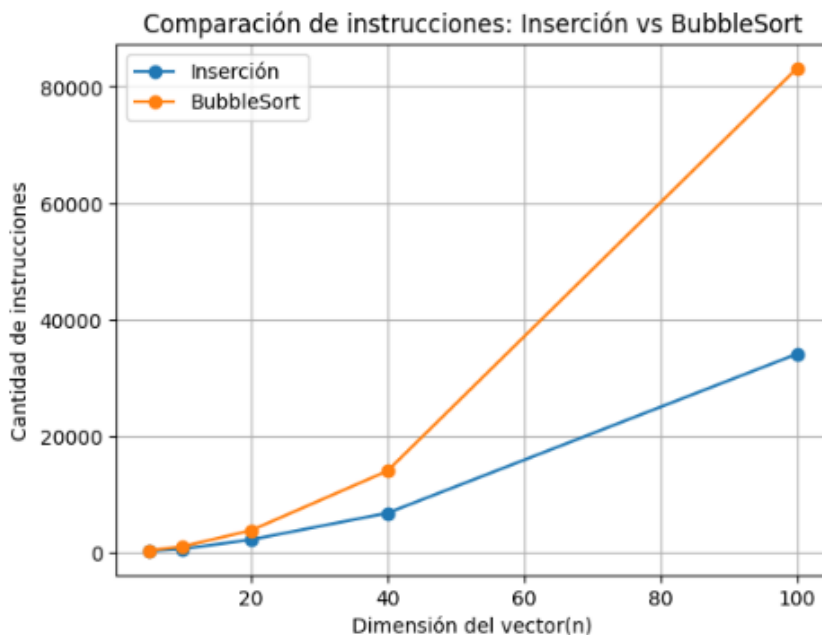


Análisis Informe extensión Máquina RAM

Autores: Javier Acosta Portocarrero(alu0101660769) y Gabriel Gallardo Noda(alu0101633961).

Hemos codificado en el lenguaje de nuestra Máquina RAM, basado en las instrucciones indicadas en el enunciado de la práctica, los algoritmos de ordenación de vectores BubbleSort e Inserción. Tanto el código de estos algoritmos como el código fuente de nuestra máquina RAM se encuentra alojado en el siguiente [repositorio](#).

Hemos ejecutado ambos algoritmos con varios vectores de distintas dimensiones para estudiar cómo cambia la cantidad de instrucciones de cada dependiendo del tamaño de los mismos. Tras recabar estos datos, hemos usado un [cuaderno](#) de Google Collab para poder comparar visualmente la evolución de la cantidad de instrucciones de cada algoritmo, obteniendo la siguiente gráfica:



Cómo se puede apreciar, el algoritmo **BubbleSort** ejecuta muchas más instrucciones que el algoritmo de **inserción**, incluso para dimensiones pequeñas. Esta diferencia se amplifica notoriamente conforme aumenta la dimensión (n). Aunque ambos algoritmos tienen complejidad $O(n^2)$, cuando medimos la complejidad de forma experimental con el número de instrucciones, observamos que **BubbleSort** es mucho menos eficiente. Esto se debe a que el algoritmo BubbleSort realiza muchos más intercambios de valores en memoria (elementos del vector) en cada iteración, mientras que el algoritmo de inserción solo realiza un intercambio por iteración, y, teniendo en cuenta que para cada intercambio son necesarias unas seis instrucciones, pues hace falta un registro auxiliar y varias cargas (LOAD) y almacenamientos (STORE), esto se traduce en una cantidad de instrucciones a la hora de la práctica mucho mayor, especialmente para vectores grandes.