Ingeniería en Desarrollo de Software

4° Semestre

Programa de la unidad didáctica:

Estructura de datos

Unidad 2
Ordenación y búsqueda

Ciudad de México, abril del 2025

Clave:

Ingeniería	**TSU**
15142419	16142419

Universidad Abierta y a Distancia de México

Índice

Presentación de la unidad 3
Logros 3
Competencia específica 3
Temario de la unidad 3
2. Ordenación y búsqueda 4
2.1. Métodos de ordenación 4
2.2. Métodos de búsqueda 10
Cierre de la unidad 13
Para saber más… 13
Fuentes de consulta 14

Unidad 2. Ordenación y búsqueda

Presentación de la unidad

En esta segunda unidad conocerás dos conceptos importantes acerca de ordenación y búsqueda de datos: métodos de ordenación y métodos de búsqueda.

Los diferentes temas abordados en esta unidad dan continuidad a la unidad anterior, en donde, como podrás recordar, aprendiste a manejar las estructuras de datos, revisaste algoritmos y su representación, así como a los métodos asociados a cada estructura en particular.

En esta unidad aprenderás a ordenar los datos, llevarás a cabo búsquedas, revisarás textos y realizarás actividades, de modo que serás capaz de realizar ejercicios de programación donde apliques los distintos métodos de ordenación y de búsqueda de datos. Estos ejercicios de programación estarán relacionados con aplicaciones reales.

Logros de la unidad

Al término de esta unidad lograrás:

- Identificar las características principales de los métodos de ordenación
- Identificar las características principales de los métodos de búsqueda de datos
- Utilizar los métodos de búsqueda y ordenación de datos

Competencia específica

- Ejecutar los métodos de búsqueda y ordenación para mejorar los tiempos de ejecución de programas mediante los algoritmos de programación.

Temario de la unidad

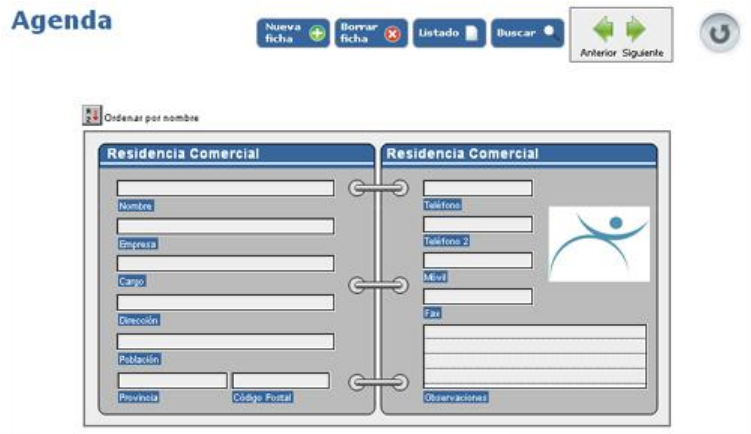
2. Ordenación y búsqueda
    - 2.1 Métodos de ordenación
        - 2.1.1. Método de la burbuja
        - 2.1.2. Método de inserción
        - 2.1.3. Método de selección
        - 2.1.4. Método *quicksort*
    - 2.1 Métodos de búsqueda
        - 2.2.1. Búsqueda secuencial
        - 2.2.2. Búsqueda binaria
        - 2.2.3. Búsqueda por interpolación

2. Ordenación y búsqueda

Esta unidad es sumamente interesante, ya que en ella podrás conocer y aprender a utilizar los diferentes métodos, tanto de ordenación como de búsqueda de datos. Cada uno de estos métodos se basa en un algoritmo y ofrece ventajas y desventajas en cuanto a los otros métodos existentes; es decir, conocerás diferentes opciones para ordenar y buscar datos, pero serás tú quien decida qué método emplearás, según sea el caso y tu preferencia. En esta unidad podrás complementar los temas vistos en la unidad anterior, puesto que en las estructuras de datos que aprendiste a utilizar, ahora podrás ordenar dichos datos y, posteriormente, realizar búsquedas bajo ciertos criterios. Esto significa que de nada serviría aprender sólo el almacenamiento correcto de datos de forma estructurada si posteriormente no se realizaran búsquedas de datos. Por ejemplo: es un desperdicio tener una excelente agenda que almacene los datos de nuestros contactos o clientes, si posteriormente no puede buscar datos de una forma ordenada o en orden alfabético ascendente.

