

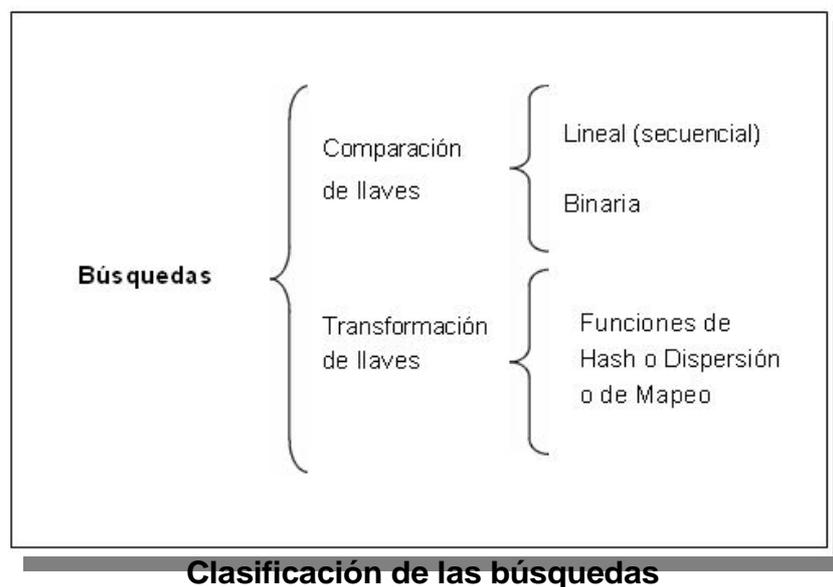


### Introducción a la unidad

Una de las funciones que con mayor frecuencia se utiliza en los sistemas de información, es el de las consultas a los datos, se hace necesario utilizar algoritmos, que permitan realizar búsquedas de forma rápida y eficiente.

La búsqueda, se puede decir que es la acción de recuperar datos o información, siendo una de las actividades que más aplicaciones tiene en los sistemas de información, más formalmente se puede definir como “La operación de búsqueda sobre una estructura de datos es aquella que permite localizar un nodo en particular si es que éste existe”<sup>1</sup>

Una posible clasificación de las búsquedas puede ser:



Siendo la llave el campo por el cual se va a realizar la búsqueda. La búsqueda por comparación de llaves son algoritmos bastante sencillos, en donde se van

<sup>1</sup> Jorge Iván Euán Ávila y Luis Gonzaga Cordero Borboa, *op. cit.*, p. 181.



### Unidad III. Métodos de búsqueda



comparando precisamente las llaves, pero pueden llegar a consumir mucho tiempo cuando se tiene un gran número de ellas, en tanto que el de transformación de llaves los algoritmos deben de ser mucho más analizados, ya que transforma las llaves por varios métodos, indicando en que posición del arreglo o del archivo (acceso directo) lo va a almacenar o a recuperar.

#### Objetivo particular de la unidad

Conocer los métodos más importantes de búsqueda y aplicar el más conveniente al conjunto de datos que se encuentran, ya sea en memoria principal o en la memoria secundaria, así mismo manejará las funciones de dispersión.

#### LO QUE SÉ

Discute en el foro el tema “Ordenamiento de registros en bases de datos y archivos”. Dentro de la discusión del tema plantea la importancia de utilizar diferentes métodos de ordenamiento de registros en bases de datos y diferentes tipos de archivos.

Pulsa el botón **Colocar un nuevo tema de discusión aquí**.

Escribe en el apartado **Asunto** el título de su aportación, redacta tu comentario en el área de texto y da clic en el botón **Enviar al foro**.



## Unidad III. Métodos de búsqueda



### Temas de la unidad III

1. Búsqueda secuencial
2. Búsqueda binaria
3. Búsqueda mediante transformación de llaves (Hashing)
4. Árboles binarios de búsqueda

### Resumen de la unidad

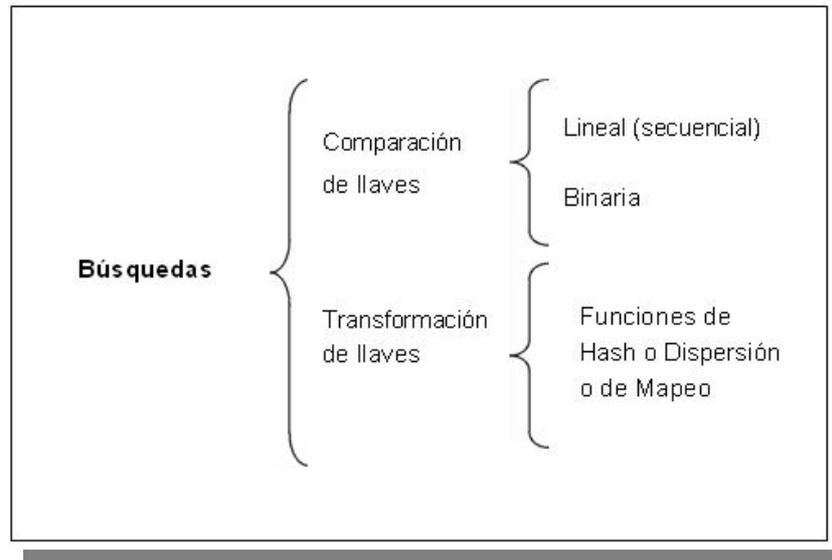
Contar con estructuras de datos es importante tanto para el buen manejo de la información, como para la buena ejecución de los programas de cómputo. Una de las estructuras de datos principales son los archivos y dentro de ellos es posible almacenar grandes cantidades de información que generalmente no está bien organizada o en su caso es muy extensa, lo que dificulta el acceso a un dato en particular. Este problema se incrementa de manera exponencial cuando pasamos del manejo de archivos a bases de datos, que pueden contener millones de veces la cantidad de información contenida en un archivo, con miles o millones de registros que en su momento serán utilizados.

Para poder resolver el problema del acceso a un registro o dato en particular se cuenta con los métodos de búsqueda, que permiten a los programadores implementar programas eficaces de manejo de información que obtengan la información de forma eficiente y rápida.

Los tipos de búsqueda pueden clasificarse de la siguiente manera:



### Unidad III. Métodos de búsqueda



Donde los primeros método de comparación de llaves son los más sencillos de implementar, ya que como su nombre se indica, se dedican a comparar el dato a buscar con cada uno de los registros almacenados en el archivo o en la base de datos, siendo una de sus principales desventajas el tiempo de espera, ya que cuando los registros son pequeños funcionan de forma eficiente, pero van deteriorando su desempeño conforme se incrementa el número de registros a comparar.

La utilización de los métodos por comparación de llaves dependen en gran medida del número de elementos, estos es, a mayor número de elementos son mayores las comparaciones.

Existe el grupo de búsquedas por transformación de llaves (Hash), que aumenta la eficiencia, en cuanto al tiempo de ejecución, ya que accede a los registros por lo general más rápidamente, pero va a depender de su implementación.





### Unidad III. Métodos de búsqueda



Finalmente, es posible auxiliarnos de árboles binarios para la realización de búsquedas, aunque el procedimiento es más complejo que los anteriores son más eficientes en cuanto al manejo de la información.

A pesar del avance de los lenguajes de programación y las técnicas de desarrollo de sistemas de información, las estructuras de datos y los métodos de búsqueda no han cambiado y siguen siendo las bases fundamentales para la creación de los sistemas.



