



El futuro digital  
es de todos

MinTIC

Ordenamiento y Búsqueda

»  
**Misión TIC 2022**

xxx



Misión  
TIC 2022



El futuro digital  
es de todos

MinTIC



# Fundamentos de Programación

x  
x  
x

## Ordenamiento y Búsqueda - General



Ordenamiento de información



Búsqueda de información





# Fundamentos de Programación



## Ordenamiento de Información

El **ordenamiento** organiza los **datos** en orden alfabético o numérico ascendente o descendente. Por ejemplo, puede ordenar una columna que enumere los valores de las ventas de un producto en orden descendente para ordenar las ventas del producto de la más alta a la más baja.



## » Ordenamiento Básico

### ● Ordenamiento iterativo – Método Burbuja

x  
x  
x



El ordenamiento burbuja funciona revisando cada elemento de la lista que va a ser ordenada con el siguiente, intercambiándolos de posición si están en el orden equivocado. Es necesario revisar varias veces toda la lista hasta que no se necesiten más intercambios, lo cual significa que la lista está ordenada.







## » Ordenamiento Básico



### Ordenamiento iterativo – Método Burbuja

x  
x  
x

Ubicar la información de una lista o vector de menor a mayor (Ascendente) o de mayor a menor (Descendente): El método de Burbuja tradicional realiza el ordenamiento a través de dos ciclos anidados y un condicional.

a[0] a[1] a[2] a[3]

8	4	6	2
---	---	---	---

Se realiza *intercambio*

4	8	6	2
---	---	---	---



4	8	6	2
---	---	---	---

No se realiza *intercambio*

4	8	6	2
---	---	---	---

4	8	6	2
---	---	---	---

Se realiza *intercambio*

2	8	6	4
---	---	---	---



*Lista inicial*

*Lista resultante*



# Ordenamiento Método Burbuja



Dada una **lista vector** de **N elementos**, se pide **ordenarla de menor a mayor (Ascendente)** mediante el método de Burbuja tradicional.

Vector de entrada:

Vector de salida:



Lista de números
5
3
4
2
1

Lista de números
1
2
3
4
5



# Ordenamiento

## Método burbuja



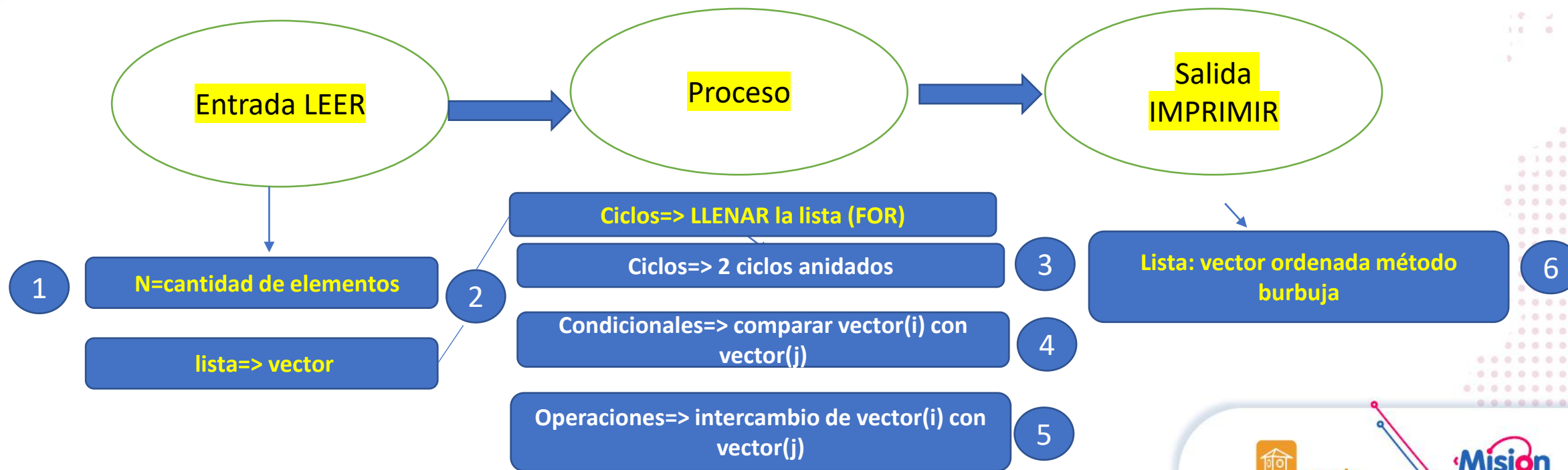
Iteraciones				
Posición				
0	5	1	1	1
1	3	3	2	2
2	4	4	4	3
3	2	2	3	4
4	1	5	5	5
N=5				
Vector: num				
Burbuja tradicional: Se debe comparar cada elemento con los siguientes elementos				
Ciclo i	inicia en el primer elemento hasta penúltimo			
Ciclo j	inicia en i+1 (siguiente de i) hasta último			
Ciclos anidados				
0	1	2	3	
1,2,3,4	2,3,4	3, 4	4	
Condicional=> Comparar la num[i] con el num[j]				