2.1. Métodos de ordenación

El primer tema que consideraremos es: métodos de ordenación. La ordenación o clasificación de datos es una operación que consiste en disponer un conjunto de datos en algún orden determinado con respecto a uno de los campos de elementos del conjunto. La ordenación se conoce también como *sort* (por su traducción al inglés). Por ejemplo, cada elemento del conjunto de datos de una agenda telefónica tiene: nombre, dirección y número de teléfono; además de estos campos, podría tener otros, como correo electrónico. La guía telefónica está organizada en orden alfabético de nombres; los elementos numéricos se pueden ordenar en forma creciente o decreciente, de acuerdo con el valor numérico del elemento.



Agenda telefónica.
resiges.com. (n.d.). [Fotografía]. resiges.com.

Existen muchos **métodos de ordenación**, por lo que se debe prestar especial atención en su elección. ¿Cómo saber cuál es el mejor método? La eficiencia es el factor que mide la calidad y rendimiento de un método. En el caso de la operación de ordenación, dos criterios se suelen seguir a la hora de decidir qué método de ordenación es el más eficiente:

- Tiempo menor de ejecución en computadora.
- Menor número de instrucciones.

Métodos de ordenación de datos. Entre los diversos métodos de ordenación, los más conocidos y utilizados, se pueden mencionar los siguientes:

- **Método de la burbuja** (*bubble sort,* por su denominación en inglés)
- **Método de inserción**.
- **Método de selección**
- M**étodo de ordenación rápida (***quicksort***,** por su denominación en inglés).

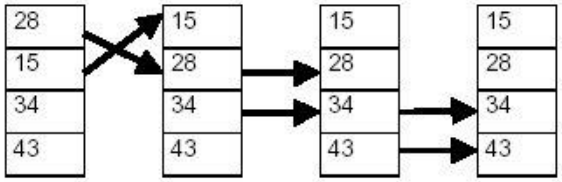
Para ampliar tu conocimiento acerca de los diferentes métodos tanto de ordenación como de búsqueda, **consulta** a Joyanes/Zahonero (2012. pp. 430 - 451), y **revisa** los diferentes ejemplos que ilustran estos métodos.

Método de burbuja

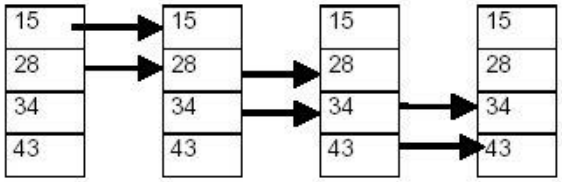
Este método funciona revisando cada elemento de la lista de elementos que va a ser ordenada con el siguiente elemento; luego, intercambian su posición si están en un orden inadecuado. Es necesario revisar varias veces toda la lista hasta que no se necesiten más intercambios, lo cual significa que la lista está ordenada.

Este método toma su nombre de la forma con la que suben los elementos durante los intercambios, como si fueran pequeñas "burbujas". También es conocido como el **método del intercambio directo** debido a que sólo usa comparaciones para operar elementos. Por ello, es considerado un método de comparación, siendo el más sencillo de implementar, aunque desafortunadamente no es muy eficiente.

En Joyanes/Zahonero (2012, pp. 438 - 442) puedes **consultar** de forma más amplia este algoritmo, al que también se le conoce como ordenación por hundimiento, debido al parecido entre las burbujas del agua que se van al fondo del recipiente que las contiene.



Nótese que el arreglo ya quedó ordenado, sin embargo, se hace una pasada más porque la computadora no advierte que el arreglo está en orden hasta que ocurre una pasada sin intercambios.