## Unidad III. Métodos de búsqueda



### Tema 1. Búsqueda secuencial.

#### Objetivo del tema

Solucionar problemas de búsquedas en archivos o bases de datos mediante la implementación de búsquedas secuenciales.

#### Desarrollo

Este tipo de búsqueda consiste en examinar, a partir del primer elemento y de uno en uno, hasta encontrar el dato buscado o bien llegar al final de la lista que puede estar almacenada en archivo o arreglo.

En este tipo de listas los elementos pueden o no estar clasificados, ya que se empieza a comparar de uno en uno los elementos de la lista y no importa su orden para realizar la búsqueda, salvo para el tiempo de ejecución.

Si el elemento que se está buscando, se encuentra al inicio de la lista, este tiempo, sería muy corto, pero si se encuentra al final, va a tardar más y si el elemento que se desea buscar, no se encuentra en la lista, se hizo necesario, recorrer toda la lista, para darse cuenta que no está en ella.

Y si se le aumenta a esto, que el número de elementos en la lista puede ser del orden de cientos o miles, va a hacer mucho más tardado su ejecución.

Esta búsqueda tiene la ventaja de tener una fácil programación de su algoritmo.

El pseudocódigo del algoritmo es el siguiente:

```
inicio
    i=0
    bandera = 0
    mientras i < n
        si k [ i ] = x
            desplegar "búsqueda exitosa y el dato es" x
            desplegar "y se encuentra en la posición" i i=n
    fin_mientras
fin_inicio
```



### Unidad III. Métodos de búsqueda



```

bandera = 1
    i=i+1
    fin
fin
si bandera = 0
    desplegar "no se encontró el dato en la lista"
fin
else

```

Para este pseudocódigo, se considera que se tienen n nodos (conjunto de uno o varios campos) en una lista como:

$$n_0, n_1, n_2, \dots, n_{n-1}$$

Cuyas llaves son:  $K_0, K_1, K_2, \dots, K_{n-1}$ , donde n es  $\sim 0$  y x es el valor del dato que se está buscando.

#### Análisis de la búsqueda secuencial

El número de comparaciones tanto en el caso promedio como en el caso más crítico, es un parámetro importante para medir la eficiencia del método. El número promedio de comparaciones en la búsqueda de un elemento es:

$$n/2$$

En el peor de los casos, cuando no se encuentra el elemento, es necesario recorrer toda la lista y entonces se realizan las siguientes comparaciones:

$$N$$

Ejemplos:

Posición dentro del arreglo y/o archivo	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]
	1	3	9	4	2	5	7	8	0	11	15	13	12

Se desean encontrar los siguientes datos:

1  
8  
12



### Unidad III. Métodos de búsqueda



16





## Unidad III. Métodos de búsqueda



### Solución A)

Posición dentro del arreglo y/o archivo [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12]  
1, 3, 9, 4, 2, 5, 7, 8, 0, 11, 15, 13, 12

↑  
ap

2

En la primera comparación se encuentra el elemento 1 (uno) que está en la posición 0 (cero).

### Solución B)

Posición dentro del arreglo y/o archivo [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12]  
1, 3, 9, 4, 2, 5, 7, 8, 0, 11, 15, 13, 12

↑  
ap

3

Se empieza a comparar desde la posición 0 (cero) y se va avanzando de uno en uno la posición, realizando la comparación, hasta que se encuentra el elemento buscado o termina la lista de datos.

Posición dentro del arreglo y/o archivo [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12]  
1, 3, 9, 4, 2, 5, 7, 8, 0, 11, 15, 13, 12

↑  
ap

4

Posición dentro del arreglo y/o archivo [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12]  
1, 3, 9, 4, 2, 5, 7, 8, 0, 11, 15, 13, 12

↑  
ap . . . . .

5

Posición dentro del arreglo y/o archivo [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12]  
1, 3, 9, 4, 2, 5, 7, 8, 0, 11, 15, 13, 12

↑  
ap

En la octava comparación se encuentra el elemento 8 (ocho) que está en la posición 7 (siete).



### Unidad III. Métodos de búsqueda



#### Solución C)

Posición dentro del arreglo y/o archivo [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12]  
 1, 3, 9, 4, 2, 5, 7, 8, 0, 11, 15, 13, 12  
 ↑  
 ap

3

Se empieza nuevamente a comparar desde la posición 0 (cero) y se va avanzando de uno en uno la posición, realizando la comparación, hacia la derecha hasta que se encuentra el elemento buscado o termina la lista de datos.

Posición dentro del arreglo y/o archivo [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12]  
 1, 3, 9, 4, 2, 5, 7, 8, 0, 11, 15, 13, 12  
 ↑  
 ap

4

Posición dentro del arreglo y/o archivo [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12]  
 1, 3, 9, 4, 2, 5, 7, 8, 0, 11, 15, 13, 12  
 ↑  
 ap

7

Posición dentro del arreglo y/o archivo [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12]  
 1, 3, 9, 4, 2, 5, 7, 8, 0, 11, 15, 13, 12  
 ↑  
 ap

8

El elemento 12 (doce) está en la posición 12 (doce), después de trece comparaciones.

#### Solución D)

Posición dentro del arreglo y/o archivo [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12]  
 1, 3, 9, 4, 2, 5, 7, 8, 0, 11, 15, 13, 12  
 ↑  
 ap

3

Se empieza nuevamente a comparar desde la posición 0 (cero) y se va avanzando de uno en uno la posición, realizando la comparación, hacia la

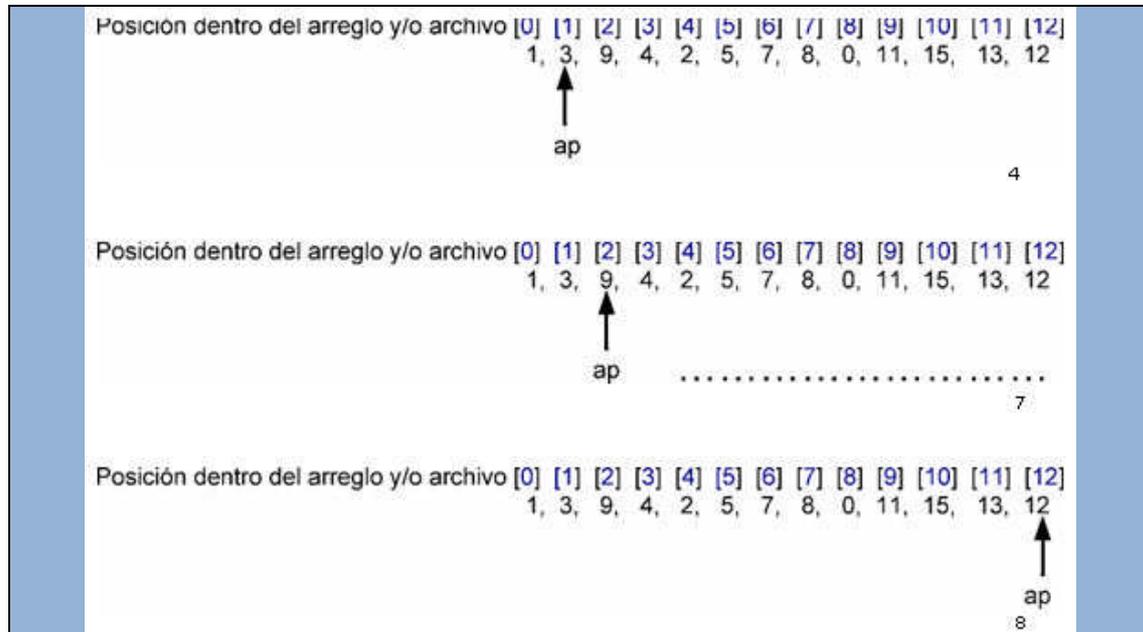




### Unidad III. Métodos de búsqueda



derecha hasta que se encuentra el elemento buscado o termina la lista de datos.



