

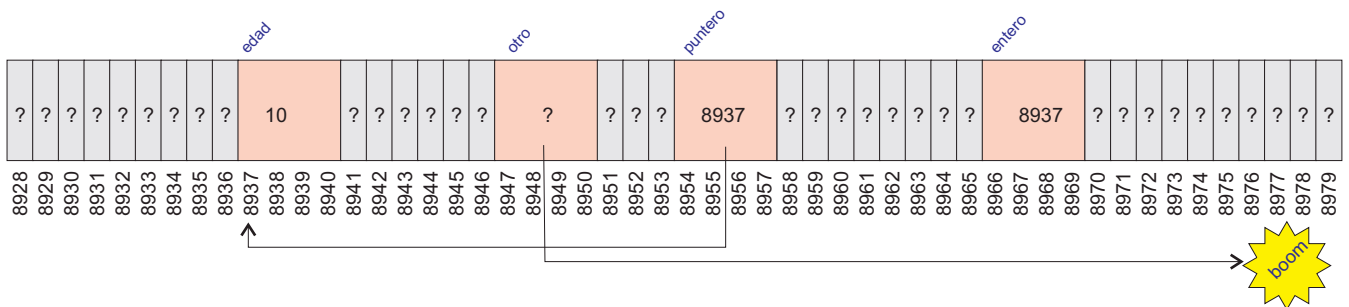
Punteros

```
int edad = 10;
// se asigna memoria para almacenar un int (4 bytes) ,y
// se inicializa a 10

int * puntero = &edad;
// se asigna memoria para almacenar un puntero a un
// entero (también 4 bytes), y se inicializa ese puntero con
// la dirección de memoria donde se encuentra edad (8937)
```

```
int entero = &edad;
// se asigna memoria para almacenar un int (4 bytes) , y se inicializa
// con la dirección donde se encuentra edad (8937)

int * otro;
// se asigna memoria para almacenar un puntero a un entero
// y no se inicializa
```



```
printf("%d", edad);
// Se imprime el valor de edad : 10
```

```
printf("%d", puntero);
// Imprime el contenido de la variable puntero, que es la dirección
// de memoria donde se encuentra edad - 8937
```

```
printf("%d", *entero);
// Dado que entero no fue declarado como puntero, a pesar de que
// contiene lo mismo que puntero, esta expresión NO COMPILA
```

```
printf("%d", puntero + 1);
// Dado que puntero apunta a un int, imprime la dirección
// de memoria donde iría el siguiente entero : 8941
```

```
printf("%d", *puntero + 1);
// Imprime el valor al que apunta puntero mas 1 : 11
```

```
printf("%d", entero);
// Imprime el contenido de la variable entero, que es la dirección de memoria
// de edad : 8937
```

```
printf("%d", *puntero);
// Imprime lo que hay en la dirección de memoria a la que apunta puntero
// (la 8937) : 10
```

```
printf("%d", *(int *) entero );
// Le indica al compilador - a pesar de que entero está declarado como int -
// que lo considere como si fuese realmente un puntero a un int, y en consecuencia
// imprima el dato al que apunta entero : 10
```

```
printf("%d", entero+1);
// Como entero no fue declarado como puntero, simplemente imprime el valor
// de entero mas 1 : 8938
```

```
printf("%d", otro);
// Imprime el valor de la variable otro : como no está inicializada, puede ser
// cualquier cosa
```

```
printf("%d", *otro);
// Imprime el valor del entero al que apunta otro, pero como otro no está inicializado
// podría estar apuntando a cualquier lado de la memoria. Resultado : boom
```