
CALIFORNIA**  
**Facultad de Ingeniería, Arquitectura y Diseño**

**Ingeniero en Software y Tecnologías Emergentes**



**Nombre Alumno:**  
Eliel Alfonso Ontiveros Ojeda

**Grupo:**  
932

08/11/2023

**1. ¿Cuál de los métodos de búsqueda, en su opinión, resultó ser el más eficiente y por qué?** El método más eficiente en este programa parece ser el "Método 1" en comparación con los otros métodos de búsqueda. Esto se debe a que el "Método 1" imprime la matriz antes de realizar la búsqueda, lo que significa que solo se imprime una vez, y las búsquedas se realizan después. Los otros métodos repiten la impresión de la matriz en cada iteración, lo que agrega un tiempo adicional de impresión que no es necesario.

**2. ¿Qué diferencias observaron al comparar los tiempos de ejecución de los métodos de búsqueda?** Las diferencias en los tiempos de ejecución entre los métodos de búsqueda se deben principalmente a la cantidad de veces que se imprime la matriz. Los métodos que imprimen la matriz antes de cada búsqueda (Métodos 1, 2, 3, 4, y 5) son más lentos en comparación con el "Método 1" que imprime la matriz una sola vez.

**3. ¿Qué creen que podría estar influyendo en la diferencia de tiempo de impresión entre los métodos? ¿Cómo podrían optimizar la velocidad de impresión?** La diferencia de tiempo de impresión entre los métodos se debe al hecho de que los métodos que imprimen la matriz antes de cada búsqueda realizan operaciones de impresión en cada iteración. Para optimizar la velocidad de impresión, se podría imprimir la matriz solo una vez antes de comenzar las iteraciones y almacenar el resultado en una variable para su posterior referencia en lugar de imprimir en cada iteración.

**4. ¿Cuál es la ventaja de utilizar punteros y fórmulas de direccionamiento en comparación con la manipulación tradicional de arrays? ¿Pueden citar ejemplos específicos donde esto resultó beneficioso?** La ventaja de utilizar punteros y fórmulas de direccionamiento en comparación con la manipulación tradicional de arrays es la flexibilidad y eficiencia. Los punteros permiten un acceso más directo a los elementos de la matriz, lo que puede ser beneficioso en operaciones de búsqueda y manipulación de datos. Además, los punteros son esenciales en situaciones donde se manejan estructuras de datos dinámicas, como listas enlazadas o árboles.

**5. ¿Pueden identificar situaciones en las que el uso de punteros y fórmulas de direccionamiento podría no ser la mejor opción? ¿En qué casos la manipulación tradicional de arrays sería preferible?** Situaciones en las que el uso de punteros y fórmulas de direccionamiento podría no ser la mejor opción incluyen cuando la manipulación de arrays es más simple y clara. En casos donde la estructura de datos es estática y de tamaño fijo, o cuando la complejidad del código aumenta significativamente con el uso de punteros, la manipulación tradicional de arrays puede ser preferible.

**6. ¿Cómo afecta la complejidad del código y la legibilidad al elegir entre métodos que utilizan punteros y fórmulas de direccionamiento y métodos más convencionales?**

**¿Existen situaciones en las que la claridad del código es más importante que la eficiencia?** La elección entre métodos que utilizan punteros y fórmulas de direccionamiento y métodos más convencionales puede afectar la complejidad del código y la legibilidad. En general, los métodos con punteros pueden ser menos legibles para quienes no están familiarizados con el uso de punteros, y esto podría afectar la mantenibilidad del código. En algunos casos, la claridad del código es más importante que la eficiencia, especialmente en proyectos donde varios desarrolladores pueden trabajar en el código.

**7. ¿Pueden proporcionar ejemplos de casos de uso en los que el rendimiento (tiempo de ejecución) es un factor crítico y, por lo tanto, los métodos con punteros y fórmulas de direccionamiento son esenciales?** Ejemplos de casos de uso en los que el rendimiento (tiempo de ejecución) es un factor crítico y, por lo tanto, los métodos con punteros y fórmulas de direccionamiento son esenciales incluyen:

- Algoritmos de procesamiento de imágenes y video en tiempo real.
- Sistemas de control en tiempo real para vehículos autónomos.
- Sistemas de bases de datos de alto rendimiento.
- Simulaciones numéricas y científicas que requieren un alto rendimiento de cálculos matemáticos.