El elemento 16 (dieciséis) no se encuentra en la lista, después de trece comparaciones y de haber recorrido toda la lista.

#### ACTIVIDAD 1

Elabora un programa en lenguaje C, tomando como base el pseudocódigo presentado en éste tema para implementar una búsqueda secuencial. Sube a la plataforma tu archivo con el código fuente y el ejecutable para su revisión.

Utilice un editor de texto, para realizar su actividad, una vez que concluya envíe su información; presione el botón **Examinar** para localizar el archivo en su computadora y por último presione **Subir este archivo**.



### Unidad III. Métodos de búsqueda



#### ACTIVIDAD 2

Relaciona las siguientes columnas.

<ul style="list-style-type: none"> <li><input type="checkbox"/> Es una de las principales desventajas de las búsqueda secuencial cuando la cantidad de datos es muy grande</li> <li><input type="checkbox"/> La simpleza del método se ve reflejada en</li> <li><input type="checkbox"/> El número promedio de comparaciones en la búsqueda de un elemento es:</li> <li><input type="checkbox"/> Cada pasada realizada en una búsqueda secuencial comienza en el elemento</li> <li><input type="checkbox"/> El número de búsquedas realizadas en el peor de los casos en el método secuencial es</li> </ul>	<ul style="list-style-type: none"> <li><b>1</b> <math>n/2</math></li> <li><b>2</b> Tiempo de ejecución.</li> <li><b>3</b> 0</li> <li><b>4</b> <math>n</math></li> <li><b>5</b> Tiempo de programación</li> </ul>
---	--

#### Bibliografía básica

Autor	Capítulo	Páginas

#### Sitios electrónicos

Sitio	Descripción





## Unidad III. Métodos de búsqueda



### Tema 2. Búsqueda binaria.

#### Objetivos del tema

Solucionar problemas de búsquedas en archivos o bases de datos mediante la implementación de algoritmos de búsqueda binaria.

#### Desarrollo

La búsqueda Binaria o por Bisección no representa mucha dificultad para la programación de su algoritmo y además, es muy rápida su ejecución.

Este algoritmo requiere que los elementos de la lista, sobre la que va a actuar, estén clasificados, ya sea en forma ascendente o descendente, cada elemento de la lista puede tener varios campos. La lista se considera que empieza a almacenar sus elementos en la posición **ceros**.

Va a utilizarse tres apuntadores, uno en la primera posición de la lista que se le denominara LI, para efectos de la explicación, otro en la última conocido como LS y el que apunte en la parte central, el cual se obtiene de la suma de LS mas LI entre dos ( $LI + LS/ 2$ ) y tomando la parte entera, el cual se le llamará M.

A diferencia de la Búsqueda Secuencial, aquí el número de comparaciones no se comporta en forma lineal, ya que procede a realizar los siguientes pasos:

- |   |
|---|
| <ul style="list-style-type: none"><li>• Dividir la lista en dos partes, al determinar el elemento central de dicha lista, con lo que se iniciará el apuntador M.</li></ul>                          |
| <ul style="list-style-type: none"><li>• Comparar el valor del elemento buscado con el central.</li></ul>  |
| <ul style="list-style-type: none"><li>• Si resultan ser iguales, las búsquedas termina con éxito, indicando en qué posición se encontró y cuáles son los datos que están en esa posición.</li></ul> |
| <ul style="list-style-type: none"><li>• En el caso de no ser iguales, se redefinen la posición de alguno de los</li></ul>   |



### Unidad III. Métodos de búsqueda



apuntadores de los extremos (LI o LS), dependiendo del valor del elemento central, sea mayor o menor que el buscado. Por ejemplo, suponiendo que la lista está clasificada en forma ascendente, se pueden presentar los siguientes casos:

+ Que el elemento buscado sea mayor que el elemento central (apuntado por M), entonces se desechará la primera mitad de la lista ya que por estar clasificada en forma ascendente se sabe que ahí no va a estar el elemento, con lo que se moverá el apuntador de inicio que es LI a la posición siguiente a donde está el elemento central que esta apuntado por M. Procediendo a considerar esta nueva lista, que es la segunda mitad, como la nueva lista.

+ Que el elemento buscado ahora sea menor que el elemento apuntador por M, entonces, se desechará la segunda mitad de la lista y el apuntador que se actualizará será entonces LS, que es el que apuntaba al último elemento y ahora se moverá a una posición inmediata anterior a donde está apuntando M.

En ambos casos se aplicaran nuevamente los puntos anteriores del método, desde el primer paso, hasta que el elemento sea encontrado o bien que la lista que vaya quedando como resultado, quede vacía.

El pseudocódigo del algoritmo es el siguiente:

```
inicio
  LI = 0
  LS = n - 1
  mientras LI ~LS
    M = Parte Entera (LI + LS) /2
    si dato < lista[M]
      LS = M -1
    en caso contrario si dato > lista [M]
      LI = M +1
    en caso contrario
      desplegar "el elemento buscado esta en la posición", M
      break
  fin
  desplegar "el elemento" dato "no se encontró en la lista"
fin
```



## Unidad III. Métodos de búsqueda



**Dato**, es el valor que se va a buscar en la lista; **lista**, es donde están los elementos clasificados, **LI** es el límite inferior, **LS** es el límite superior, **M** es el apuntador de en medio y **N** el número de elementos de la lista.

### Análisis de la Búsqueda Binaria

Este método en cada iteración el intervalo de búsqueda se divide por mitad, por lo que el número de comparaciones tiende a:

$$\log_2 n$$

Leer archivo: **TEMA 2 (ANEXO 1)**

#### ACTIVIDAD 1

Realiza un programa en el lenguaje de programación C para implementar la búsqueda binaria, partiendo del pseudo código visto en este tema. Sube a la plataforma tus archivos con el código fuente y el ejecutable para su revisión.

Utilice un editor de texto, para realizar su actividad, una vez que concluya envíe su información; presione el botón **Examinar** para localizar el archivo en su computadora y por último presione **Subir este archivo**.

#### ACTIVIDAD 2

Participa en el foro de discusión “Búsquedas sobre registros”, en donde se plantearán las principales ventajas y desventajas de la utilización de búsquedas binarias y secuenciales.

Utilice un editor de texto, para realizar su actividad, una vez que concluya envíe su información; presione el botón **Examinar** para localizar el archivo en su computadora y por último presione **Subir este archivo**.



### Unidad III. Métodos de búsqueda



#### Autoevaluación

Relaciona las siguientes columnas, arrastrando la frase que complete cada oración sobre las preguntas. Al final obtendrás tu calificación de manera automática.