La ordenación por burbujeo se llama así porque los números más pequeños ascienden como burbujas hasta la parte superior, mientras que los mayores se hunden y caen hasta el fondo. Está garantizado que cada pasada pone el siguiente número más grande en su lugar, aunque pueden colocarse más de ellos en su lugar por casualidad.

Método de la burbuja.
Universidad de Murcia. (n.d.). [Fotografía]. http://www.um.es/

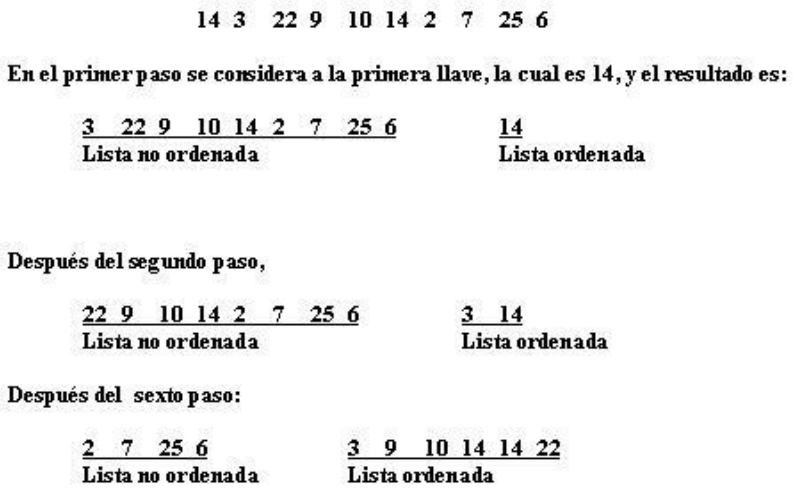
Método de inserción

Este método es similar al proceso típico de ordenar tarjetas de nombres (cartas de una baraja) por orden alfabético, el cual consiste en insertar un nombre en su posición correcta dentro de una lista o archivo que ya está ordenado.

Joyanes (2010, pp. 368-370) muestra diferentes ejemplos que ilustran este método de ordenación; y en su obra también encontrarás algunos pseudocódigos y diagramas de flujo que explican dicho método.

El algoritmo correspondiente a la ordenación por inserción consta de los siguientes pasos:

1. El primer elemento A[0] se considera ordenado; es decir, la lista inicial consta de un elemento.
2. Se inserta A[1] en la posición correcta, delante o detrás de A[0], dependiendo de si es menor o mayor.
3. Por cada bucle, ciclo o iteración i (desde i=1 hasta n-1), se explora la sublista A[i-1]. A[0], buscando la posición correcta de inserción. A la vez, se mueve hacia abajo (a la derecha en la sublista) una posición todos los elementos mayores que el elemento a insertar A[i], para dejar vacía esa posición.
4. Insertar el elemento en la posición correcta.



Método de inserción.
educacioninformatica.com. (n.d.). [Fotografía]. http://www.educacioninformatica.com

Método de selección

El ordenamiento por selección es un método que requiere O (n^2) operaciones para ordenar una lista de *n* elementos. La idea básica de un ordenamiento por selección es la selección repetida de la llave menor restante en una lista de datos no clasificados, como la siguiente llave (dato o registro), en una lista de datos ordenada que crece.

La totalidad de la lista de llaves no ordenadas debe estar disponible para que nosotros podamos seleccionar la llave con valor mínimo en esa lista. Sin embargo, la lista ordenada podrá ser puesta en la salida, a medida que avancemos.

En Joyanes (2010, pp. 370-373) puedes **revisar** diferentes ejemplos que ilustran este método de ordenación, y encontrarás también algunos pseudocódigos y diagramas de flujo que explican dicho método.

Este algoritmo es sencillo, y consiste en lo siguiente:

- Buscas el elemento más pequeño de la lista.
- Lo intercambias con el elemento ubicado en la primera posición de la lista.
- Buscas el segundo elemento más pequeño de la lista.
- Lo intercambias con el elemento que ocupa la segunda posición en la lista.
- Repites este proceso hasta que hayas ordenado toda la lista.

