



Profesor: _____

Fecha: ____/____/____

Clase 14

Punteros: Dominando la Memoria en C

¡Hola a todos!

En la clase de hoy, nos adentraremos en el fascinante mundo de los **punteros** en C. Aprenderemos a trabajar con ellos para acceder y manipular directamente la memoria del ordenador, permitiéndonos llevar nuestro control del programa a un nuevo nivel.



Punteros

Título: Punteros: Dominando la Memoria en C**Introducción****Contenido:**

- Un **puntero** es una variable que almacena la dirección de memoria de otra variable.
- Permiten acceder de forma directa a la memoria del ordenador y modificar su contenido.
- Los punteros son una herramienta poderosa, pero también compleja, que requiere un uso cuidadoso.

Analogía: Imaginen los punteros como las etiquetas que se utilizan para identificar las diferentes ubicaciones en una biblioteca. Con ellas, podemos encontrar y acceder a la información que necesitamos de forma rápida y precisa.

Declaración de un puntero

- Se utiliza el tipo de dato de la variable que queremos apuntar seguido de un asterisco (*) para declarar un puntero.

Ejemplo:

```
int numero = 10;
int *punteroNumero; // Declara un puntero a una variable de tipo entero

punteroNumero = &numero; // Almacena la dirección de memoria de la variable "numero" en el puntero "punteroNumero"
```

Operadores de punteros

- Existen diferentes operadores para trabajar con punteros, como:
 - & (operador de dirección): Obtiene la dirección de memoria de una variable.
 - * (operador de desreferenciación): Obtiene el valor almacenado en la dirección de memoria que apunta un puntero.

Ejemplo:

```
int valor = *punteroNumero; // Almacena el valor al que apunta el puntero "punteroNumero" en la variable "valor"
*punteroNumero = 20; // Modifica el valor de la variable "numero" a través del puntero "punteroNumero"
```

Precauciones al usar punteros

- Es importante utilizar los punteros con cuidado para evitar errores y problemas en el programa.
- Se deben tener en cuenta las siguientes precauciones:
 - **Inicializar los punteros** antes de usarlos.
 - **Validar la memoria** antes de acceder a ella a través de un puntero.
 - **Evitar los accesos nulos** a memoria.
 - **Liberar la memoria** que ya no se usa.

Cierre:

- ¿Qué son los punteros en C?
- ¿Cómo se declara un puntero?
- ¿Qué son los operadores de dirección y desreferenciación?
- ¿Qué precauciones se deben tener al usar punteros?

Recuerda:

- Los punteros son una herramienta poderosa que puede mejorar la eficiencia y el control del programa.
- Es importante comprender cómo usarlos correctamente para evitar errores y problemas.
- La práctica es la mejor forma de aprender a usar los punteros de manera efectiva.

¡Espero que esta clase te haya dado una base sólida para dominar el mundo de los punteros en la programación en C!