



**UNRaf**

UNIVERSIDAD  
NACIONAL DE  
RAFAELA

Bv. J.A. Roca 989 / CP: 2300  
**Rafaela** - Santa Fe - Argentina

T: +54 (03492) 501155

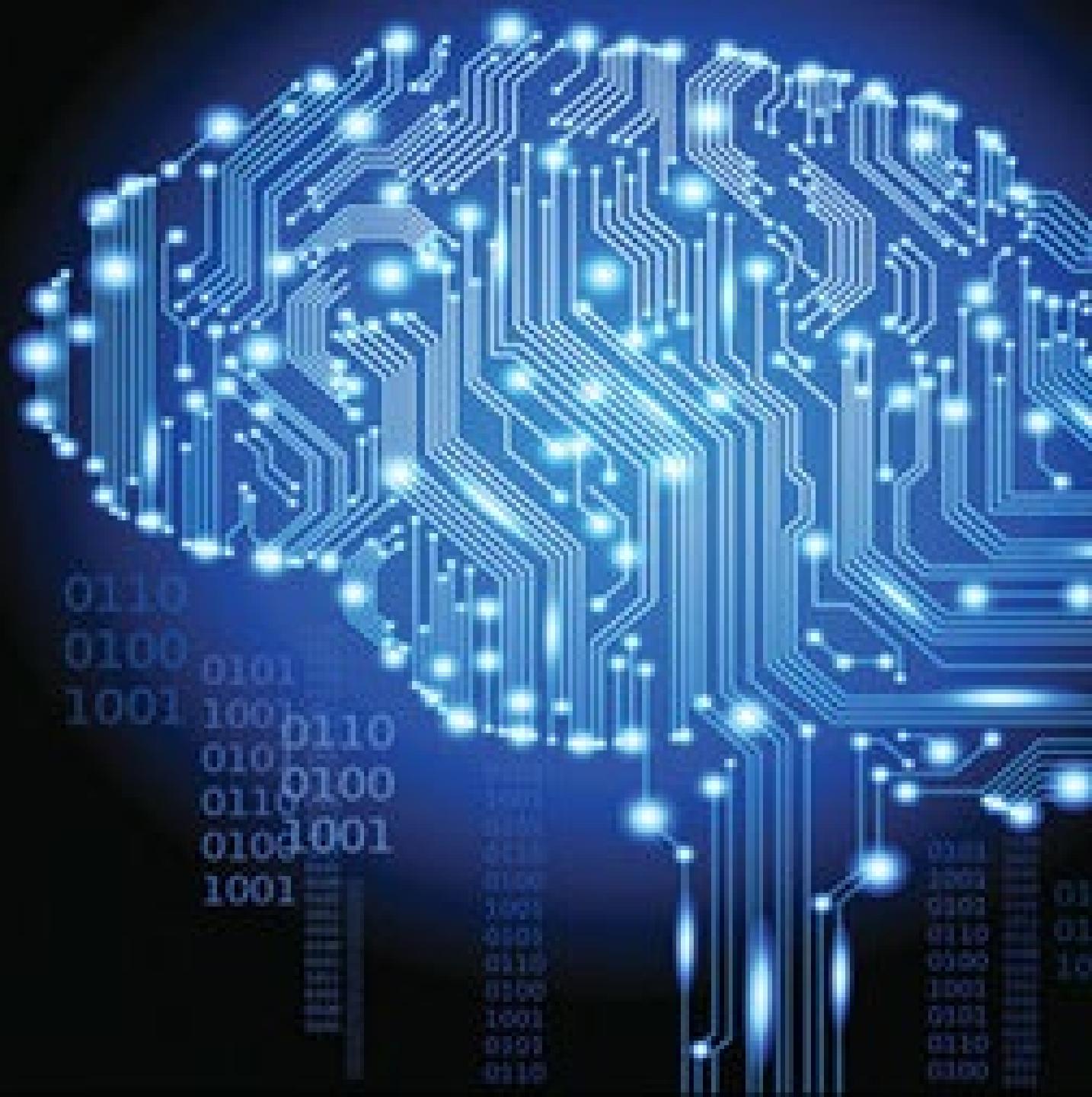
[info@unraf.edu.ar](mailto:info@unraf.edu.ar)  
[www.unraf.edu.ar](http://www.unraf.edu.ar)



Ingeniería en  
Computación

**ALGORITMOS Y  
ESTRUCTURAS  
DE DATOS**

# ALGORITMOS Y ESTRUCTURAS DE DATOS



i ?

# AGENDA

- Repaso
- Ordenamiento y búsqueda
- Métodos de Ordenamiento
  - Burbuja
  - Inserción
  - Selección
  - Mezcla
  - Quicksort
- Métodos de Búsqueda
  - Lineal
  - Binaria
- Ejemplos



# MÉTODOS

**Operaciones importantes en Computación:**  
**Ordenación** (en inglés sorting) de los  
elementos de un conjunto.

**Búsqueda** (en inglés searching) de un  
elemento dentro de un conjunto.

Las computadoras invierten la mitad de su  
tiempo realizando estas operaciones.



# ORDENAMIENTO

Proceso de **clasificar** un conjunto de objetos en un **orden específico**.

El **propósito** de la ordenación es **facilitar la búsqueda** de elementos en el conjunto ordenado.



