



# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

# UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

## ESTRUCTURA DE DATOS

Ing. Mayra Alvarez, MSc.  
[mialvarez2@espe.edu.ec](mailto:mialvarez2@espe.edu.ec)



## Lección 2.3 : Algoritmos de Búsqueda



# Algoritmos de Búsqueda

## Definición

- Un algoritmo de búsqueda es aquel que está diseñado para localizar un elemento concreto dentro de una estructura de datos.
- Consiste en solucionar un problema de **EXISTENCIA** o no de un elemento determinado en un conjunto finito de elementos, es decir, si el elemento en cuestión pertenece o no a dicho conjunto, además de su localización dentro de éste.



Elemento a buscar: 80

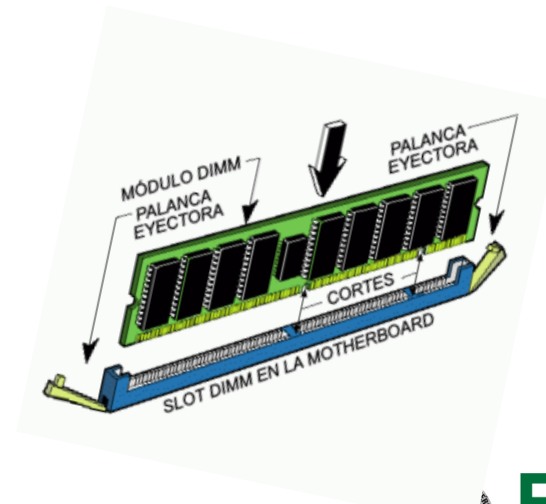
80						
20	40	60	80	100	120	140



**ESPE**  
ESCUELA POLITÉCNICA DEL EJÉRCITO  
CAMINO A LA EXCELENCIA

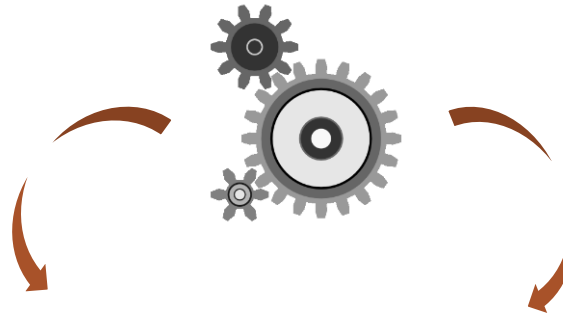
## Definición

- La búsqueda de un elemento dentro de un **ARRAY** es una de las operaciones más importantes en el procesamiento de la información, y permite la recuperación de datos previamente almacenados.
- El tipo de búsqueda se puede clasificar como interna o externa, según el lugar en el que esté almacenada la información (en memoria o en dispositivos externos).



## Importante

- Todos los algoritmos de búsqueda tienen dos finalidades:



Determinar si el elemento buscado se encuentra en el conjunto en el que se busca.

Si el elemento está en el conjunto, hallar la posición en la que se encuentra.

# Algoritmos de Búsqueda

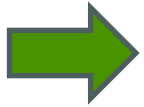
## Tipos

- ✓ Búsqueda Lineal
- ✓ Búsqueda Binaria
- ✓ Búsqueda Hash



## Lección 2.3.1 : Búsqueda Lineal / Secuencial



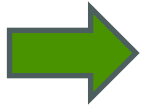


## Búsqueda Lineal / Secuencial

- Consiste en recorrer y examinar cada uno de los elementos del ARRAY hasta encontrar el o los elementos buscados, o hasta que se han revisado todos los elementos del ARRAY.
- Es el método de búsqueda más lento, pero si nuestra información se encuentra completamente desordenada es el único que nos podrá ayudar a encontrar el dato que buscamos.

