

# Reporte Práctica Arreglo

Chávez Olea Mauricio

Navarrete Baltazar Mario

1. Explica la estructura de tu código, explica en más detalle tu implementación del método *obtenerIndice()*:

**R: El método obtenerIndice() es el más importante, ya que con este método se implementa los demás métodos de la clase. Este método lo implementamos basándonos en la definición teórica de la práctica, de la siguiente manera:**

La definición teórica nos dice que la posición (o el índice) es igual a la suma iniciando en 1 hasta n (donde n es igual a la longitud del arreglo de índices menos 1), de  $f_j * i_j$ , donde  $f_j$  es igual a 1 si  $j=n$ , o es el producto desde  $k=j+1$  hasta n de las dimension k si  $1 \leq j < n$ .

Esto lo implementamos con ciclos for anidados, en el cual el producto se implementa como un for anidado en el for que hacia nuestra suma. El for de la suma iniciaba en cero e iba hasta n, para asegurarnos que pasaba por cada uno de las localidades del arreglo de índices pasado como parámetro. Este for declaraba una variable entera "f", inicializada como 1. Entonces, metimos un for para hacer nuestro producto, el cual iniciaba en i+1 (donde i era donde se encontraba el for de la suma) e iba hasta n. En este for se multiplicaban todas las dimensiones menos la primera, y pos eso asignamos  $f=1$ , para no afectar el resultado de ese producto. Despues saliamos de este for para hacer la suma correspondiente, la cual sumaba el f obtenido con el for pasado multiplicado a el indice[i].

Los demás métodos solo obtienen o almacenan elementos al arreglo unidimensional, obteniendo su indice correspondiente con el método obtenerIndice() y buscando su localidad en el arreglo unidimensional.

2. ¿Cuál es el orden de complejidad de cada método?

**R: La complejidad de cada método es  $O(n^2)$ , ya que almacenarElemento() y obtenerElemento() ocupan el método obtenerIndice(), el cual tiene un for anidado en otro for. Sabemos que la complejidad de cada for es lineal, y al anidarlos en otro for, nos queda una complejidad cuadrática.**