<ul style="list-style-type: none"> <li><input type="checkbox"/> La búsqueda binaria se realiza de forma:</li> <li><input type="checkbox"/> El número de búsquedas en el método binario tiende a</li> <li><input type="checkbox"/> La posición del apuntador central en el método binario se calcula de la forma:</li> <li><input type="checkbox"/> Un requerimiento del algoritmo de la búsqueda binaria es:</li> </ul>	<ul style="list-style-type: none"> <li><b>1</b> Elementos clasificados</li> <li><b>2</b> No lineal</li> <li><b>3</b> <math>M = \text{Parte entera } [(LI+LS)/2]</math></li> <li><b>4</b> <math>\text{Log}_2 n</math></li> </ul>
---	---

#### Bibliografía básica

Autor	Capítulo	Páginas

#### Sitios electrónicos

Sitio	Descripción





## Unidad III. Métodos de búsqueda



### Tema 3. Búsqueda mediante transformación de llaves (Hashing).

#### Objetivos del tema

Diferenciar formas de realizar búsquedas sobre archivos o bases de datos empleando las funciones de transformación de llaves.

#### Desarrollo

La utilización de los métodos por comparación de llaves son muy dependientes del número de elementos, estos es, a mayor número de elementos son mayores las comparaciones.

Existe el grupo de búsquedas por transformación de llaves (Hash), que aumenta la eficiencia, en cuanto al tiempo de ejecución, ya que accede a los registros por lo general más rápidamente, pero va a depender de su implementación.

Entre las ventajas que trae consigo están:

- Aumentar la velocidad de ejecución, independientemente del número de elementos que contenga la lista.
- Los elementos de la lista donde se realiza la búsqueda no forzosamente deben estar clasificados. La lista de datos, puede estar en un archivo o en un arreglo.
- No se necesita recorrer todos los registros anteriores (como en la búsqueda secuencial) o realizar muchas comparaciones (como podría llegar a ser en la búsqueda binaria). Idealmente encontraría directamente el registro que se está buscando.



## Unidad III. Métodos de búsqueda



Entre las desventajas se encuentran:

- Tener que definir un campo llave, que recordando del primer capítulo, es aquel que aparece en todos los registros o nodos de la lista, pero cuyo contenido no se repite.
- La eficiencia de este tipo de búsquedas, va a depender de que tan robusta sea la función HASH, que va a transformar la llave para indicar la posición en la cual se encuentra el elemento buscado, lo ideal es que sea de uno a uno, esto es que para cada valor de llave, esta función la transforme e indique en qué posición va encontrar ese registro.

La transformación de llaves o claves, permite tener acceso a los elementos de manera directa, esto es sin tener la necesidad de recorrer los elementos anteriores, antes de llegar al buscado. Algunos autores también la llaman función de mapeo, ya que transforma o mapea la llave de entrada.

Trabaja en base a la función de transformación HASH que comúnmente se denomina como H (llave), la cual va a convertir, una llave dada, en una dirección dentro del arreglo o del archivo.

### **Función Hash**

Las funciones de Hash, para lograr su objetivo de proporcionar la dirección donde debe estar el elemento buscado a partir de su llave, lo logran haciendo algunas operaciones sobre los caracteres que la componen.

Es importante hacer notar que la misma función de Hash, que se utiliza, para almacenar, es la misma que se ocupa para recuperar la llave (considerando que sólo se está almacenando por la llave, pero que puede ser un registro o estructura que puede contener varios campos).



### Unidad III. Métodos de búsqueda



Cuando se transforma la llave por el método de suma, por ejemplo al sumar el primer dígito con el tercero, dará como resultado la dirección donde se almacenará el registro, pero se puede presentar el caso de que la dirección ya esté ocupada por otro elemento, como sería el siguiente caso.

La función de Hash (primer dígito + tercer dígito) aplicado a las siguientes llaves:

$$K1=123 \text{ y } K2=292$$

al aplicar la función de Hash se tendría:

$$\begin{aligned} H(K1) &= 4 \\ H(K2) &= 4 \end{aligned}$$

Lo que implicaría que estas dos llaves, con valores diferentes, deberían de estar en la misma dirección, a este suceso se le conoce como **colisión**.

Existen diferentes formas de resolver las colisiones entre las más comunes se tienen las de **direccionamiento abierto** y las de **encadenamiento**.

Para utilizar las búsquedas por transformación de llaves se recomienda considerar las siguientes características:

- Que la función de Hash sea fácil de calcular e implementar.
- También la función de Hash logre la distribución de las llaves lo más uniformemente en la lista (archivo o arreglo).
- Métodos eficientes en para resolver colisiones en caso de que se presenten.

Las funciones de Hash pueden implementarse de acuerdo al tipo de llaves que se van a manejar, por ejemplo, si se va a utilizar el número de folio de las facturas, que emite una compañía dada y estos folios se van a utilizar como llaves, suponiendo que varían del 1 al 99, entonces las llaves convendrían que fueran de tipo numérico y fueran el folio completo.



### Unidad III. Métodos de búsqueda



Pero si el valor de la llave estuviera en un rango, en donde no se ocuparan todos los valores, por ejemplo el número de cuenta de los clientes de tarjetas de crédito de un banco, donde aparecen una gran cantidad de dígitos y pueden que no sean consecutivos, ya que los dos primeros dígitos pueden indicar a lo mejor el número de la sucursal, los otros dos, el tipo de tarjeta (por ejemplo Master Card®, Visa®, etc.), los otros dos, a lo mejor el tipo de moneda (pesos, dólares, euros, etc.), entonces no se podría utilizar toda la llave, como función de Hash, ya que habría mucho desperdicio de memoria, por los rangos que no se llegan a utilizar.

El peor de los casos es cuando en la llave se encuentren caracteres que no sólo sean numéricos, como podrían ser también letras como por ejemplo el Registro Federal de Contribuyentes (RFC) o la Clave Única del Registro de Población (CURP), no se podría usar directamente como la dirección donde se almacenara o recuperará, por lo que será necesario realizar algunas conversiones, para manejarla numéricamente, como el asignarle un número a cada letra, ya sea por la posición dentro del abecedario o bien por el valor del código, en el cual se encuentre representado que los más comunes son el ASCII y el EBCDIC.

Entre las transformaciones que se realizan sobre las llaves, se encuentran algunas funciones que pueden involucran desde las operaciones básicas de sumas, restas, multiplicaciones y divisiones, así como las de módulo, cuadrado, cambio de base entre otras. En general las funciones de Hash, el mapear una llave por la dirección, indica en qué dirección debe almacenarse el registros a que pertenece esa llave. Para transformar estas llaves, las funciones de Hash, puede seccionar toda o parte de las llaves, siendo estas partes sobre las que se aplicarán las operaciones antes mencionadas.



### Unidad III. Métodos de búsqueda



Como las operaciones que se utilizan para la transformación puede darse el caso de que den un valor de cero, se debe contemplar la dirección cero en el archivo o en el arreglo si se utiliza un lenguaje de programación que maneje arreglos con índices a partir de cero, como lo hace C, no hay problema, pero si se trata de arreglos como los que utilizan otros lenguajes como FORTRAN, que empiezan en el índice uno será necesario, al resultado de la transformación sumarle uno, para evitar los índices inválidos.

A continuación, se describen algunas de las funciones de Hash más utilizadas.

Leer archivo: **TEMA 3** (ANEXO 2)

#### ACTIVIDAD 1

Elabora un mapa conceptual sobre las principales características de las funciones Hash. Sube tu documento a la plataforma para su revisión.

Utilice un editor de texto, para realizar su actividad, una vez que concluya envíe su información; presione el botón **Examinar** para localizar el archivo en su computadora y por último presione **Subir este archivo**.

#### ACTIVIDAD 2

Realiza una breve investigación sobre las aplicaciones de las funciones Hash en los sistemas informáticos. Elabora un documento en Word de máximo 5 cuartillas, donde incluyas una reseña de tu investigación, tus conclusiones y la bibliografía y/o referencias utilizadas. Sube tu documento a la plataforma para su revisión.

