

**AEDD - Guía Práctica 12: Punteros**

**Ejercicio 1:** Para cada uno de los siguientes enunciados, escriba el código correspondiente. Suponer que se han declarado las siguientes variables:

```
float num1 = 3.1415926, num2 = 0.12345;
```

- (a) Declarar un puntero a float llamado `pnum`.
- (b) Asignar la dirección de la variable `num1` a `pnum`.
- (c) Imprimir el valor de la variable apuntada por `pnum`.
- (d) Asignar el valor de la variable apuntada por `pnum` a la variable `num2`.
- (e) Imprimir el valor de `num2`.
- (f) Imprimir la dirección de `num2` y la dirección almacenada en `pnum`. ¿Es la misma dirección?

**Ejercicio 2:** Para cada uno de los siguientes enunciados, escriba el código correspondiente.

- (a) Declarar un arreglo de tipo float, llamado *numeros* e inicialice los elementos con los siguientes valores: {0.1, 0.2, 0.3, 0.4, 0.5}. Suponga que la constante `SIZE` ha sido definida como 5.
- (b) Declarar un puntero `pnum` que apunte a una variable de tipo float.
- (c) Imprimir los elementos del arreglo utilizando subíndices de arreglos.
- (d) Idem (c) pero utilizando únicamente la variable `pnum`.
- (e) Suponiendo que `pnum` apunta al principio del arreglo, ¿Cuál es la dirección referenciada por `(pnum+3)`? ¿Y su valor?
- (f) Suponiendo que `pnum` apunta al cuarto elemento del arreglo (`numeros[3]`). ¿A qué elemento del arreglo apunta `(pnum-2)`?

**Ejercicio 3:** Crear un array de 10 números aleatorios y después un array de 10 punteros donde cada uno apunte a los números anteriores.

**Ejercicio 4:** Dado un vector de 10 elementos = {1, 2, 3, 4, 4, 7, 8, 9, 5, 4}, escribir un programa en C++ (haciendo uso de punteros) que muestre las direcciones de memoria de cada elemento del vector.

**Ejercicio 5:** Dada una cadena, invertirla haciendo uso de la aritmética de punteros.

**Ejercicio 6:** Una aplicación de gestión de fotografías en tu teléfono celular tiene definido el catálogo de fotos de la siguiente forma:

2