

Introducción a la programación en C

Manejo de memoria

Bach. Constantino Bolaños Araya

Escuela de Veranillo en HPC, 2018

Arreglos

Arreglos en C

- Los arreglos son un tipo de estructura de datos.
- Es una colección de variables del mismo tipo.

```
int calificaciones[4];
```

```
int serie[] = {1, 1, 2, 3, 5, 8, 13};
```

```
calificaciones[0] = 7;
```

```
calificaciones[4] = 8; // Que pasa aqui?
```

```
float res = serie[0] - serie[1] + serie[2];
```

Punteros y referencias

Punteros y referencias

- Cada variable es en realidad una ubicación en memoria, en la cual está almacenado el valor de dicha variable.
- Un *puntero* es una variable cuyo valor es la dirección de otra variable.
 - En otras palabras, un puntero **apunta** a otra variable

```
int x = 10;
```

```
int *b = &x; // b apunta a x
```

```
print("valor de x: %d", x) // 10
```

```
print("valor al que apunta b: %d", *b) // 10
```

Memoria dinámica

- Los arreglos nos permiten guardar un número de variables en un conjunto con un tamaño determinado.
- ¿Y si no sabemos cuántos elementos queremos guardar de antemano?
- Memoria dinámica nos permite solicitar memoria “en el vuelo”
- **¡IMPORTANTE!** La memoria debe solicitarse, y luego liberarse, **siempre**.

- Hay cuatro funciones empleadas para trabajar con memoria dinámica en C:
 1. `void *calloc(int num, int size)`: alojar un arreglo de `num` elementos, cada uno de `size` bytes de tamaño.
 2. `void free(void *address)`: liberar un bloque de memoria especificado por `address`.
 3. `void *malloc(int num)`: alojar `num` bytes sin inicializar.
 4. `void *realloc(void *address, int newsize)`: realojar memoria extendiéndola en tamaño a `newsize`.