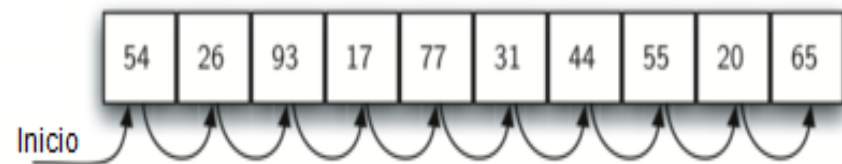
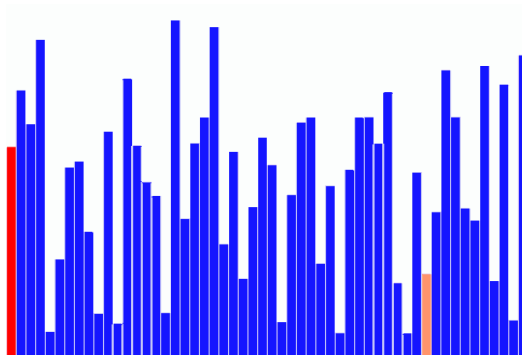






## Rendimiento (búsqueda secuencial / lineal)

- El siguiente gráfico muestran el aumento en el número de transacciones en relación con el número de elementos a ser encontrados.
- Este algoritmo puede usarse en cualquier situación, pero se recomienda usarlo solo en listas que no estén ordenadas.



## Complejidad (búsqueda secuencial / lineal)

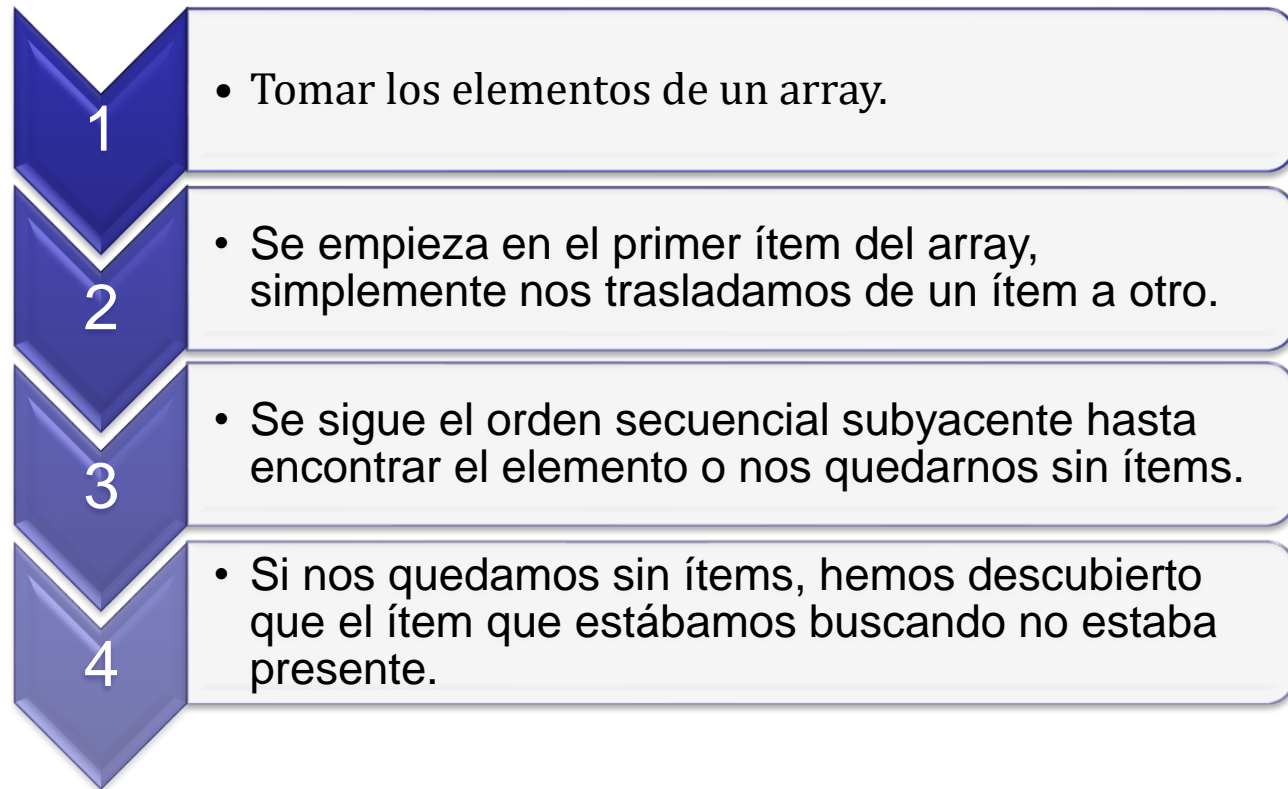
Este es el algoritmo de más simple implementación pero no el más efectivo.

- **MEJOR CASO:** El algoritmo de búsqueda lineal termina tan pronto como encuentra el elemento buscado en el array.
- **PEOR CASO:** Sucede cuando encontramos el elemento en la última posición del array.