Su funcionamiento se indica enseguida:

- Buscar el mínimo elemento de la lista.
- Intercambiarlo con el primero.
- Buscar el mínimo en el resto de la lista.

- Intercambiarlo con el segundo.

Los pasos anteriores se pueden resumir:

- Buscar el mínimo elemento entre una posición i y el final de la lista.
- Intercambiar el mínimo con el elemento de la posición i.

El siguiente pseudocódigo ilustra el algoritmo en el método de ordenamiento por selección (*sort* en inglés).

```
para i=1 hasta n-1
    minimo = i;
    para j=i+1 hasta n
        si lista[j] < lista[minimo] entonces
            minimo = j /* (!) */
        fin si
    fin para
    intercambiar(lista[i], lista[minimo])
fin para
```

Este algoritmo resulta un poco mejor que el de burbuja. En el caso de tener que ordenar un vector de enteros, no visualizamos gran mejora, pero cuando hay que ordenar un vector de estructuras más complejas, la operación intercambiar() sería más difícil en este caso. Este algoritmo realiza muchas menos operaciones intercambiar() que el de burbuja, por lo que lo mejora en algo. Si sustituimos la línea comentada con (!) por intercambiar(lista[i], lista[j]), tendríamos una versión del algoritmo de la burbuja (naturalmente eliminando el orden intercambiar del final).

Otra desventaja de este algoritmo respecto a otros, como el de burbuja o de inserción directa, es que no mejora su rendimiento cuando los datos ya están ordenados o parcialmente ordenados. Por ejemplo, en el caso de la ordenación de burbuja, se requeriría una única pasada para detectar que el vector ya está ordenado y finalizar; por su parte, en la ordenación por selección se realizarían el mismo número de pasadas, independientemente de si los datos están ordenados o no.

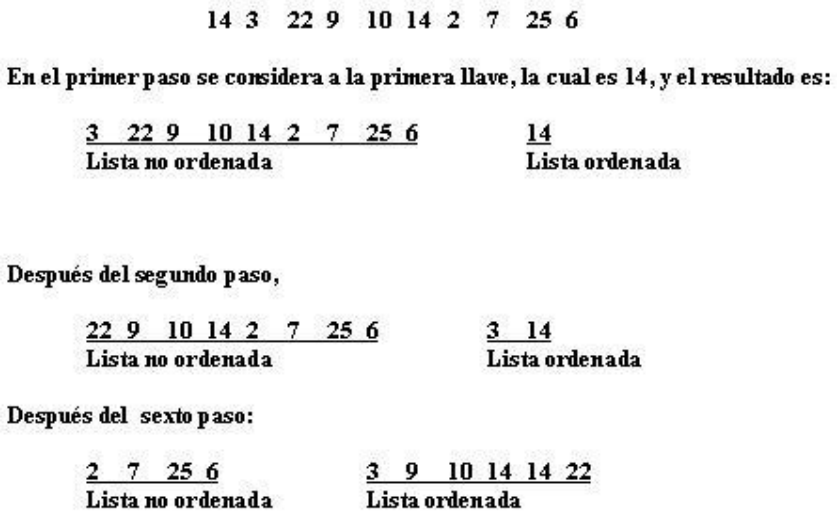
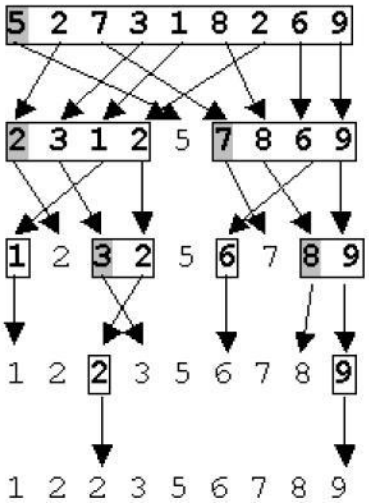


Figura. Algoritmo de selección.
educacioninformatica.com. (n.d.). [Fotografía]. educacioninformatica.com.

