

# PUNTEROS "RAW" EN C++

- **Puntero:** variable que almacena una dirección de memoria ("apunta" a algo).
- **Declaración:** se debe indicar a qué tipo de dato apuntará. Ejemplo de puntero a int: `int* n;`
- Cada puntero tiene, también, una dirección de memoria propia.
- **Desreferencia:** operación que accede al valor guardado en la dirección de memoria apuntada.
- El valor "apuntado" puede estar en stack o en heap. Los punteros también pueden estar en ambas.
- Los valores guardados en heap deben accederse mediante punteros, ya que son variables sin identificador.
- Puede haber más de un puntero apuntando a la misma dirección. También puede modificarse un puntero.
- **Operación new:** reserva un espacio de memoria heap.
- **Operación delete:** libera un espacio de memoria heap.
- **Valor nullptr:** indica que el puntero no apunta a nada.



PROGRAMACIÓN  
DESDE CERO

"Fuga de memoria": si un dato en memoria heap no está apuntado por ningún puntero, ese dato continúa ocupando espacio pero ya no hay forma de acceder a él, ni siquiera para eliminarlo.

"Puntero colgante": habiendo dos punteros a la misma dirección, si se ejecuta *delete* en uno de ellos, el otro sigue apuntando a esa dirección, que ahora ha quedado inválida.