

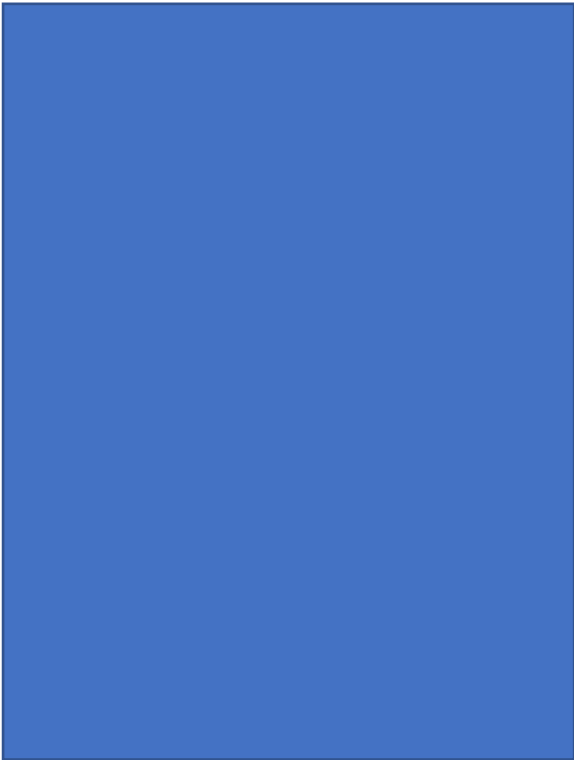
Uso de memoria y punteros

Herramientas de diseño electrónico

Memoria

Memoria dinámica

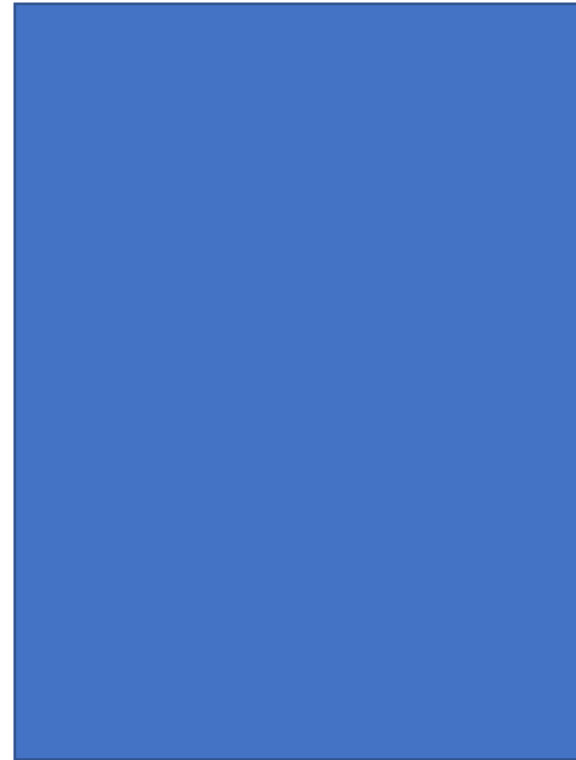
0xb547a8d3



0xb0016ec5

Memoria estática

0xa021cba7



0xa2c99725

Tamaño en memoria

Type	Size (bytes)
Char	1
Int	2 o 4
Float	4
double	8
Long	4
uint	2 o 4
Long double	10
Short int	2

`sizeof(<type>)`

Funciones de memoria

`malloc()` //reserva un espacio de memoria y retorna la dirección del espacio reservado

Ej: `int *p=NULL;`

`p=malloc(sizeof(int));`

`calloc()` //reserva un espacio de memoria y lo inicializa en cero. Retorna la dirección del espacio

`p=calloc(n, sizeof(<type>));` //n indica el # de bloques a reservar

Funciones de memoria

`realloc()` //redimensiona un espacio reservado anteriormente

`realloc(ptr, 10);` //10 es el numero de bytes reservados

`free()` //libera memoria previamente asignada

`free(ptr);` //el puntero ptr tiene un valor NULL

Punteros

- Tipo de variable que apunta a la dirección de memoria de otra variable
- Se indica el tipo de variable a la cual se va a apuntar
- Pueden apuntar a CUALQUIER tipo de variable

<type> *<pointer name>

int *ptr

int num=50; //se crea una variable tipo entero

int *ptr; //se crea un puntero para una variable tipo entero

ptr = # //se le asigna la dirección de num al puntero ptr

*ptr=3; //se cambia el contenido de la variable apuntada

ptr=7453; //**WARNING!!!!** Se cambia la dirección a la que se apunta

More in:

<https://hackaday.com/2018/04/04/the-basics-and-pitfalls-of-pointers-in-c/>

Sumador con punteros

```
int *ptr1, ptr2;
```

```
int main(void)
```

```
{
```

```
    ptr1=malloc(sizeof(int)); ptr2=malloc(sizeof(int));
```

```
    printf("Ingrese los números a sumar \n");
```

```
    scanf("%d\n", ptr1);
```

```
    scanf("%d\n", ptr2);
```

```
    printf("La suma de los números es: ", *ptr1+*ptr2);
```

```
}
```