Utilice un editor de texto, para realizar su actividad, una vez que concluya envíe su información; presione el botón **Examinar** para localizar el archivo en su computadora y por último presione **Subir este archivo**.



## Unidad III. Métodos de búsqueda



### ACTIVIDAD 3

En un documento en Word, describe en qué consiste cada uno de los métodos utilizados para resolver las colisiones. Sube tu documento a la plataforma.

Utilice un editor de texto, para realizar su actividad, una vez que concluya envíe su información; presione el botón **Examinar** para localizar el archivo en su computadora y por último presione **Subir este archivo**

### Autoevaluación

Relaciona las siguientes columnas.

<input type="checkbox"/> Es una desventaja del método de transformación de llaves	<b>1</b> Encadenamiento.
<input type="checkbox"/> Cuando dos llaves con valor diferente se encuentran en una misma posición se denomina	<b>2</b> . Colisión.
<input type="checkbox"/> Es un método para resolver colisiones de llaves	<b>3</b> Truncamiento
<input type="checkbox"/> La función Hash que consiste en la separación de cada uno de los dígitos que componen a la llave y tomar algunos de ellos para formar la dirección que sería la salida de la función, recibe el nombre de:	<b>4</b> Definición de llaves.
	<b>5</b> Elementos no clasificados.



## Unidad III. Métodos de búsqueda



### Bibliografía básica

Autor	Capítulo	Páginas

### Sitios electrónicos

Sitio	Descripción



## Unidad III. Métodos de búsqueda



### Tema 4. Árboles binarios de búsqueda.

#### Objetivo del tema

Reconocer la importancia de la utilización de árboles binarios en la implementación de métodos de búsqueda sobre archivos o bases de datos.

#### Desarrollo

Otra de las herramientas básicas para realizar búsquedas, es a través de los árboles binarios de búsqueda, para lo cual se recordarán algunos conceptos, como la definición de árboles y de grafos, a continuación se explican detalladamente:

#### Grafo o gráfica o diagrama

La forma más conocida de representar un grafo es por medio de un dibujo donde se tienen los siguientes elementos:

- Los puntos, también llamados vértices, nodos, o unión.
- Las líneas que unen a los puntos, conocidas como arcos, ramas o elementos.

Para representar en la memoria de una computadora al grafo  $G = (E, V)$  será necesario utilizar dos conjuntos de elementos;

Uno para los arcos

$$E=(e1,e2,....en)$$

Otro para los vértices

$$V=(v1,v2,....vn)$$



### Unidad III. Métodos de búsqueda



Por ejemplo:

$V = (v1, v2, v3, v4, v5, v6, v7, v8)$

$E = (e1, e2, e3, e4, e5, e6, e7, e8)$

#### Árboles.

Un árbol es un grafo que tiene las siguientes características:

<ul style="list-style-type: none"><li>▪ Es un grafo simple y finito, ya que no tiene ni ciclos, ni arcos paralelos y tiene un conjunto finito de nodos.</li></ul>
<ul style="list-style-type: none"><li>▪ Hay un nodo específico llamado raíz y es el único que no tiene ningún nodo del cual descienda, esto es, el nodo del cual descienden todos los demás que componen al árbol.</li></ul>
<ul style="list-style-type: none"><li>▪ El resto de los nodos están separados en <math>n</math> conjuntos desordenados, denominados <math>T1</math>, <math>T2</math>, <math>T3</math>, . . . <math>T</math>, donde <math>n</math> es mayor o igual a cero. Cada uno de estos conjuntos son a su vez árboles, llamados subárboles de la raíz. Esta parte de la definición es recursiva.</li></ul>
<ul style="list-style-type: none"><li>▪ Existe sólo un patrón entre cada par de vértices. Un patrón es un conjunto de arcos y nodos, que parten de un origen y llegan a otro nodo destino, en el cual ni vértices ni arcos aparecen más de una vez.</li></ul>
<ul style="list-style-type: none"><li>▪ El número de vértices <math>n</math> que contiene es igual a <math>n-1</math> arcos.</li></ul>
<ul style="list-style-type: none"><li>▪ La suma de los grados de todos los vértices es <math>2(n-1)</math>, donde <math>n</math> es el número de vértices del grafo.</li></ul>



## Unidad III. Métodos de búsqueda



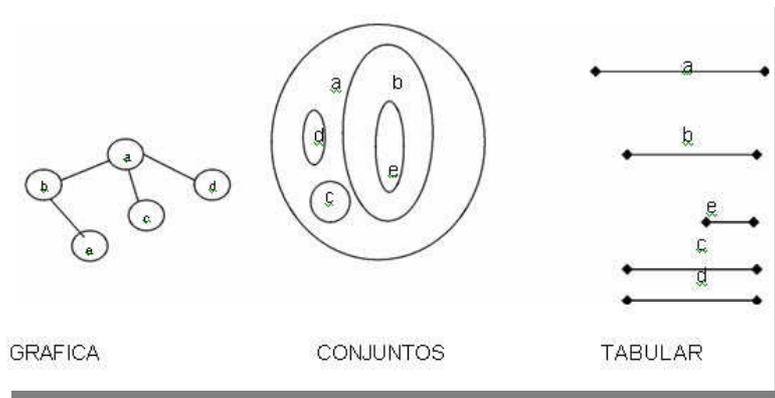
### Niveles del árbol

Un árbol es como un árbol genealógico en donde a cada generación se le conoce como nivel. De todos los nodos que componen al árbol, existe uno que no tiene antecesor, que se llama raíz, el cual para algunos autores es el nivel cero y para otros el nivel uno, también vale la pena mencionar que el nivel de un nodo es igual al nivel de su antecesor más uno.

Los nodos que tienen descendientes se les conocen como nodos padres y a los descendientes como nodos hijos. Un nodo padre puede tener 0 (cero) o varios descendientes, pero un nodo descendiente no puede tener más que a un solo nodo padre. No existe ninguna relación entre los nodos del mismo nivel, solo hacia sus descendientes o ascendentes.

### Representación de árboles

Un árbol se puede representar en forma gráfica ya sea en forma de barras o de conjuntos.

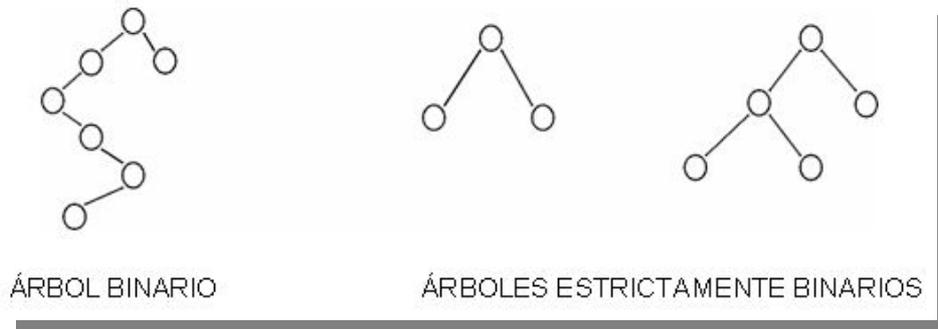


### Árbol de Knuth y estrictamente binario

Un árbol es binario o de Knuth cuando puede tener cero, uno o dos descendientes, y si sólo puede tener cero o dos descendientes se llama estrictamente binario.