Una búsqueda más eficiente puede hacerse sobre un arreglo ordenado. Una de éstas es la **BÚSQUEDA BINARIA**.



## Búsqueda Secuencial

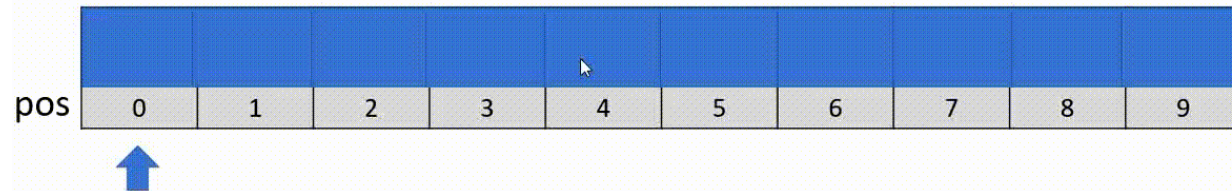


## Búsqueda Secuencial

Linear Search



Buscar el número 78



## Lección 2.3.2 : Búsqueda Binaria



## Definición Búsqueda Binaria

- El algoritmo consiste en reducir paulatinamente el ámbito de búsqueda a la mitad de los elementos, basándose en comparar el elemento a buscar con el elemento que se encuentra en la mitad del intervalo.

### Binary search

1	2	3	9	11	13	17	25	57	90
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]



## Definición Búsqueda Binaria

- Es un método que se basa en la división sucesiva del espacio ocupado por el vector en sucesivas mitades, hasta encontrar el elemento buscado.
- Esta búsqueda utiliza un método de “DIVIDE Y VENCERÁS” para localizar el valor deseado.
- Divide el array por su elemento medio en dos subarrays más pequeños, y comparar el elemento con el del centro.
  - Si el elemento es menor, debe estar en el primer subarray.
  - Si es mayor está en el segundo subarray.



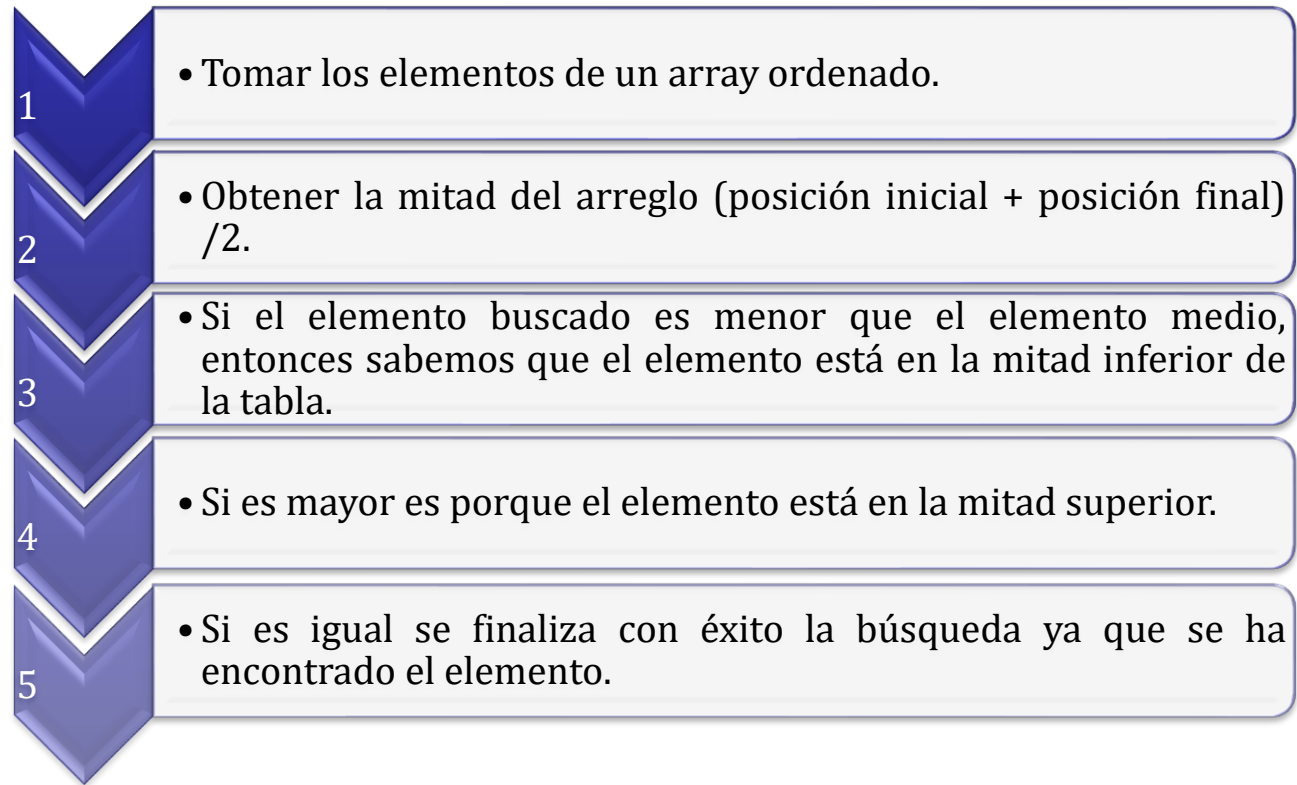
## Complejidad (búsqueda binaria)