# Ordenamiento



Análisis → Ejercicio método burbuja







# Ordenamiento con Funciones

## Ordenamiento Burbuja



Parámetros de entrada

vector



**MODULO (FUNCION)**  
**ordenamiento\_burbuja**

Ciclos anidados  
Condicional  
Intercambio



Parámetros de salida

Vector ordenado

**FUNCIÓN retorna o regresa un solo valor**

3

4

5



# Ordenamiento

Construcción → Ordenamiento burbuja



```
# programa para el método de ordenamiento Burbuja
# Autor: Sergio Medina
# Fecha: 03/06/2021

#Funciones
def valida_entero(etiqueta):
    while True:
        try:
            dato = int(input(etiqueta))
            break
        except ValueError:
            print(etiqueta, " debe ser entero Intenta de nuevo...")
    return dato

def ordenamiento_burbuja(numeros):
    for i in range (0,N-1):
        for j in range (i+1,N):
            if numeros[i]>numeros[j]:
                t=numeros[i]
                numeros[i]=numeros[j]
                numeros[j]=t
    return numeros
```



# Ordenamiento

Construcción → Ordenamiento burbuja



```
#Programa principal
N=valida_entero("Ingrese cantidad de elementos: ")
numeros=[]
for i in range(N):
    num=valida_entero("Número: ")
    numeros.append(num)
print( "Lista Numeros: ",numeros)
#Llamado a la función
numeros=ordenamiento_burbuja(numeros)
print ("Lista ordenada: ",numeros)
```



## » Búsqueda de información



Las **búsquedas de información** consisten en un proceso iterativo en el **que** se realiza una **búsqueda**, se analizan los resultados y se va modificando la estrategia de **búsqueda** hasta identificar los términos y las fuentes de **información que** nos proporcionan los resultados más pertinentes.

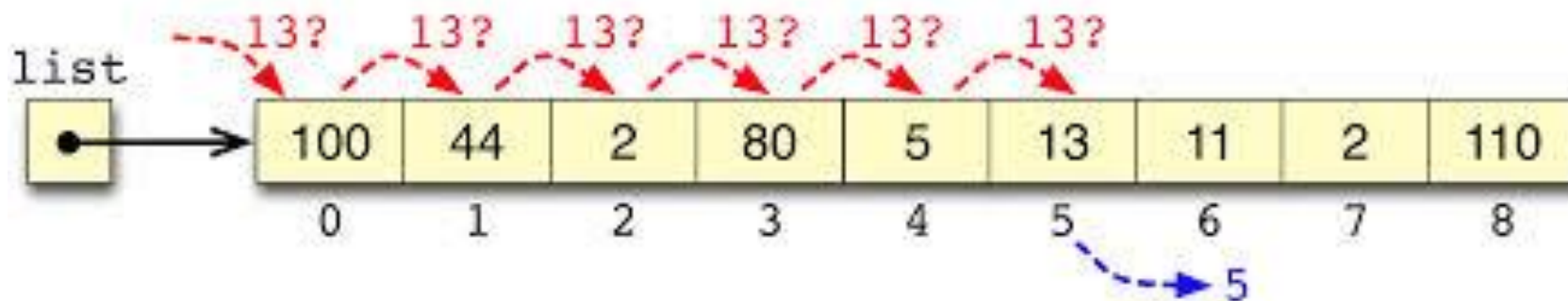




xxx

## Búsqueda de información – Búsqueda secuencial o lineal

El Método de búsqueda secuencial o lineal, tiene como objetivo buscar una información suministrada dentro de una lista. El proceso de búsqueda secuencial o lineal consiste en comparar el elemento a buscar con cada uno de los elementos de la lista hasta ser encontrado o finalizar la lista.





## Búsqueda secuencial de información



Dada una **lista vector de N elementos** y una **información a buscar** en la lista, se pide realizar el proceso de búsqueda lineal o secuencial e indicar en cual posición de la lista se encuentra la información.