## Unidad III. Métodos de búsqueda



### Árboles binarios

Tienen varias aplicaciones como:

- ∅ Implementación de diferentes algoritmos como el de clasificación del Heap, visto con anterioridad.
- ∅ Aplicaciones de búsqueda binaria.

Para los árboles estrictamente binarios y binarios, se pueden ocupar los cinco tipos de recorridos clásicos, que son los que describen los siguientes algoritmos:

- **Arriba-abajo.**- el recorrido es partiendo de la raíz hacia abajo y de izquierda a derecha.
- **Abajo-arriba.**- parte del nivel más bajo hacia arriba y de izquierda a derecha.
- **Preorden.**- visitar raíz, recorrer rama izquierda y recorrer rama derecha.
- **Inorden.** - recorrer rama izquierda, visitar raíz y recorrer rama derecha.
- **Posorden.** - recorrer rama izquierda, recorrer rama derecha y visitar raíz.

Estos recorridos son utilizados en algunas de las aplicaciones anteriormente mencionadas.

Cuando el algoritmo indica que se visita la raíz es cuando se recupera el dato de ese nodo y cuando se menciona que se recorra alguna de las ramas izquierda o derecha, se realiza, siempre y cuando existan esas ramas del árbol.



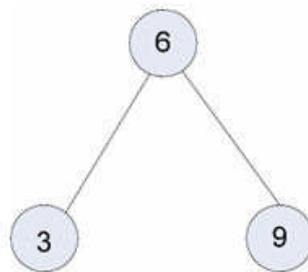
## Unidad III. Métodos de búsqueda



### Árboles de búsqueda binaria (ABB)

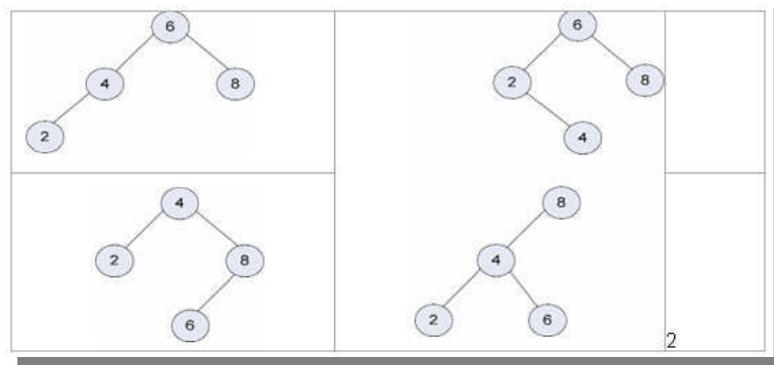
Es un árbol binario que cumple con las siguientes características:

- El dato que contiene cada uno de sus nodos no debe de estar repetido
- El valor del dato que contiene la raíz de la rama izquierda (si es que existe), debe de tener un valor menor al de la raíz original
- El dato de la rama derecha es mayor al de la raíz



Como se observa la raíz del árbol, es un valor mayor al de la raíz de su rama izquierda, pero menor al valor de la raíz de la rama derecha, esta característica es la que se aprovecha en las búsquedas.

El árbol binario de búsqueda para un conjunto de elementos puede representarse de diferentes formas, como se muestra en el siguiente ejemplo.





## Unidad III. Métodos de búsqueda



El recorrido Inorden de cualquier de estos árboles binarios de búsqueda, produce la misma secuencia clasificada en forma ascendente.

### Operaciones

Existen diferentes operaciones que se pueden realizar en este tipo de árboles, entre las que se encuentran:

- > Búsquedas.
- > Inserciones.
- > Bajas
- > Búsquedas

inicio

- si el árbol esta vacío entonces
  - desplegar “árbol vacío y no se encuentra el elemento”
- en caso contrario
  - si el valor del dato del nodo raíz es igual al del elemento que se busca
    - desplegar “búsqueda con éxito”
  - en caso contrario
    - si el valor del dato buscado es  $<$  al valor del nodo raíz
- se continúa la búsqueda en el subárbol izquierdo
  - (llamándose recursivamente este algoritmo)
- en caso contrario
  - si el valor del dato que se desea encontrar es  $>$  que el valor del nodo raíz entonces
    - búsqueda se realizará en el subárbol derecho

Si el elemento buscado se encuentra en alguna dirección se puede dar a través de un apuntador que la indique, pero si no lo encuentra entonces se tendrá la dirección de NULL (Nulo), en dicho apuntador.

Leer archivo: **TEMA 4** (ANEXO 3)



### Unidad III. Métodos de búsqueda



#### ACTIVIDAD 1

Elabora en un documento en Word o en una diapositiva en Power Point, un mapa conceptual o un cuadro sinóptico sobre el tema de árboles de búsqueda. Sube el documento a la plataforma para su revisión.

Realiza tu actividad en un procesador de textos, guárdela en tu computadora y una vez concluyas, presiona el botón **Examinar**. Localiza el archivo, ya seleccionado, presiona **Subir este archivo** para guardarlo en la plataforma

#### ACTIVIDAD 2

Realiza una investigación sobre las aplicaciones de los árboles binarios de búsqueda en el manejo de las bases de datos. Escribe una síntesis de tu investigación en Word junto con tus conclusiones y bibliografía. Sube el documento a la plataforma para su revisión.

Realiza tu actividad en un procesador de textos, guárdela en tu computadora y una vez concluyas, presiona el botón **Examinar**. Localiza el archivo, ya seleccionado, presiona **Subir este archivo** para guardarlo en la plataforma



## Unidad III. Métodos de búsqueda



### ACTIVIDAD 3

Implementa un programa en C para desarrollar la búsqueda en un árbol de búsqueda binaria, utilizando la sentencia struct. Sube a la plataforma tus archivos de código fuente y ejecutable para su revisión.

Realiza tu actividad en un procesador de textos, guárdela en tu computadora y una vez concluyas, presiona el botón **Examinar**. Localiza el archivo, ya seleccionado, presiona **Subir este archivo** para guardarlo en la plataforma



### Unidad III. Métodos de búsqueda



#### Autoevaluación

Relaciona las siguientes columnas.

<ul style="list-style-type: none"><li><input type="checkbox"/> Nodo de un árbol que no tiene ancestros:</li><li><input type="checkbox"/> Se trata de una representación gráfica de un árbol:</li><li><input type="checkbox"/> Forma de recorrido de un árbol que comienza con visitar raíz, recorrer rama izquierda y recorrer rama derecha</li><li><input type="checkbox"/> Recorrido que consta de recorrer rama izquierda, recorrer rama derecha y visitar raíz.</li><li><input type="checkbox"/> Es una operación que puede ser realizada en un árbol binario</li></ul>	<ul style="list-style-type: none"><li><b>1</b> Preorden</li><li><b>2</b> Raíz</li><li><b>3</b> Baja</li><li><b>4</b> Conjunto</li><li><b>5</b> Posorden</li></ul>
---	---





## Unidad III. Métodos de búsqueda



### Cuestionario

1. ¿Qué es una colisión?
2. ¿Qué es una función de Hash?
3. ¿Con qué métodos se resuelven las colisiones?
4. ¿Cómo se define un Árbol de Búsqueda Binaria?
5. ¿Cómo se llama el recorrido, que se utiliza en un Árbol de Búsqueda Binaria, que siempre muestra los nodos en forma ascendente?
6. ¿Los métodos de búsqueda se dividen en?
7. ¿Cómo se llaman los métodos de búsqueda por comparación de llaves?
8. ¿Qué método de búsqueda, es independiente del número de elementos, sobre los que va a actuar?
9. En la función de Hash por la operación cambio de base, ¿cuál es la recomendación, para reducir el número de colisiones, en cuanto a el manejo del tamaño de las bases?
10. ¿Cuál es el método de búsqueda que requiere que los elementos, estén previamente clasificado, en forma ascendente o descendente?

