

Arrays Paralelos.

***Programación I – Laboratorio I.
Tecnicatura Superior en Programación.
UTN-FRA***

Autores: *Pablo Gil
Hector Farina*

Revisores: *Ing. Ernesto Gigliotti
Lic. Mauricio Dávila*

Versión : 1



Esta obra está bajo una [Licencia Creative Commons Atribución-CompartirIgual 4.0 Internacional](http://creativecommons.org/licenses/by-sa/4.0/).

Índice de contenido

1Arrays Paralelos.....	3
1.1Ordenamiento por un criterio.....	3
1.2Ordenamiento por dos criterios.....	4

1 Arrays Paralelos

Este concepto se da cuando hay una relación entre los componentes de igual subíndice (misma posición) de un vector y otro. Por ejemplo si tenemos dos vectores de 5 elementos cada uno:

```
char nombre[][5] = {"Pedro", "Carlos", "Juan", "Ernesto", "Mario"};
char apellido[][5] = {"Gomez", "Fernandez", "Paz", "Perez", "Lopez"};
```

Índice	Nombre
0	Pedro
1	Carlos
2	Juan
3	Ernesto
4	Mario

Índice	Apellido
0	Gomez
1	Fernandez
2	Paz
3	Perez
4	Lopez

En uno se almacenan los nombres de personas en el otro los apellidos de dichas personas, decimos que el vector nombres es paralelo al vector de apellidos si en la componente 0 de cada vector se almacena información relacionada con la misma persona.

1.1 Ordenamiento por un criterio

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define ELEMENTOS 5

int main()
{
    int i, j;
    char nombre[ELEMENTOS][50] = {"Pedro", "Carlos", "Juan", "Ernesto", "Mario"};
    char apellido[ELEMENTOS][50] = {"Gomez", "Fernandez", "Paz", "Perez", "Lopez"};
    char auxiliar[50];

    for(i=0; i< ELEMENTOS-1; i++)
    {
        for(j=i+1; j<ELEMENTOS; j++)
        {
            if(strcmp(nombre[i], nombre[j])>1)
            {
                strcpy(auxiliar, nombre[i]);
                strcpy(nombre[i], nombre[j]);
                strcpy(nombre[j], auxiliar);

                strcpy(auxiliar, apellido[i]);
                strcpy(apellido[i], apellido[j]);
                strcpy(apellido[j], auxiliar);
            }
        }
    }
}
```

```
printf("\n\nLista ordenada por nombres");
for(i=0;i<ELEMENTOS;i++)
{
    printf("\nNombre: %s - Apellido: %s",nombre[i],apellido[i]);
}

return 0;
}
```

1.2 Ordenamiento por dos criterios

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX 5

int main(void){
    int legajo[MAX], legajoAux, i, j;
    float salario[MAX], salarioAux;
    char nombre[MAX][31], nombreAux[31];

    for(i=0; i<MAX; i++)
    {
        system("cls");

        printf("Ingrese el legajo del empleado: ");
        scanf("%d", &legajo[i]);

        printf("Ingrese el nombre del empleado: ");
        fflush(stdin);
        gets(nombre[i]);

        printf("Ingrese el salario del empleado: ");
        scanf("%f", &salario[i]);
    }

    for(i=0; i<MAX-1; i++)
    {
        for(j=i+1; j<MAX; j++)
        {
            if(salario[i] < salario[j])
            {
                strcpy(nombreAux, nombre[i]);
                strcpy(nombre[i], nombre[j]);
                strcpy(nombre[j], nombreAux);

                legajoAux = legajo[i];
                legajo[i] = legajo[j];
                legajo[j] = legajoAux;

                salarioAux = salario[i];
                salario[i] = salario[j];
                salario[j] = salarioAux;
            }
        }
    }
}
```

```
if(salario[i] == salario[j]){
    if(strcmp(nombre[i], nombre[j]) > 0){
        strcpy(nombreAux, nombre[i]);
        strcpy(nombre[i], nombre[j]);
        strcpy(nombre[j], nombreAux);

        legajoAux = legajo[i];
        legajo[i] = legajo[j];
        legajo[j] = legajoAux;

        /*salarioAux = salario[i];
        salario[i] = salario[j];
        salario[j] = salarioAux;*/
    }
}

system("cls");
printf("\nLegajo\tNombre\t\t\tSalario ");
for(i=0; i<MAX; i++)
{
    printf("\n%d\t%s\t\t\t%.2f", legajo[i], nombre[i], salario[i]);
}

return 0;
}
```