# ORDENAMIENTO

Dados los elementos:  $a_1, a_2, \dots, a_n$

Ordenar consiste en permutar esos elementos en un orden:  $a_{k_1}, a_{k_2}, \dots, a_{k_n}$

tal que dada una función de ordenamiento f:  $f(a_{k_1}) \leq f(a_{k_2}) \leq \dots \leq f(a_{k_n})$

# ALGORITMOS DE ORDENAMIENTO

Burbuja

Inserción

Selección

Mezcla

Quicksort



# BURBUJA

```
def burbuja(lista, n):  
    i = 0  
    while(i < n-1):  
        j = n-1  
        while(j>i):  
            if(lista[j] < lista[j - 1]):  
                lista[j - 1], lista[j] = lista[j], lista[j - 1]  
            j-=1  
        i+=1
```

6 5 3 1 8 7 2 4



# INSERCIÓN

```
def insercion(lista, n):
    p = 1
    while(p < n):
        temp = lista[p]
        j=p
        while(j > 0 and temp < lista[j - 1]):
            lista[j] = lista[j - 1]
            j-=1
        lista[j] = temp
    p+=1
```

6 5 3 1 8 7 2 4



# SELECCIÓN

```
def seleccion(lista, n):
    i = 0
    while(i<n):
        j=i+1
        min = i
        while(j<n):
            if(lista[j] < lista[min]):
                min = j
            j+=1
        lista[i], lista[min] = lista[min], lista[i]
        i+=1
```

8  
5  
2  
6  
9  
3  
1  
4  
0  
7

# MEZCLA

```
def mezcla(obj):
    if len(obj) > 1:
        mid = len(obj)/2
        left = obj[:mid]
        right = obj[mid:]
        mezcla(left)
        mezcla(right)
        i = 0
        j = 0
        k = 0
        while i < len(left) and j < len(right):
            if left[i] < right[j]:
                obj[k] = left[i]
                i += 1
            else:
                obj[k] = right[j]
                j += 1
            k += 1
        while i < len(left):
            obj[k] = left[i]
            i += 1
            k += 1
        while j < len(right):
            obj[k] = right[j]
            j += 1
            k += 1
```

6 5 3 1 8 7 2 4

# QUICKSORT

```
def partition(lista, izq, der, pivot):
    while(izq <= der):
        while(lista[izq] < pivot):
            izq+=1
        while(lista[der] > pivot):
            der-=1
        if(izq <= der):
            lista[izq], lista[der] = lista[der], lista[izq] # Intercambio
            izq+=1
            der-=1
    return izq

def quick_sort(lista, izq, der):
    if(izq >= der):
        return
    pivot = lista[ (izq + der) / 2 ]
    index = partition(lista, izq, der, pivot)
    quick_sort(lista, izq, index-1)
    quick_sort(lista, index, der)
```

6 5 3 1 8 7 2 4

# BÚSQUEDA

Se refiere a localizar un dato dentro de un conjunto.

El dato a buscar puede o no estar entre los elementos del conjunto.

Formalmente: Dado un conjunto de  $n$  elementos distintos y un dato  $K$  llamado argumento, determinar:

Si  $K$  pertenece al conjunto y en este caso, indicar cual es su posición.

Si  $K$  no pertenece al conjunto.



Encontrar un argumento  $K$ , dentro de un conjunto de  $n$  elementos se hace **comparando  $K$  con cada elemento del conjunto hasta que este se encuentre, o bien se haya recorrido todo el conjunto.**



# BÚSQUEDA SECUENCIAL

```
def secuencial(lista,item):  
    pos = 0  
    found = False  
    while(pos<len(lista) and not found):  
        if lista[pos] == item:  
            found = True  
        pos = pos +1  
    return found  
  
print(secuencial([1,2,3,5,8],6))  
print(secuencial([1,2,3,5,8],5))
```



# BÚSQUEDA BINARIA

Los elementos del arreglo han sido **ordenados previamente**, por cualquiera de los algoritmos antes vistos.

El método de búsqueda binaria consiste en **localizar, aproximadamente, la posición media del arreglo** y examinar el valor allí encontrado.



# BÚSQUEDA BINARIA

Si este valor es **mayor que el buscado**, entonces, se busca el argumento en la **primera mitad del arreglo**, repitiéndose el proceso en la **mitad correspondiente** del mismo, hasta que, se encuentre el elemento deseado.



# BÚSQUEDA BINARIA

Si por el contrario, el **valor es menor que el buscado**, se prosigue con la segunda mitad del **arreglo**, localizándose el **elemento situado en el centro de ella** y, continuando con el **procedimiento**, hasta haber **encontrado el elemento buscado**, o bien, el **intervalo de búsqueda haya quedado vacío**.



# BÚSQUEDA BINARIA

Después de comparar K con a(i) existen tres posibilidades:

1.  $K < a(i)$  Los elementos  $a(i), a(i+1), \dots, a(n)$  quedan eliminados de la búsqueda.
2.  $K = a(i)$  La búsqueda ha terminado.
3.  $K > a(i)$  Los elementos  $a(1), a(2), \dots, a(i)$  quedan eliminados de la búsqueda.



# BÚSQUEDA BINARIA

```
def binaria(lista, item):
    first = 0
    last = len(lista)-1
    found = False
    while( first<=last and not found):
        mid = int((first + last)/2)
        if lista[mid] == item :
            found = True
        else:
            if item < lista[mid]:
                last = mid - 1
            else:
                first = mid + 1
    return found

print(binaria([1,2,3,5,8],6))
print(binaria([1,2,3,5,8],5))
```

i ?



# UNRaf

UNIVERSIDAD  
NACIONAL DE  
RAFAELA

Bv. J.A. Roca 989 / CP: 2300  
**Rafaela** - Santa Fe - Argentina

T: +54 (03492) 501155

[info@unraf.edu.ar](mailto:info@unraf.edu.ar)  
[www.unraf.edu.ar](http://www.unraf.edu.ar)