- **MEJOR CASO:** La búsqueda binaria podría toparse con el elemento buscado en el primer punto medio, requiriéndose sólo una comparación de elementos.
- **PEOR CASO:** La búsqueda binaria podría toparse con el elemento buscado en la última comparación.

El algoritmo de búsqueda binaria progresivamente va disminuyendo el número de elementos sobre el que realizar la búsqueda a la mitad



## Búsqueda Binaria



## Búsqueda Binaria

En el canal de YouTube de Martino encontramos la descripción de  
Búsqueda Binaria

<https://youtu.be/wAmu0Ly5ook>



## Búsqueda Binaria

Se toma el elemento central y se divide el array en dos. Elemento a **buscar 3**:

1,2,3,4,5,6,7,8,9      {1,2,3,4} - **5** - {6,7,8,9}

Como el elemento buscado (3) es menor que el central (5), debe estar en el primer subarray. Se vuelve a dividir el array en dos:

{1,2,3,4}      {1} - **2** - {3,4}

## Búsqueda Binaria

Como el elemento buscado es mayor que el central, debe estar en el segundo subarray

$\{3,4\}$

Se vuelve a dividir en dos:

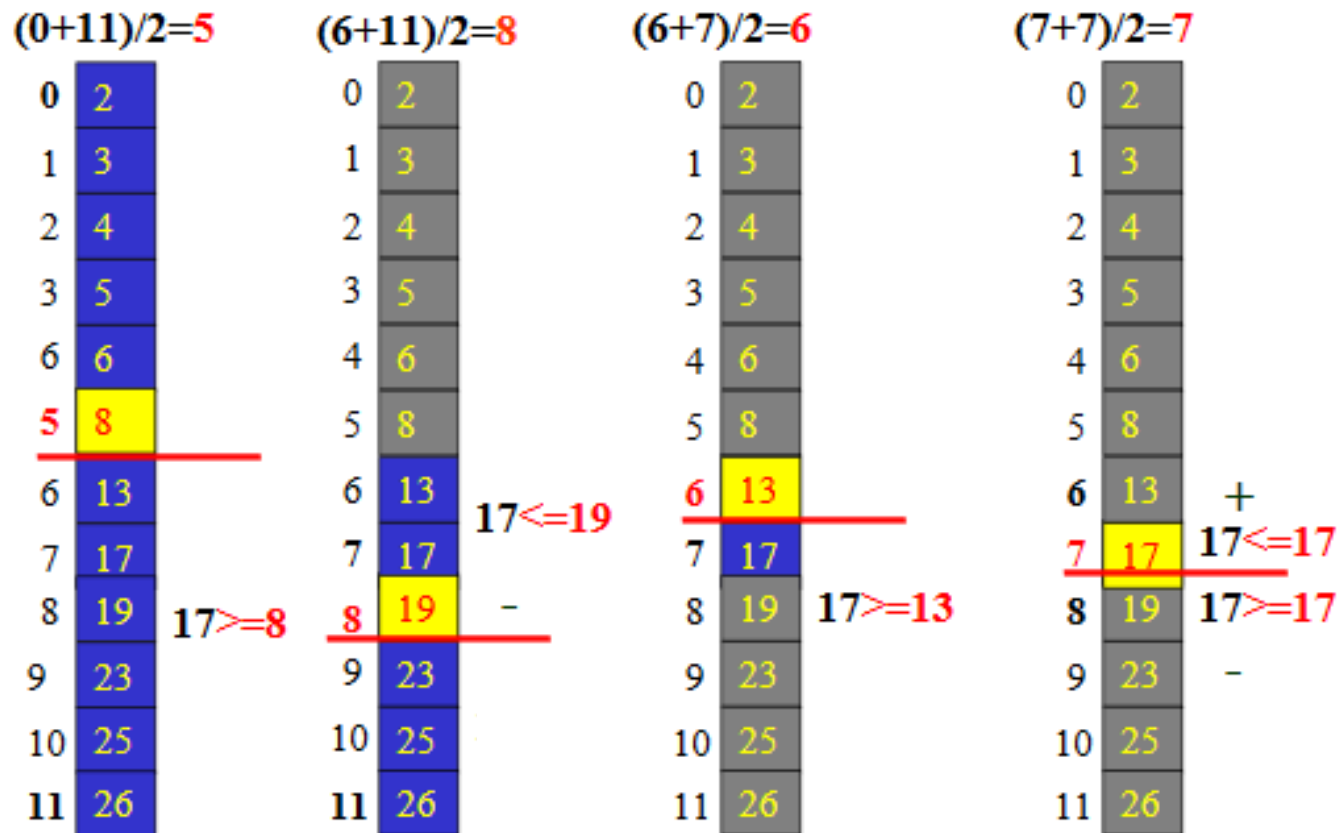
$\{\} - 3 - \{4\}$

Como el elemento buscado coincide con el central, lo hemos encontrado.

# Algoritmos de Búsqueda

## Búsqueda Binaria

Número a buscar **17**