El algoritmo, conocido como ***quicksort*** (ordenación rápida), recibe el nombre de su autor, Tony Hoare. Este método o algoritmo es muy sencillo: consiste en la división en particiones de la lista a ordenar, por lo que se puede considerar que aplica la técnica *divide y vencerás*. El método es posiblemente el más pequeño de código, más rápido, más elegante, más interesante y eficiente de los algoritmos de ordenación conocidos. Para enriquecer tu conocimiento acerca de este método, consulta la obra de Goodrich/Tamassia (2010, pp. 467-477), pues explica ampliamente cómo se utiliza este método a través de diferentes ejemplos y extractos de código en java.



Algoritmo *quicksort*.
eui.upm.es. (n.d.). [Fotografía]. eui.upm.es.

2.2. Métodos de búsqueda

En el tema anterior hablamos de los diferentes métodos o algoritmos de ordenación que existen; y en este tema abordaremos la **búsqueda**, la cual complementa la ordenación, ya que, no tendría caso ordenar datos si luego no es posible recobrarlos mediante una buena búsqueda. Por ejemplo, de nada nos serviría ordenar alfabéticamente nuestros contactos en una agenda si luego no vamos a buscar los datos de alguien en particular. Por lo tanto, la ordenación y la búsqueda van estrechamente relacionadas.

Existen diferentes **métodos** o algoritmos de **búsqueda de datos.** Los más conocidos son el **método de búsqueda secuencial**, el **método de búsqueda binaria,** y el **método de búsqueda por interpolación**.

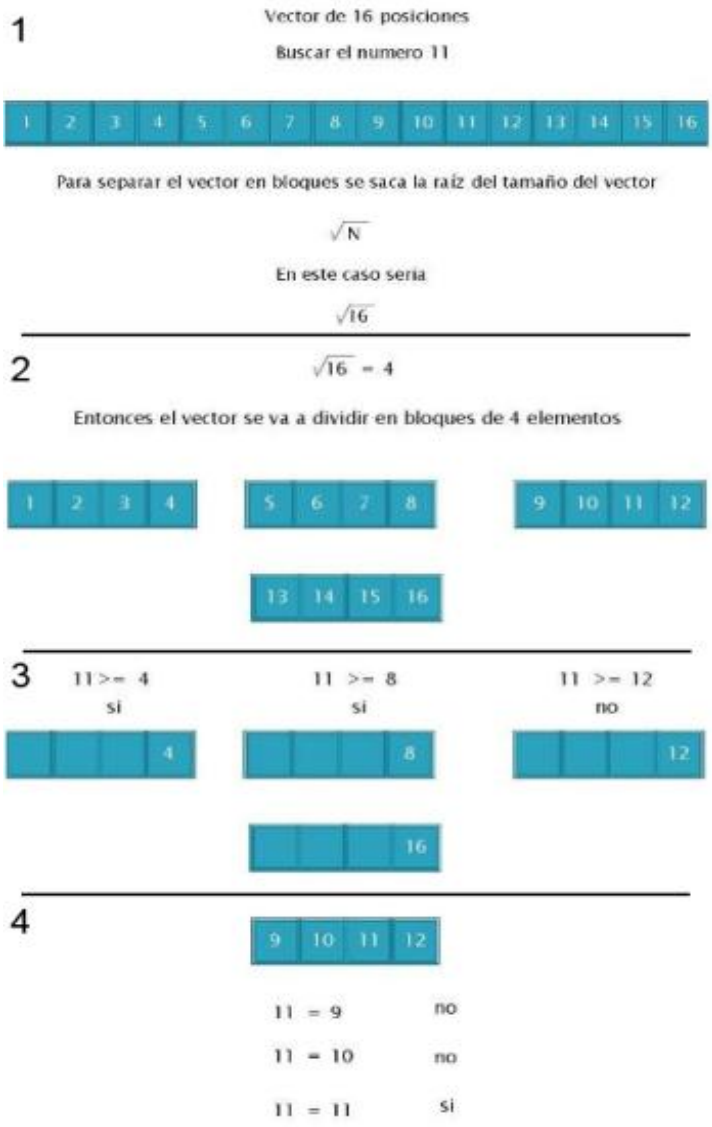
Para abundar más en los diferentes **métodos de búsqueda** que existen, así como en su forma de utilizarlos y en algunos ejemplos, **consulta** el texto de Joyanes (2010, pp. 379-398), en el que encontrarás pseudocódigos y diagramas de flujo que explican de manera clara estos algoritmos para cada método.