Entrada:

Lista de números
2
4
1
5
3

Salida:

Posición donde se encontró la información
3

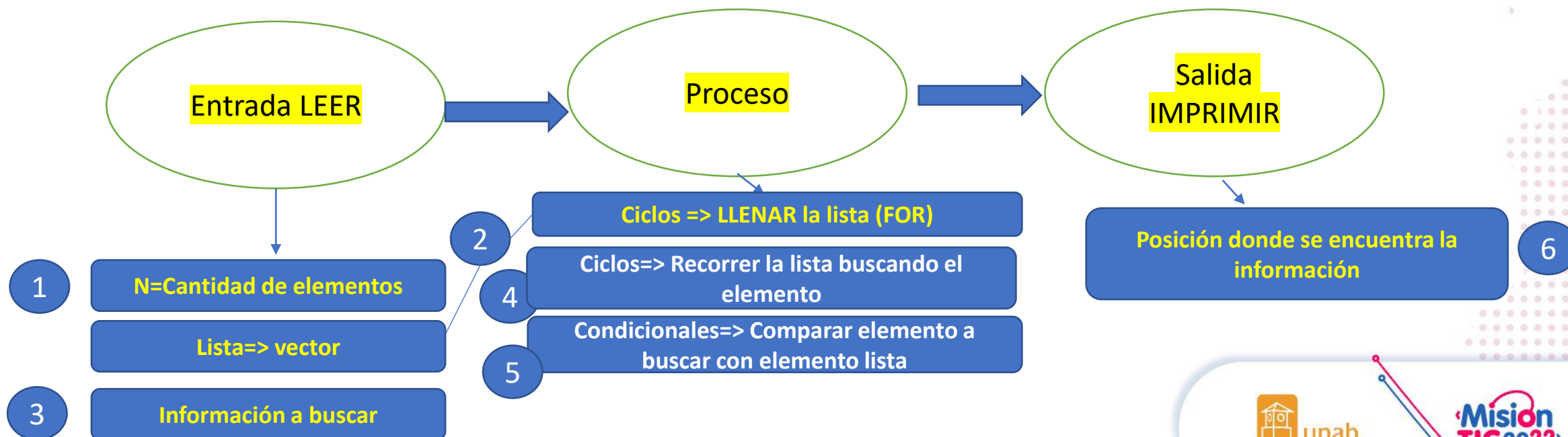
Información a buscar: 5



# Búsqueda secuencial de información



Análisis → Ejercicio método búsqueda secuencial





# Búsqueda con Funciones

## Ejercicios



Ordenamiento Búsqueda secuencial con Modularidad

Parámetros de entrada

Vector  
Info\_buscar



**MODULO (FUNCION)**  
**Búsqueda\_lineal**

Ciclo For para recorrer la lista  
Condicional para comparar



Parámetros de salida

Posición donde  
encuentra elemento

**FUNCIÓN retorna o regresa un solo valor**

4

5





# Búsqueda secuencial de información

Construcción → Programa



```
#Programa para la búsqueda lineal o secuencial
#Autor: Sergio Medina
#Fecha: 04/06/2021

#Funciones
def valida_entero(etiqueta):
    while True:
        try:
            dato=int(input(etiqueta))
            break
        except ValueError:
            print(etiqueta," debe ser dato ENTERO")
    return dato

def busqueda_lineal(numeros,buscar):
    for i in range(N):
        if numeros[i]==buscar:
            return i
    return -1
```



# Búsqueda secuencial de información

Construcción → Programa



```
#Programa principal
numeros=[]
N=valida_entero("Ingrese cantidad de elementos: ")
for i in range(N):
    num=valida_entero("Número: ")
    numeros.append(num)
print("Lista: ",numeros)

buscar=valida_entero("Información a buscar: ")
posicion=busqueda_lineal(numeros,buscar)
if posicion==-1:
    print("Información no encontrada")
else:
    print("Información encontrada en posición: ",posicion)
```

