

### **ORDENAMIENTO INTERNO**

	<b>Ventajas</b>	<b>Desventajas</b>	<b>Característica Principal</b>	<b>Breve Descripción</b>
<b>QuickSort</b>	<ul style="list-style-type: none"><li>- No requiere memoria adicional</li></ul>	<ul style="list-style-type: none"><li>- La implementación es más complicada</li><li>- La recursividad utiliza muchos recursos</li></ul>	<ul style="list-style-type: none"><li>- División por pivote</li></ul>	<ul style="list-style-type: none"><li>- Utiliza un pivote y ordena los elementos según el</li></ul>
<b>Insertion</b>	<ul style="list-style-type: none"><li>- Fácil de implementar</li><li>- Requerimiento mínimo de memoria</li></ul>	<ul style="list-style-type: none"><li>- Lento</li><li>- Realiza numerosas comparaciones</li></ul>	<ul style="list-style-type: none"><li>- Se puede llegar a demorar mucho</li></ul>	<ul style="list-style-type: none"><li>- Toma uno por uno los elementos y avanza hacia su posición con respecto a los anteriormente ordenados hasta recorrer todo el arreglo</li></ul>
<b>ShellSort</b>	<ul style="list-style-type: none"><li>- No requiere memoria adicional</li><li>- Mejor rendimiento que el método de inserción clásico</li></ul>	<ul style="list-style-type: none"><li>- Implementación confusa</li><li>- Realiza numerosas comparaciones e intercambios</li></ul>	<ul style="list-style-type: none"><li>- Compara e intercambia</li></ul>	<ul style="list-style-type: none"><li>- Asigna una distancia y ordena entre ellos</li></ul>
<b>Burbuja</b>	<ul style="list-style-type: none"><li>- Fácil implementación</li><li>- No requiere memoria adicional</li></ul>	<ul style="list-style-type: none"><li>- Realiza numerosas comparaciones</li><li>- Lento</li></ul>	<ul style="list-style-type: none"><li>- Toma el elemento mayor y lo va recorriendo de posición en posición hasta ponerlo en su lugar</li></ul>	<ul style="list-style-type: none"><li>- Se recorre el arreglo intercambiando los elementos adyacentes que estén desordenados. Se recorre el arreglo tantas veces hasta que ya no haya cambios que realizar</li></ul>
<b>Radix</b>	<ul style="list-style-type: none"><li>- El ordenamiento es razonablemente eficiente si el número de dígitos en las llaves no es demasiado grande</li></ul>	<ul style="list-style-type: none"><li>- no funciona tan bien cuando los números son muy largos, ya que el total de tiempo es proporcional a la longitud del número más grande y al número de elementos a ordenar</li></ul>	<ul style="list-style-type: none"><li>- Ordena los elementos basándose en los dígitos de los elementos individuales</li></ul>	<ul style="list-style-type: none"><li>- algoritmo de ordenación no comparativo</li></ul>

## **ORDENAMIENTO EXTERNO**

	<b><u>Ventajas</u></b>	<b><u>Desventajas</u></b>	<b><u>Característica Principal</u></b>	<b><u>Breve Descripción</u></b>
<b><u>Mezcla Directa</u></b>	<ul style="list-style-type: none"><li>- Efectivo para conjunto de datos a los que se puede acceder secuencialmente (arreglos, vectores, etc.)</li></ul>	<ul style="list-style-type: none"><li>- está definido recursivamente. Si se desea implementarla no recursivamente se debe emplear una pila y se requiere un espacio adicional de memoria para almacenarla.</li></ul>	<ul style="list-style-type: none"><li>- El destino de los elementos combinados se cambia después de que cada par ha sido ordenado para llenar uniformemente las dos secuencias que son el destino.</li></ul>	<ul style="list-style-type: none"><li>- Combina dos o más secuencias en una sola secuencia ordenada por medio de la selección repetida de los componentes accesibles en ese momento.</li></ul>
<b><u>Mezcla Natural</u></b>	<ul style="list-style-type: none"><li>- Mejora la complejidad de la mezcla directa para vectores inicialmente ordenados</li><li>- Es más eficiente que la mezcla directa debido a un menor tiempo de ejecución</li></ul>	<ul style="list-style-type: none"><li>- El proceso de ordenamiento incrementa sus tiempos de ejecución con todas aquellas tareas relacionadas con búsqueda de elemento, transmisión de elemento, comparaciones y permutaciones lo que da como resultado que el tiempo de ejecución sea más largo.</li></ul>	<ul style="list-style-type: none"><li>- Las particiones almacenadas se fusionan posteriormente para crear secuencias ordenadas cuya longitud se incrementa arbitrariamente hasta conseguir la total ordenación de los datos contenidos en el archivo original</li></ul>	<ul style="list-style-type: none"><li>- Aprovecha la existencia de secuencia ya ordenadas dentro de los datos del archivo</li></ul>