# Algoritmos de Búsqueda

Inicio

```
Entero n, i, num, primero, ultimo, medio
Entero A[n]
Escribir("Ingrese un arreglo ordenado")
Escribir("Cuantos elementos te gustaría ingresar?: ")
Leer(n)
```

// Leer elementos del arreglo

```
Escribir("Ingrese el numero que desea buscar: ")
Leer(num)
```

```
primero <- 0
ultimo <- n - 1
medio <- (primero + ultimo) / 2
```

Mientras primero <= ultimo hacer

    Si A[medio] < num entonces

        primero <- medio + 1

    Sino Si A[medio] == num entonces

        Escribir("Se encontro la posición ", medio + 1)

        Romper el bucle

    Sino

        ultimo <- medio - 1

    Fin Si

    medio <- (primero + ultimo) / 2

Fin Mientras

// Verificar si el número no se encontró

Fin

## Algoritmo de Búsqueda Binaria



**E S P E**  
ESCUELA POLITÉCNICA DEL EJÉRCITO  
CAMINO A LA EXCELENCIA

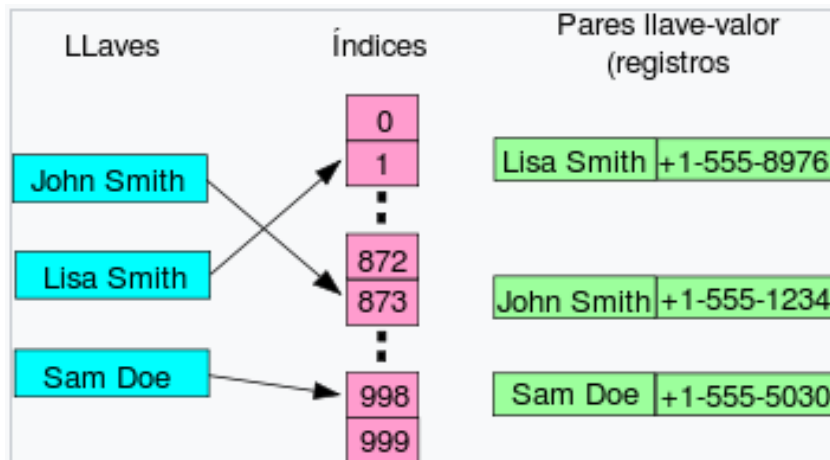
## Lección 2.3.3 : Búsqueda Hash



## Definición

El método de búsqueda hash o por transformación de clave, aumenta la velocidad de búsqueda sin necesidad de que los elementos estén previamente ordenados, comparándolo con los métodos anteriores.

Además tiene la ventaja de que el tiempo de búsqueda es independiente del número de elementos de la estructura que los almacena.

