



Punteros y Matrices en C - (6)

versión 1.1 - 1/04/2008

```

struct Alumno {
    char *nombre;
    int dni;
    char letraDNI;
    int edad;
};

// Define un tipo de estructuras llamadas "Alumno"

```

```

struct Alumno pedro = { "Pedro", 1355963, 'K', 22};

```

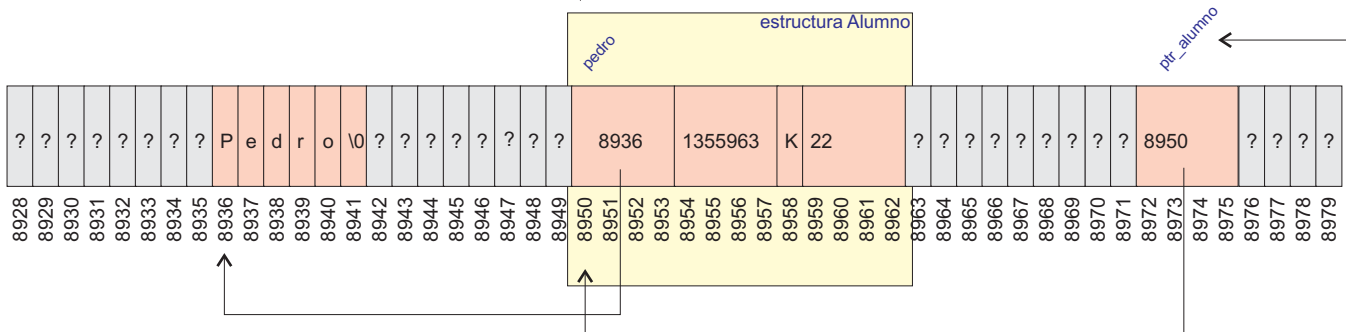
// Declara una variable llamada pedro, de tipo Alumno, que en su interior // contiene los datos indicados.

```

struct Alumno* ptr_alumno = &pedro;

```

// Declara una variable llamada ptr_alumno, que apunta a una estructura // de tipo Alumno y vale inicialmente la dirección de memoria de la variable pedro.



```

pedro.nombre
// Representa la cadena de caracteres del nombre

```

```

pedro -> nombre
// NO COMPILA, ya que pedro no es un puntero a una estructura
// sino que es la propia estructura

```

```

ptr_alumno.pedro
// NO COMPILA, ya que ptr_alumno no es una estructura, sino un puntero
// a una estructura

```

```

ptr_alumno -> edad
// Correcto, representa el campo edad de la estructura a la que apunta ptr_alumno

```

```

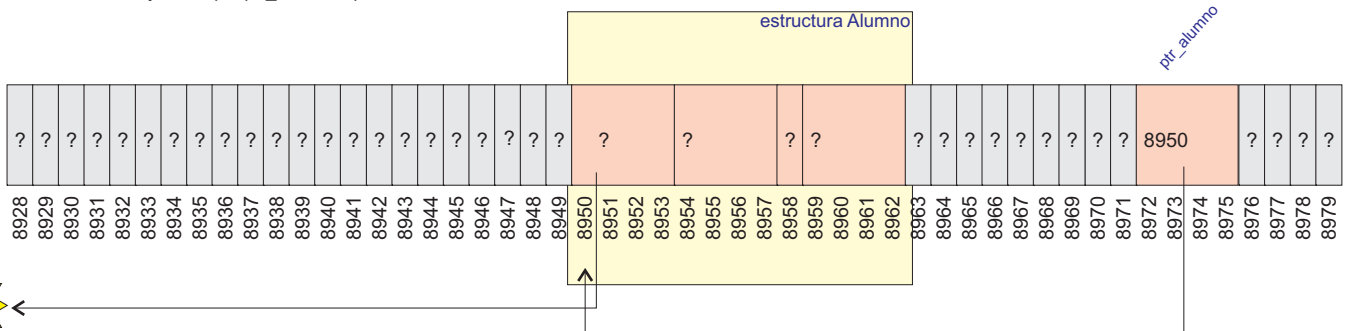
(*ptr_alumno).edad
// Idéntico al caso anterior.

```

```

ptr_alumno = (struct Alumno*)malloc( sizeof(struct Alumno) );
// Asigna una zona de memoria del tamaño de la estructura
// Alumno y hace que ptr_alumno apunte a ella

```



```

printf("%s", ptr_alumno->nombre);
// Imprime el campo nombre de la estructura a la que apunta
// ptr_alumno. Sin embargo, este campo no ha sido inicializado
// y puede estar apuntando a cualquier lado. Resultado : boom

```