Realiza tu actividad en un procesador de textos, guárdela en tu computadora y una vez concluyas, presiona el botón **Examinar**. Localiza el archivo, ya seleccionado, presiona **Subir este archivo** para guardarlo en la plataforma



### Unidad III. Métodos de búsqueda



#### Examen de autoevaluación

Relaciona la columna de la izquierda con la de la derecha y coloca la respuesta que consideres correcta dentro del paréntesis.

(     ) 1.   Árbol cuya rama izquierda tiene un valor menor y la rama derecha mayor al de su raíz	a) Árbol binario
(     ) 2.   Son los elementos de un Grafo	b) Encadenamiento
(     ) 3.   Es un grafo que sólo puede tener cero, uno o dos descendientes	c) Búsqueda binaria.
(     ) 4.   Es la acción de localizar un nodo o elemento en particular dentro de una lista (archivo o arreglo).	d) Colisión
(     ) 5.   Técnica para resolver las colisiones, que revisa la tabla en forma secuencial, hasta encontrar el elemento deseado o un espacio libre	e) Búsqueda por comparación de llaves lineal
(     ) 6.   Técnica para resolver las Colisiones, que utiliza una lista simplemente ligada	f) Árboles.
(     ) 7.   Es una búsqueda, donde es necesario que las llaves estén clasificadas ascendente o descendientemente	g) Árbol binario de búsqueda
(     ) 8.   Este tipo de búsqueda, puede o no tener las llaves clasificadas	h) Direccionamiento abierto
(     ) 9.   Ocurre cuando al utilizar una función de Hash, esta le asigna a un nodo una dirección que ya ha sido ocupado por otro diferente	i) Búsqueda





### Unidad III. Métodos de búsqueda



( ) 10. Las representaciones gráfica, de conjuntos y tabular sirven para representar	j) Punto y líneas.
--	--------------------

#### Bibliografía básica

Autor	Capítulo	Páginas

#### Sitios electrónicos

Sitio	Descripción

#### LO QUE APRENDÍ

Participa en el foro de discusión sobre “Uso de métodos de ordenamiento en registros de archivos”, en donde se discutirá qué métodos de ordenamiento son más adecuados para trabajar con los distintos tipos de archivos existentes.

Pulsa el botón **Colocar un nuevo tema de discusión aquí**.

Escribe en el apartado **Asunto** el título de su aportación, redacta tu comentario en el área de texto y da clic en el botón **Enviar al foro**



## Unidad III. Métodos de búsqueda



### Glosario de la unidad

#### Acceso aleatorio

O acceso directo. Es un método de acceder a los datos que, a diferencia del acceso secuencial, permite acceder a cualquier dato sin pasar por los precedentes. Usa direcciones calculadas.

#### Actualización

Muy usado en informática y de significado similar al uso coloquial. Revisión del contenido de un programa o de un archivo.

#### Acumulador

Es uno de los elementos de la unidad aritmético-lógica de todo ordenador. En programación es una variable usada para dar totales de datos similares leídos.

#### Algoritmo

Es una descripción de los pasos de una tarea.

#### Análisis

Con los términos de análisis de sistemas, funcional y orgánico se entienden las tareas previas a la programación que sirven para dar solución informática a un problema.

#### Analógico

Aplicable a circuito u ordenador. Contrasta con el término digital en que aquél trata magnitudes que varían en rango continuo, mientras que los digitales operan sobre magnitudes discretas de forma binaria.

#### Árbol

Es uno de los posibles tipos de estructurar los datos. Todos los elementos menos uno, (el raíz), tienen un antecedente. Todos pueden tener otros elementos asociados llamados consecuentes.



## Unidad III. Métodos de búsqueda



### **Ascii**

American standard code for information interchange. Es un código muy utilizado para el intercambio de datos. Como los ordenadores solo procesan dígitos binarios, cada letra, dígito o carácter especial tiene su correspondiente código binario.

### **Asignación**

Usado en programación. Es la acción de poner en una variable un valor, el contenido de otra variable o el resultado de evaluar una expresión. En la mayoría de los lenguajes se representa con el signo igual.

### **Base de datos**

Es un conjunto de datos estructurados de forma que se hagan manejables de forma sencilla a pesar de su volumen o de la complejidad de las relaciones entre ellos.

### **Basic**

(Beginner's all-purpose symbolic instruction code). Código de instrucciones simbólicas multiuso para principiantes. Es un lenguaje de programación diseñado en 1964 para la enseñanza de la programación.

### **Batch**

Modo de procesamiento de programas. El ordenador por medio del sistema operativo gestiona la ejecución de una cola de programas sin interacción con el usuario que los mandó ejecutar.

### **Bifurcación**

En programación: transferencia de la secuencia de ejecución a otra parte del programa. Puede ser condicional o incondicional aunque esta última está mal vista en los lenguajes estructurados.

### **Binario**





## Unidad III. Métodos de búsqueda



Sistema de numeración en base dos. Sus únicos dígitos son el 0 y el 1. Los ordenadores usan internamente este código.

### **Bit**

Contracción de binary digit. Es la unidad elemental de información en el ordenador. Su valor es 0 ó 1 puesto que todos los dispositivos, interruptores y unidades de almacenamiento disponen únicamente de dos estados.

### **Bucle**

En un programa es un grupo de instrucciones que se repiten hasta que se cumple una condición.

### **Buffer**

Área de almacenamiento temporal usado en la transferencia de datos entre la memoria principal y los periféricos.

### **Bus**

Conexión en paralelo por donde circulan los datos, las direcciones o las señales de control en el interior del ordenador. Todos los componentes del ordenador están conectados mediante buses.

### **Byte**

Conjunto de bits que contiene el código para un carácter. El byte de 8 bits, llamado octeto, es el más usado actualmente. Las capacidades de almacenamiento se miden en bytes o en sus múltiplos. 1 k (kilobytes) = 1024 bytes. 1 MB (megabyte) = 1024k. 1GB (gigabyte) = 1024 MB.

### **C**

Lenguaje de programación de alto nivel que posee características de los de bajo nivel y esto lo hace adecuado para escribir software de sistemas.

### **Cadena**



## Unidad III. Métodos de búsqueda



Tipo de estructura de datos que es un conjunto de caracteres almacenados consecutivamente. También recibe este nombre un conjunto de programas relacionados lógicamente para realizar una tarea.

### **Campo**

Son los elementos diferenciados de un registro de información. Son los lugares destinados para contener un dato determinado en un fichero o en una base de datos.

### **Carácter**

Dígito, letra o símbolo que puede ser tratado por un ordenador. Es la unidad elemental de información desde el punto de vista externo.

### **Clave**

Campo de datos que identifica a los registros.

### **Codificar**

Es la parte de la programación consistente en escribir en un lenguaje informático el algoritmo que resuelve un problema.

### **Código**

Tabla de traducción entre todos los posibles símbolos de dos alfabetos. El más conocido es el código ascii (american standard code for information interchange).

### **Cola**

Estructura de información consistente en una lista lineal en la que la nueva información se coloca en un extremo y la salida para procesar es por el otro extremo. Ejemplos: la cola de trabajos pendientes de ejecutar o la cola de espera de impresión de resultados.