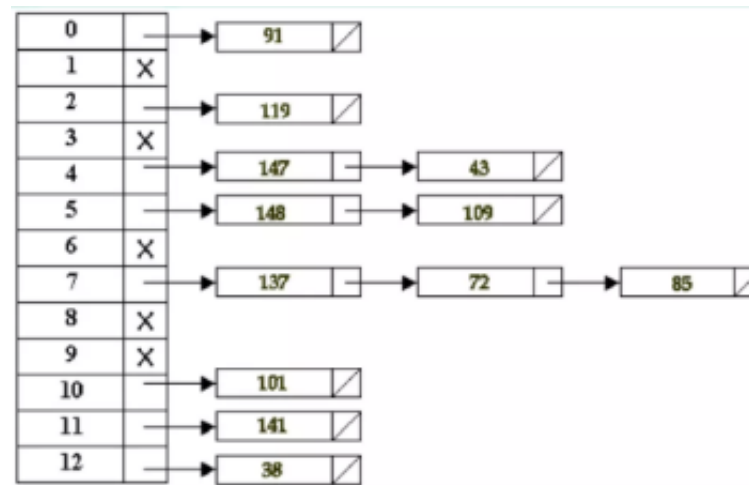




## Definición

El método de búsqueda hash sirve para buscar información contenida en una base de datos dentro de un sistema.

El uso de tablas hash provee un acceso casi directo a los archivos, es decir, una búsqueda puede llegar a requerir solo uno o dos intentos en la memoria o archivo que contiene dicha información.



## Definición

Prácticamente trabaja con una función que transforma la llave o dato clave en una dirección (índice) dentro de la estructura y que en ocasiones puede generar una **colisión**, que se define como una misma dirección para dos o más claves distintas.

Para trabajar con este método de búsqueda debe elegir previamente dos cosas:

- Una función hash que sea fácil de calcular y que distribuya uniformemente las direcciones.
- Un método para resolver colisiones, generando posiciones alternativas.

## Definición

Para encontrar la función hash no existe una regla que permita determinar cuál será la función más apropiada para generar un conjunto de claves que aseguren la máxima uniformidad en la distribución de las mismas.

Algunas de las funciones hash más utilizadas son las siguientes:

- Función módulo (por división).
- Función cuadrada.
- Función plegamiento.
- Función truncamiento.

