

# UT3-TA1

## Ejercicio 1

### Consumo de variables:

- Nombre del producto: String →  $2 \times (\text{N}^\circ \text{ de char}) + 38 \text{ bytes}$
- Código de identificación: String →  $2 \times (\text{N}^\circ \text{ de char}) + 38 \text{ bytes}$
- Precio unitario: Float → 4 bytes

### Peor de los casos:

Nombre de producto: 30 char máximo

$$2 \times 30 + 38 = 98$$

$$2 \times 20 + 38 = 78$$

Código de identificación: 20 char máximo

4

Precio unitario: 4 bytes

180 bytes

### Ventajas y desventajas de arrays y listas encadenadas:

#### Arrays

##### ★ Ventajas

- Acceso rápido a los elementos por índice
- Menos sobrecarga de memoria por punteros

##### ★ Desventajas:

- Tamaño fijo: si se llena hay que redimensionar, lo cual es costoso
- Se desperdicia memoria en caso de baja ocupación

#### Listas Encadenadas:

##### ★ Ventajas

- Tamaño dinámico: Crece y decrece según las necesidades
- Facilita la inserción y eliminación de elementos

##### ★ Desventajas:

- Acceso más lento: Para buscar un elemento, hay que recorrer los nodos (con costo)
- Mayor uso de memoria debido a los punteros

Caso - Suponemos 10 productos

Array	Lista
<p>Memoria objeto producto</p> <p><math>180 \text{ bytes} \times 10 \text{ objetos} + 12 \text{ bytes}</math> del head del array</p> <p>1812 bytes en total en Objetos</p>	<p>Memoria objeto producto</p> <p><math>184 \text{ bytes} \times 10 \text{ objetos} + 24</math> por lista encadenada <math>+ 16 + 220</math> por nodo que contiene la referencia al nodo y el objeto <math>+ 40</math></p> <p>2140 bytes en total en objetos</p>

Observación: Creando un array de tamaño justo es más efectivo respecto a las listas. El problema es cuando se guarda memoria sin usar

Respuesta pregunta 1:

Costo memoria array: El costo es menor pero no es conveniente si no se sabe exactamente cuántos objetos se van a necesitar

Costo memoria lista encadenada: El costo es mayor pero es más eficiente ya que no es más fácil insertar o eliminar los objetos

Respuesta pregunta 2:

El problema con el array sería la redimensión del mismo que implicaría mucho tiempo en pasar los productos viejos al nuevo array más grande.

Si se elimina un producto porque se deja de vender habría que correr todos los productos un lugar en el array

Con las listas no existe este problema ya que las listas se pueden redimensionar sin necesidad de pasar los productos a una nueva lista

Respuesta pregunta 3:

Array: es solo eficiente si se sabe exactamente cuántos productos se van a vender (y que ninguno se deje de vender)

Lista: Es menos eficiente en términos de costo pero más eficiente a la hora de agregar o eliminar productos (no hay que redimensionar)