Búsqueda de datos en la Sección Amarilla.
sexenio.com.mx. (n.d.). [Fotografía]. http://www.sexenio.com.mx/diario/20110805/7236.jpg

Búsqueda secuencial

Método de búsqueda secuencial, también llamado **búsqueda lineal,** se denomina al método que busca un elemento de una lista utilizando un valor destino llamado clave. En este método, los elementos de una lista o vector se examinan en secuencia: uno después de otro. La búsqueda secuencial es necesaria, por ejemplo, si se desea encontrar a la persona cuyo número de teléfono es 7141414 en el directorio telefónico de la localidad. Los directorios de teléfonos están organizados alfabéticamente por el nombre del titular o dueño de la línea, en lugar de por números de teléfono, de modo que deben examinarse todos los números, uno después de otro, esperando encontrar el número 7141414.

El algoritmo de búsqueda secuencial compara cada elemento del arreglo con la clave de búsqueda. Dado que el arreglo no está en un orden prefijado, es probable que el elemento a buscar pueda ser el primer elemento, el último elemento, o cualquier otro. El programa tendrá que comparar la clave de búsqueda con, por lo menos, la mitad de los elementos del arreglo. El **método de búsqueda lineal** funcionará bien con arreglos pequeños o no ordenados. La eficiencia de la búsqueda secuencial es mínima, y tiene complejidad lineal O(n). **Consulta** a Joyanes/Zahonero (2010, pág. 442) para una breve y concisa explicación de este método de búsqueda.

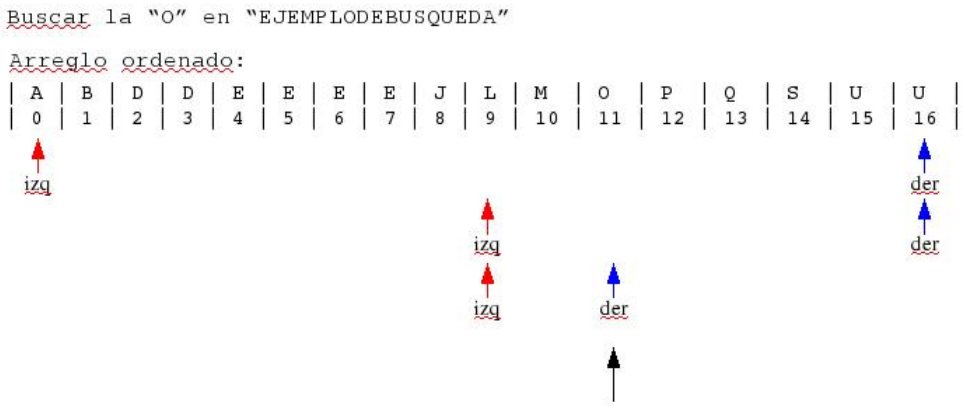


Búsqueda secuencial.
recursosdelweb.com. (n.d.). [Fotografía]. http://www.recursosdelweb.com/

Búsqueda binaria

El **método de búsqueda binaria** proporciona una técnica de búsqueda mejorada, por ejemplo: una palabra en un diccionario; teniendo la palabra a buscar, se abre el diccionario, ya sea al principio, en el medio o al final, dependiendo de la letra con que inicia la palabra.

Generalmente, a menos que tengamos un separador de libros o algo similar, no encontramos la página a la primera, tenemos que estar cambiando de hoja; tal vez estemos muy próximos a la letra inicial buscada. Por ejemplo, si la palabra comienza con “N” y se está en la “P”, se mueve uno hacia atrás. El proceso continúa hasta que se encuentra la página buscada o hasta que se descubre que la palabra no está en la lista. En el texto de Joyanes/Zahonero (2010, pág. 445) se describe el proceso para ejecutar este método de búsqueda.



