



UNIVERSIDAD  
TECNOLÓGICA  
DE PANAMÁ

Facultad de Ingeniería de Sistemas Computacionales

# Capítulo IV

## Punteros

Kexy Rodríguez

# Contenido

- Punteros y Arreglos
- Arreglos dinámicos utilizando punteros

# Punteros y Arreglos Unidimensionales

- Hay que tomar en cuenta que el nombre de un array, es en realidad un puntero, contiene la dirección en memoria de comienzo de la secuencia de elementos que forma el arreglo.
- Por tanto si **x** es un array unidimensional, entonces la dirección de su primer elemento se puede escribir como **&x[0]**, o simplemente **x**. De igual forma, la dirección del segundo elemento se puede escribir como **&x[1]**, o como **(x+1)**.

(Janitza de Justiniani, 2012).

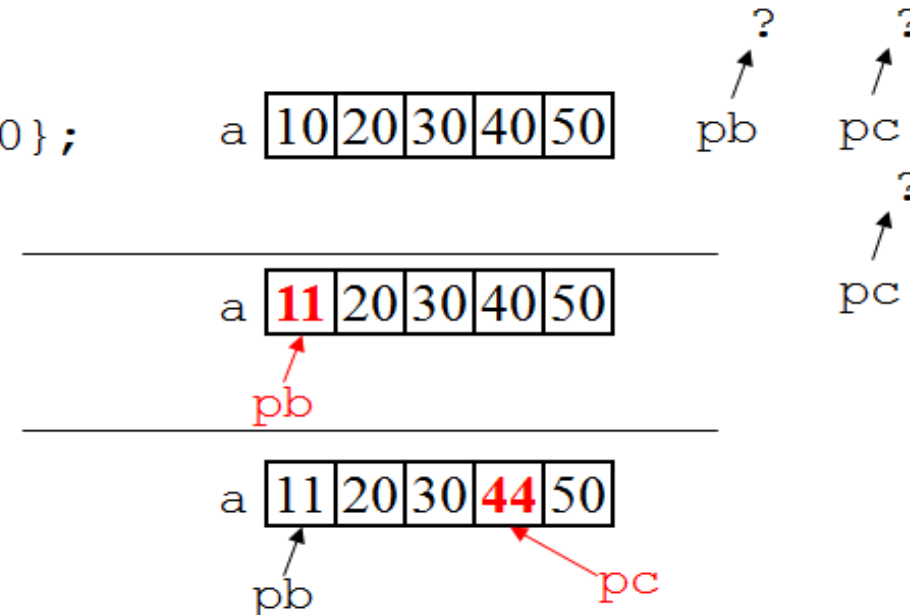
# Ejemplo

```
int main() {  
    //Bloque de declarativa  
    int x[5] = {2,3,4,5,8};  
    //Bloque de instrucciones  
    printf("\n%d",x); // dirección de memoria del primer elemento  
    printf("\n%d",&x[0]); // dirección de memoria del primer elemento  
    printf("\n%d",&x[1]); // dirección de memoria del segundo elemento  
    printf("\n%d",(x+1)); // dirección de memoria del segundo elemento  
    printf("\n%d",*(x+2)); // valor del tercer elemento: 4  
    printf("\n%d",x[2]); // valor del tercer elemento: 4  
    return 0;  
}
```

# Ejemplo

El identificador de una variable array tiene el valor de la dirección de inicio del mismo. Por lo tanto, su valor puede usarse como un puntero.

```
int *pb, *pc;  
int a[5]={10,20,30,40,50};  
pb=a;  
*pb=11;  
pc=&a[3];  
*pc=44;
```





# Arreglos dinámicos

Anteriormente hemos utilizado los arreglos estáticos. Sin embargo, dependiendo del problema se requiere utilizar arreglos dinámicos. Para esto se puede utilizar direcciones de memorias por medio de punteros.

Para crear arreglos dinámicos hay que conocer dos funciones:

“La función **malloc**, abreviación de memory allocation (asignación de memoria), tiene como propósito reservar y asignar un bloque de memoria, de tamaño suficientemente grande para almacenar una cantidad de datos”.

“Una desventaja de la función malloc es que los valores del vector dinámico no son inicializados, sino que contienen cualquier basura que pudiera haber estado ya en la memoria. Por suerte, la biblioteca `stdlib.h` ofrece también la función **calloc**, que significa clear allocation (asignación limpia)”.

# Arreglos dinámicos

## Ejemplo 1

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

int main() {
    //Bloque de declarativa
    float *calificaciones;
    int tamano, i;
    //Bloque de instrucciones
    printf("Ingrese cantidad de calificaciones: ");
    scanf("%d", &tamano);

    //se reserva 'tamano' posiciones, cada una ocupa 'sizeof(int)' bytes
    // las posiciones son inicializadas en ceros
    calificaciones = (float *) calloc(tamano, sizeof(int));

    for (i=0;i<tamano;i++)
    {
        printf("\n Introduzca la calificación N° %d", i+1);
        scanf("%f", &calificaciones[i]);
    }

    for (i=0;i<tamano;i++)
    {
        printf("\n Calificación N°%d es de %.2f", i+1, calificaciones[i]);
    }
    free(calificaciones);
    return 0;
}
```

```
#include <stdio.h>
#include <stdlib.h>
int main() {
```

```
    //Bloque de declarativa
    float **matriz;
    int filas, columnas, x,y;
    //Bloque de instrucciones
    printf("\n Ingrese cantidad de filas: ");
    scanf("%d",&filas);

    printf("\n Ingrese cantidad de filas: ");
    scanf("%d",&columnas);
    // se dimensiona el vector en base a su tamaño en bytes
    // en total se reservan 'tamano*sizeof(int)' bytes de memoria
    // Se crean los espacios de las filas
    matriz = (float **) calloc(filas, sizeof(int*));
    // Recorre cada espacio de la fila para crear los espacios de la columna.
    for (x=0;x<filas;x++)
    {
        matriz[x] = (float *) calloc(columnas, sizeof(int));
    }
    for (x=0;x<filas;x++)
    {
        for (y=0;y<columnas;y++)
        {
            matriz[x][y]=x+y;
        }
    }

    for (x=0;x<filas;x++)
    {
        for (y=0;y<columnas;y++)
        {
            printf("%.2f ",matriz[x][y]);
        }
        printf("\n");
    }

    free(matriz);
```

```
return 0;
}
```

# Arreglos dinámicos

## Ejemplo 2





UNIVERSIDAD  
TECNOLÓGICA  
DE PANAMÁ