## Tablas Hash

Índice	Valor
0	21
1	14
2	56
3	10
4	100
5	33
6	20
7	12

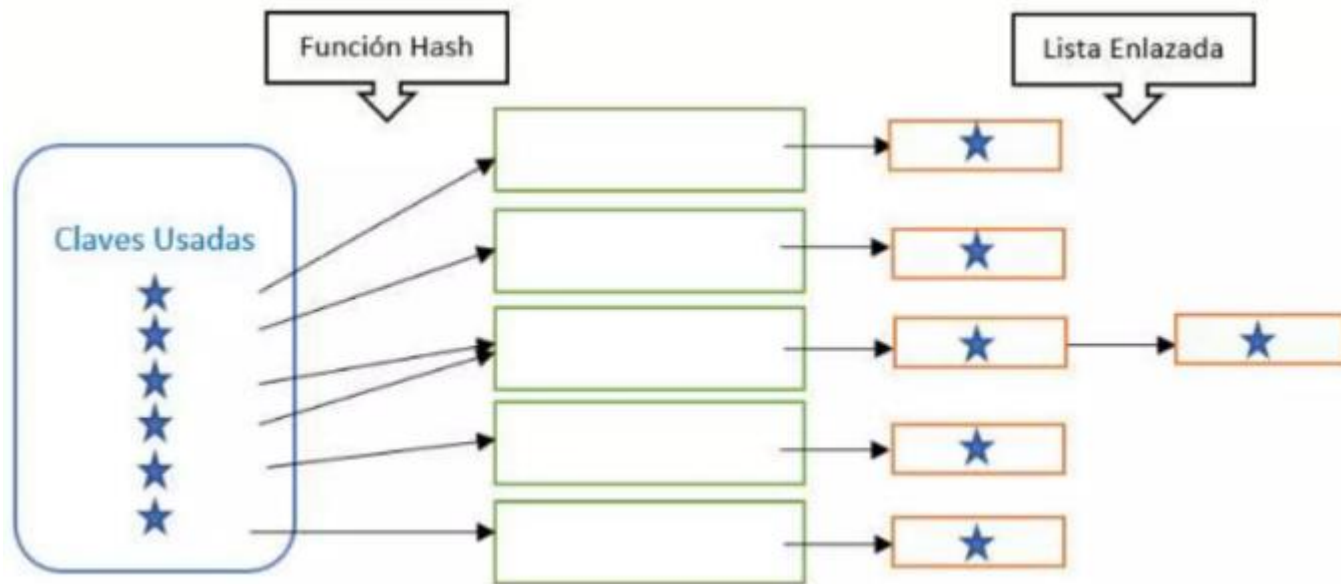
20, 33, 21, 10, 12, 14, 56, 100

Tamaño = 8  
Index = 7

Al buscar el elemento ya nos basamos en la clave.

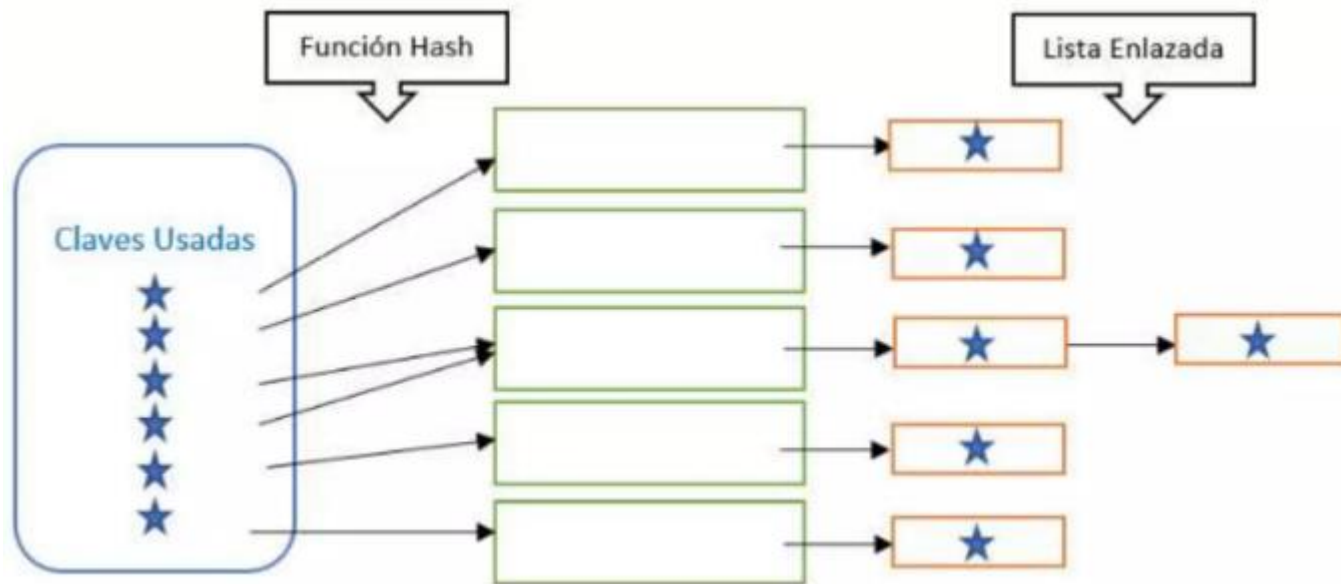
## Hash Abierto

Las claves se almacenan en listas enlazadas unidas a celdas de una tabla hash.



## Hash Cerrado

Todas las claves se almacenan la propia tabla de hash sin el uso de listas enlazadas.



Preguntas ?



## Proyecto

Para la búsqueda por hash realizar las siguientes funciones:

- Crear tabla hash.
- Insertar elementos.
- Imprimir tabla hash.
- Buscar elementos en la tabla.
- Destruir la tabla hash.

**Nota: Implementar TDA**





## Proyecto

**Herrera – Jami**

**Gualotuña – Cedeño**

**Obando – Peñaranda**

**Alvarez – Colina**

**Nicoll – Alomia**

**Arico – Sacancela**

**Mena – Carrillo**

**Pillajo – Chancusig**

**Gordon - Luzuariaga**



## Extra

Actividad opcional **individual**.

- Medos de ordenamiento externo.
- Búsqueda hash abierto y cerrado con TDAs.



## Realizar

Realizar un menú con las opciones de búsqueda simple y búsqueda binaria.

- Realice el proceso de búsqueda simple.
- Traduzca el algoritmo de búsqueda binaria.