



## Guía de Ejercicios: Pilas, Colas y Listas

### Resolver los siguientes ejercicios utilizando las estructuras de datos Pila y Cola:

1. Escribir un programa que lee una secuencia de caracteres y los imprime en orden inverso.
2. Escribir un programa que verifique si una palabra es palíndroma (es decir que una cadena es igual a su inversa. Por ejemplo: las cadenas "1456541" y "145541" son palíndromos).
3. Escribir un método que, dadas dos colas de enteros, construya una cola con el resultado de poner una a continuación de la otra. Por ejemplo: por ejemplo:  
  
si  $q_1 = [1, 2, 3]$  (con 1 al frente y 3 al final) y  $q_2 = [5, 7]$ , el resultado es  $[1, 2, 3, 5, 7]$  (con 1 al frente y 7 al final).
4. Escribir un método que evalúa si una cadena de paréntesis, corchetes y llaves está bien balanceada y devuelve `true` si está bien balanceada y `false` si está mal balanceada. Por ejemplo: `[()]{ }{ [ ( ) ] ( ) }` debe devolver `true`, mientras que `[ ( ] )` debe devolver `false`.
5. Escribir un programa que evalúa expresiones aritméticas escritas en notación polaca inversa. Por ejemplo: si el programa recibe la expresión `4 7 8 * +` tiene que devolver como resultado el número 60. Si recibe la expresión `4 + 7 * 8` tiene que devolver un error.
6. Escribir un método para determinar si una secuencia de caracteres de entrada es de la forma:  
  
`X & Y` siendo X una cadena de caracteres e Y la cadena inversa. El carácter & es el separador.

### Resolver los siguientes ejercicios utilizando Listas e Iteradores:

1. Escriba un programa que imprima por pantalla la lista ordenada por Apellido de clientes de un negocio, y si tienen el mismo Apellido ordenada por Nombre. Para esto construya la clase `cliente` con los atributos: nombre, apellido, teléfono y D.N.I, y que los datos que se muestren de un cliente sean: Apellido, Nombre y Teléfono.
2. Escribir un método que reciba dos listas de números enteros ordenados y devuelva una tercera lista de números enteros, con todos los elementos de las listas que recibió, manteniendo el orden.  
  
Por ejemplo, si recibe las listas `[1,3,5,6,8,9,10]` y `[2,4,7]`, debe devolver `[1,2,3,4,5,6,7,8,9,10]`.
3. Escribe el método `eliminarDuplicados`, que recibe una lista de enteros y devuelve una nueva lista con los mismos elementos, pero eliminando los repetidos.  
  
Por ejemplo, si la lista de entrada es: `[1, 2, 2, 1, 4, 2, 4, 3]`, la salida sería: `[1, 2, 4, 3]`.
4. Escriba un método que reciba una lista de enteros y la devuelva invertida.  
  
Por ejemplo, si recibe la lista `[1, 2, 3, 4, 5]` devolverá la lista `[5, 4, 3, 2, 1]`.
5. Escriba un método que reciba una lista de enteros y devuelva `true` si la lista contiene dos elementos tales que el número resultante de sumarlos también sea un elemento de la lista.



6. Escriba un método que devuelva true si una lista de enteros es sublista de otra. Por ejemplo:
- si tenemos las listas  $L1 = [22, 14, 6]$  y  $L2 = [39, 41, 17, 22, 14, 6, 3, 11, 73, 81]$  entonces el método devolverá true porque L1 es una sublista de L2.
- Pero si tenemos otra lista  $L3 = [39, 41, 22, 17, 14, 3, 11, 73, 6, 81]$ , vemos que L1 no es sublista de L3 por lo que el método llamado con L1 y L3 debe devolver false.