

# Punteros en C: Una Explicación Detallada con **malloc**

Oscar Reyes Avila

November 19, 2023

## Introducción

En C, los punteros son variables especiales que almacenan direcciones de memoria. Este concepto es poderoso pero debe manejarse con cuidado para evitar problemas en el programa.

## Declaración de Punteros

Para declarar un puntero en C, utilizamos el operador de indirección **\***:

```
int *p;
```

## Asignación de Dirección de Memoria

Podemos asignar la dirección de memoria de una variable a un puntero utilizando el operador de dirección **&**:

```
int x = 10;  
int *p = &x;
```

## Acceso a Valor a través de Punteros

Para acceder al valor almacenado en la dirección de memoria apuntada por un puntero, utilizamos el operador de indirección **\***:

```
int y = *p;
```

## Uso de malloc

La función `malloc` se utiliza para asignar memoria dinámicamente en tiempo de ejecución. Devuelve un puntero al bloque de memoria asignado:

```
int *arr = (int*) malloc(5 * sizeof(int));
```

## Liberar Memoria con free

Es importante liberar la memoria asignada dinámicamente cuando ya no la necesitamos para evitar fugas de memoria:

```
free(arr);
```

## Ejemplo Completo

A continuación se muestra un ejemplo completo que muestra la declaración, asignación, acceso a valores y liberación de memoria usando punteros y `malloc` en C.

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

int main() {
    int x = 10;

    int *p = &x;

    int y = *p;

    int *arr = (int*) malloc(5 * sizeof(int));

    free(arr);

    return 0;
}
```