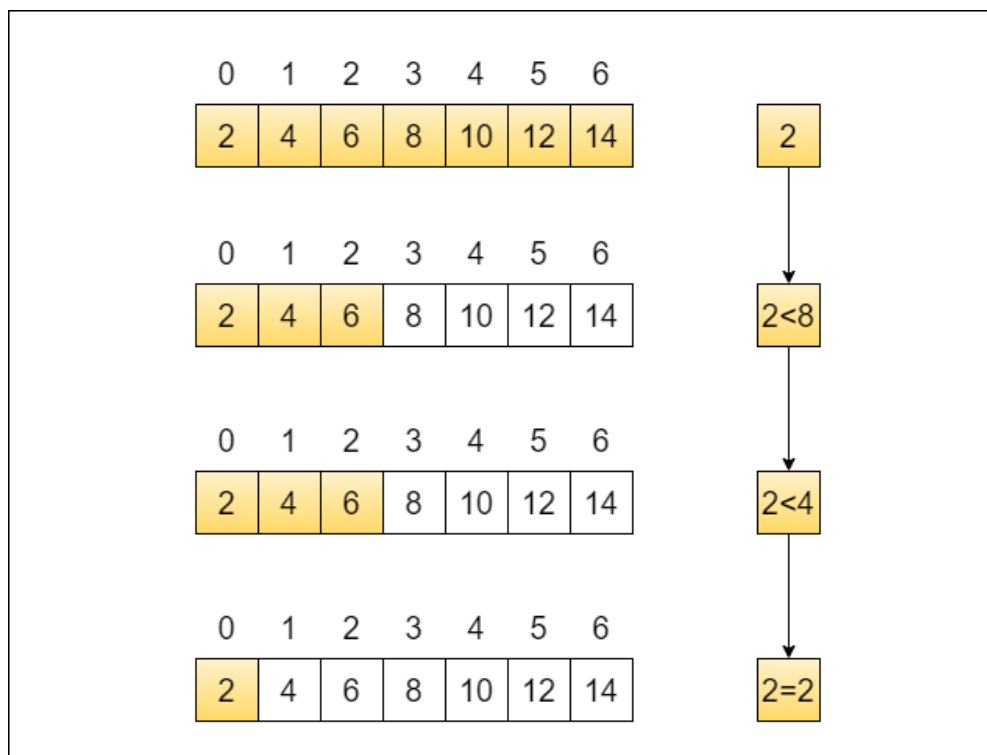


## Búsqueda de información



### Búsqueda de información – Búsqueda binaria (No recursivo)

La **búsqueda binaria** es un algoritmo eficiente para **encontrar un elemento** en una **lista ordenada** de elementos. Funciona al **dividir repetidamente a la mitad la porción** de la lista que podría contener al elemento, hasta reducir las ubicaciones posibles a solo una.





## » Búsqueda binaria de información



×  
×  
× La búsqueda binaria:

1. Debo tener izq y der de la lista (Posición primera y última de la lista)

Primera vez:  $izq=0$   $der=len(lista)-1$

2. Me ubico en la mitad de lista=>  $med=(izq+der)//2$

3. Comparar  $lista[med]$  con información a buscar:

Si es igual, lo encontré, retorne la posición

Si es menor, voy a la lista izquierda ( $der=med-1$ )

Si es mayor, voy a la lista derecha ( $izq=med+1$ )

4. Repetir el proceso...mientras  $izq \leq der$





## Búsqueda binaria de información



Dada una **lista vector** elementos y una **información a buscar** en la lista, se pide realizar el proceso de búsqueda binaria e indicar en cual **posición de la lista se encuentra la información**. (Vector está ordenado)

Entrada:

Lista de números
1
2
3
4
5
6
7
8

Información  
a buscar: 3

Salida:

Posición donde se  
encontró la  
información

2



# Búsqueda binaria de información



Análisis → Ejercicio método búsqueda binaria

Proceso

Entrada LEER

Salida  
IMPRIMIR

1

Lista vector

2

info\_buscar

3

Ciclos => WHILE para a búsqueda , mientras  
izq<=der

4

Condicionales=> si info\_buscar es igual a  
lista[med]  
Si info\_buscar es menor lista[med] voy a la  
izquierda  
Si info\_buscar es mayor lista[med] voy a la  
derecha

5

Operaciones=>  
 $med = (iz + der) // 2$   
Der=med-1  
Izq=med+1

6

posición de la lista donde se  
encuentra la información



El futuro digital  
es de todos

MinTIC

# Búsqueda binaria de información



Ordenamiento Búsqueda binaria

Parámetros de entrada

Vector  
Info\_buscar



**MODULO (FUNCION)**  
Búsqueda\_binaria

Ciclo While  
Condicionales  
operaciones



Parámetros de salida

Posición

**FUNCIÓN retorna o regresa un solo valor**

3

4

5



# Búsqueda binaria de información



```
#Programa para la busqueda binaria
def valida_entero(etiqueta):
    while True:
        try:
            dato=int(input(etiqueta))
            break
        except ValueError:
            print(etiqueta," debe ser dato ENTERO")
    return dato

def busqueda_binaria(vector,elemento):
    izq=0
    der=len(vector)-1

    while izq<=der:
        med=(izq+der)//2

        if vector[med]==elemento:
            return med
        elif vector[med]>elemento:
            der=med-1
        else:
            izq=med+1
    return -1
```

Construcción → Programa





# Búsqueda binaria de información



Construcción → Programa

```
vector=[1,2,3,4,5,6,7,8,9,10]
elemento=valida_entero("Elemento: ")
if busqueda_binaria(vector,elemento)==-1:
    print("Elemento no encontrado en la lista")
else:
    print("Posición elemento: ",busqueda_binaria(vector,elemento))
```



El futuro digital  
es de todos

MinTIC

Ordenamiento y Búsqueda

»  
**Misión TIC 2022**

xxx



Misión  
TIC 2022