

# Ejercicios de Listas



## Relación de ejercicios

**Ejercicio 1.** Implementa un programa que declare una lista y almacene el nombre de 5 miembros de tu familia.

**Ejercicio 2.** Partiendo del ejercicio anterior y con la lista ya inicializada, crea un método que permita insertar el nombre de un nuevo miembro familiar en la posición que se desee. El método debe recibir como parámetros: la lista de familiares, la posición que queremos que ocupe el nuevo miembro en la lista y el nombre del nuevo miembro familiar.

**Ejercicio 3.** Partiendo de una lista de familiares inicializada (con sus nombres), implementa un método que reciba por parámetro dicha lista y muestre por pantalla los nombres de los familiares.

**Ejercicio 4.** Partiendo de una lista de familiares inicializada, pide al usuario que introduzca un número de familiar y muestre por pantalla su nombre (el del familiar). La lista de familiares debe ser una variable (atributo) de la clase.

**Ejercicio 5.** Partiendo de una lista de familiares inicializada, pide al usuario que introduzca un número de familiar (*posFamiliar*), un nuevo nombre de familiar (*nuevoFamiliar*) y cambie el nombre del familiar que ocupa la posición *posFamiliar* por el nombre *nuevoFamiliar*.

**Ejercicio 6.** Implementa un programa que tenga los siguientes métodos para poder operar sobre una lista que contiene los nombres de los familiares. Influyen las mayúsculas y las minúsculas.

- Indique si la lista contiene un determinado nombre.
- Indique la primera vez que aparece un nombre.
- Indique la última vez que aparece un nombre.
- Elimine un nombre de la lista.
- Devuelva una porción de la lista. Por ejemplo, del familiar 2 al familiar 5.
- Junte dos listas y muestre la unión de las mismas. Por ejemplo, partiendo de la lista de abuelos paternos y la lista de abuelos maternos, debe existir una lista que contenga los nombres de los 4 abuelos y los muestre por pantalla.
- Indique si dos listas son iguales.
- Borre todos los nombres de la lista.

**Ejercicio 7.** Implementa un programa que lea frases por pantalla hasta que el usuario introduzca la palabra END. Una vez que el usuario introduzca dicha palabra se debe...

- a) Mostrar por pantalla una única cadena de texto que contenga todo el texto leído.
- b) Obtener el número de palabras del texto completo.
- c) Indicar cuántas veces aparece en el texto una palabra que leamos por pantalla.  
Obtendremos por consola: "La palabra .... se repite n veces en el texto"
- d) Asegurar que todas las frases introducidas tengan la primera letra en mayúsculas y el resto de la frase esté en minúsculas.

**Ejercicio 8.** Implementa un programa que permita al usuario comprar tantos productos como desee y una vez que haya comprado, se le muestre la factura. Un ejemplo de factura está en la imagen. Para implementar el programa debes crear un menú que le pregunte al usuario si quiere seguir comprando o desea ya pagar (que es cuando se le mostrará la factura).

|         |      |        |          |
|---------|------|--------|----------|
| *****   |      |        |          |
| FACTURA |      |        |          |
| *****   |      |        |          |
| Nombre  | Uds. | Precio | Subtotal |
| -----   | ---- | -----  | -----    |
| Fresa   | 2,80 | 3,25   | 9,10     |
| Tomate  | 5,00 | 2,05   | 10,25    |
| Plátano | 4,25 | 1,50   | 6,38     |
| -----   |      |        |          |
| TOTAL:  |      |        | 25,73    |

**Ejercicio 9.** Implementa un programa que haga al usuario 5 preguntas que deban ser respondidas con un (Sí, Si, S,Y) o con un (No, N). Las 4 primeras respuestas se consideran afirmativas y las dos últimas se consideran negativas. Debemos permitir que el usuario las introduzca en mayúsculas o minúsculas así como con tildes o sin ellas. Las preguntas deben formar parte de una lista sobre la que se deberá ir iterando (recorriendo pregunta a pregunta). Finalmente, debemos mostrar un informe con el porcentaje de respuestas afirmativas y negativas que ha dado el usuario.