Búsqueda binaria.
ica.luz.ve. (n.d.). [Fotografía]. ica.luz.ve.

Búsqueda por interpolación

Finalmente, revisaremos el tercer tipo de búsqueda, la **búsqueda por interpolación**.
Este método consiste en tratar de acertar en qué parte del intervalo está la clave que se está buscando, en lugar de dividir ciegamente el arreglo a la mitad. Este método se puede aplicar solamente a tablas o archivos ordenados. Como su nombre lo indica, se trata de llegar al elemento buscado por medio de la interpolación lineal. El procedimiento es recursivo; como en el caso de la búsqueda binaria, en cada paso se van modificando los límites, disminuyendo el intervalo, hasta llegar al elemento buscado.

El algoritmo es similar al de búsqueda binaria. La diferencia está en que, en lugar de dividir el área en mitades, se delimita por medio de los valores resultantes de la interpolación. Joyanes/Zahonero (2010, pág. 445) comenta que la búsqueda por interpolación es más eficiente que la búsqueda binaria, aunque después de todo se usan de forma muy parecida.

Hemos terminado de revisar los diferentes métodos de búsqueda que existen. Es importante conocer el funcionamiento de cada método para aplicar el que mejor se adecúe a la situación planteada. Por lo tanto, sigue las indicaciones que a continuación se te presentan.

Cierre de la unidad

En esta unidad se abordaron diferentes temas de gran interés en la programación: **métodos de ordenación** y **métodos de búsqueda**. Después de revisar los temas de pilas, listas y colas en la unidad anterior, con la presente unidad puedes identificar un complemento a las principales estructuras de datos.

Se partió de cómo se almacenan y manipulan datos según la estructura consecuente. Es decir, se aplica un algoritmo distinto para cada estructura; luego, mediante los diferentes métodos de ordenación, sabrás cómo puedes ordenar los datos que previamente se almacenaron en alguna estructura. Asimismo, aprendiste a manejar diferentes métodos de búsqueda de datos.

Todo eso lleva te lleva a saber que no es útil almacenar información si ésta no es ordenada y posteriormente se realiza una búsqueda. Haciendo una analogía con una agenda telefónica, generalmente se realiza el registro de la información en forma alfabética ascendente porque luego habrá la necesidad de realizar la búsqueda de algún registro en particular. Con los temas revisados en la unidad didáctica *Estructura de datos* serás capaz de realizar programas robustos, como puede ser una agenda telefónica que te permita ordenar los distintos datos, así como realizar búsquedas eficientes.

Para saber más…

Existen diversas fuentes que puedes consultar para ampliar tus conocimientos. Aquí se sugieren algunas:

En esta fuente podrás encontrar información de los temas abordados en esta unidad: ordenación y búsqueda. La página se puede consultar en línea o puedes descargar el documento en formato PDF.

> Chaves Torres, A. N. (2017). *Aprenda a diseñar algoritmos*. https://repository.unad.edu.co/handle/10596/11970

Esta fuente proporciona ejemplos reales que facilitan la comprensión de los diferentes conceptos abordados en la unidad. Entre los diferentes temas que esta fuente ofrece, figuran aquellos tratados en esta unidad, y existen vínculos que te llevarán a cada tema en particular.

López, B. (2022). *Estructuras de datos orientadas a objetos.* Perlego. https://www.perlego.com/es/book/3522238/estructuras-de-datos-orientadas-a-objetos-pdf

Fuentes de consulta

- Goodrich, M. & Tamassia, R. (2010). *Estructura de datos y algoritmos en Java*. CECSA.
- Joyanes, L. (2010). *Fundamentos de programación: Algoritmos, estructuras de datos y objetos*. McGraw-Hill.
- Joyanes, L. & Zahonero, I. (2012). *Programación en Java 2: Algoritmos, estructuras de datos y programación orientada a objetos*. McGraw-Hill.