## Unidad III. Métodos de búsqueda



### **Comando**

Mandato ejecutable por el ordenador. Cada una de las instrucciones que componen el sistema operativo y que el ordenador puede traducir en acciones específicas.

### **Compatibilidad**

Dos ordenadores son compatibles cuando los programas que funcionan en uno de ellos también lo hacen en el otro. Los grandes fabricantes de ordenadores son quienes fijan los estándares de compatibilidad.

### **Compilador**

Programa que traduce un programa escrito en un lenguaje de alto nivel a lenguaje de máquina. Además de traducir se detectan posibles errores de sintaxis.

### **Datos**

La materia prima de que se nutren los programas para producir resultados. Pueden ser de varios tipos: numéricos, alfabéticos, alfanuméricos (cualquier conjunto de símbolos) y lógicos (solo dos valores posibles, verdadero o falso).

### **Debugging**

Depuración. Proceso de eliminar errores de un programa.

### **Diccionario de datos**

Tabla con las propiedades y estructura de los ficheros que forman una base de datos.

### **Digital**

Es una cantidad que sólo puede asumir valores discretos.

### **Dirección**

Etiqueta, nombre o número que identifica la posición de un dato en la memoria o en una unidad de almacenamiento.



## Unidad III. Métodos de búsqueda



### **Directorio**

Tabla de los ficheros que se encuentran en un disco.

### **Directorio raíz**

Es el directorio de nivel más alto en una estructura jerárquica de directorios.

### **Editor**

Programa que permite escribir y corregir ficheros de texto como documentos o programas.

### **Ejecutar**

Realizar una instrucción en código máquina o bien hacer que se realicen las instrucciones de un programa.

### **Ensamblador**

Se conoce con este nombre tanto a un lenguaje de bajo nivel (cercano al lenguaje máquina) como al programa que traduce a lenguaje máquina los programas escritos en lenguaje ensamblador.

### **Estructura de datos**

Conjunto de datos en que sus componentes están relacionados entre sí de una forma particular y sobre los que se pueden realizar ciertas operaciones según sea el tipo de relación que hay entre ellos.

### **Expresión.**

Conjunto de variables, constantes, funciones y símbolos de operación que expresan un cálculo a realizar. Proporcionan un valor que será del tipo (aritmético, alfanumérico o lógico) de las variables y constantes que contiene.

### **Factor de bloqueo**

Cantidad de registros lógicos que hay en cada registro físico de un fichero.



## Unidad III. Métodos de búsqueda



### **Fichero**

Conjunto organizado de información almacenado en un soporte. Está formado por registros que contienen la información de cada elemento del fichero. Los registros a su vez están divididos en campos, siendo cada campo uno de los datos del registro. Por ejemplo en un fichero de clientes, cada cliente sería un registro y tendríamos los campos: nombre, apellidos, dirección, teléfono, etc.

### **Fuente**

Origen. Se dice que un programa está en lenguaje fuente cuando aún no ha sido traducido por ningún compilador.

### **Implementa**

Instalar y hacer funcionar.

### **Indexación**

Operación que consiste en clasificar un fichero o una tabla según el valor de un índice o de una clave.

### **Índice**

Símbolo o número usado para identificar un elemento particular en una tabla.

### **Instrucción**

Se llama instrucción a cada una de las líneas que componen un programa de ordenador.

### **Intérprete**

Programa que traduce instrucciones de lenguaje de alto nivel a lenguaje máquina y las va ejecutando tras su traducción.

### **Iteración**

Cada una de las repeticiones de las acciones contenidas en un bucle de programa.



## Unidad III. Métodos de búsqueda



### **K o kbyte**

Es la unidad de medida de la capacidad de almacenamiento de información, tanto de la memoria principal como de las memorias auxiliares (discos y disquetes). 1 k son 1024 bytes, ( $2^{10}$  elevado a 10 caracteres).

### **Lenguaje de programación**

Es un lenguaje, con su léxico y su sintaxis, que se emplea para que el ordenador ejecute las acciones que se han determinado en un algoritmo y se han codificado en dicho lenguaje. Una primera clasificación de los lenguajes es en lenguajes de bajo nivel (lenguajes máquina y ensambladores) y lenguajes de alto nivel (ada, basic, cobol, fortran, pascal, etc.). Estos últimos, que son independientes de una máquina particular, pueden clasificarse como de propósito especial y general; también como lenguajes de procedimiento y declarativos. Actualmente se evoluciona hacia lenguajes de más alto nivel o lenguajes de especificación.

### **Librería**

Conjunto de programas o módulos almacenados generalmente sobre un dispositivo magnético.

### **Lista**

Conjunto de elementos de datos organizados de modo que se conoce la posición relativa de cada elemento.

### **Longitud**

Aplicado a una cadena de caracteres, a un campo o a un registro, es el número de bytes que ocupa.

### **Loop**

Bucle. Repetición de un cierto número de instrucciones el número de veces que indique un contador o bien mientras se cumpla una condición.



## Unidad III. Métodos de búsqueda



### **Macroinstrucción**

Instrucción en un lenguaje ensamblador que genera varias instrucciones en lenguaje máquina. En paquetes de programas se entiende por macros las instrucciones que desencadenan otro conjunto más amplio de instrucciones.

### **Megabyte**

Unidad de medida de capacidad de almacenamiento de información. Un megabyte es 1024 kbytes y por tanto 1.048.576 bytes.

### **Módulo**

En programación, cada uno de los bloques lógicos en que se puede dividir un programa.

### **Multiprogramación**

Técnica de utilización de un ordenador que permite la ejecución simultánea de varios programas porque les dedica el procesador un pequeño tiempo a cada uno de ellos.

### **Pila**

Un tipo de estructura de datos en el que el último dato en llegar es el primero en salir.

### **Programación**

Técnicas destinadas a la resolución de un problema, usando para ello los lenguajes de programación. Se manejan estructuras lógicas de los tipos: secuencial, condicional e iterativa.

### **Recursividad**

Propiedad de algunos lenguajes de programación de permitir que un programa solicite su propia ejecución en el curso de su desarrollo.

### **Registro**



## Unidad III. Métodos de búsqueda



Conjunto de uno o más campos consecutivos que son tratados por un programa como unidad de información. Son los componentes lógicos de los ficheros.

### **Sort**

Programa para clasificar un conjunto de datos.

### **Subrutina**

Conjunto de instrucciones que efectúan una tarea específica dentro de un programa y al que es posible referirse.

### **Variable**

En programación es una entidad dedicada a contener valores que pueden cambiar durante la ejecución de un programa.

### **Velocidad**

La rapidez en ejecutar instrucciones depende de la frecuencia en que trabaje un reloj interno. Se mide en megahercios.

### **Velocidad de acceso**

La velocidad de acceso a los datos contenidos en un disco duro se mide por el tiempo de recuperación de un dato. Se expresa en milisegundos.



## Unidad III. Métodos de búsqueda



### **MESOGRAFÍA**

#### **Bibliografía básica**

#### **Bibliografía complementaria**

#### **Sitios electrónicos**



## Unidad III. Métodos de búsqueda



**(ANEXO 1)** TEMA 2.pdf



### Unidad III. Métodos de búsqueda



**(ANEXO 2)** TEMA 3.pdf





## Unidad III. Métodos de búsqueda



**(ANEXO 3)** TEMA 4.pdf