

## Algoritmos y Estructura de Datos

Guía de ejercicios estilo 2da evaluación parcial - parte 1 (v. 2.1)

**Temas:** Archivos. Estructura cola, pila, lista, lista doblemente enlazada.

1) **Calculadora.** Se necesita validar la sintaxis de las ecuaciones ingresadas. Para ello, se debe analizar el equilibrio de los signos agrupadores (), [] y {}. El algoritmo debe agregar a la estructura de datos cada vez que encuentre el símbolo abierto, es decir: (, [ o { y evaluar si el siguiente símbolo es el mismo pero cerrado, o sea: ), ] o }, por ejemplo:

Correcto: ( ) ( ( ) { ( ( ) ) }

Correcto: ( ( ( ) ( ) { ( ( ) ) } ) )

Incorrecto: ) ( ( ) { ( ( ) ) }

Incorrecto: { ( [ ] }

Incorrecto: (

Se pide:

Crear la función **evaluar()** que reciba como parámetro un array de caracteres e itere sus elementos evaluando si existe alguna inconsistencia. Retornará true si es correcto. Probar con:

entrada1 = {'a','+','b','\*','-','[','(','-1','\*','c',')','/','7',']','\*','d'};

entrada2 = {'3a','+','{','-5x','-a','+','(','9x','-a','-x',')','}'};

2) **Videojuego.** Se debe crear la estructura de datos que dé soporte a un selector de autos de un simulador de carreras. Cada nodo deberá contener una estructura datos con los siguientes atributos: nombre, ruta a la imagen (cadena de caracteres), velocidad (entero), aceleración (decimal), peso (decimal) y coeficiente aerodinámico (decimal).



Se pide: El objetivo es definir una estructura que permita recorrer los nodos en ambos sentidos con la función **mostrar()**, que recibirá por parámetro un puntero a la función que determine el sentido (**mostrarIzqDer()** y **mostrarDerIzq()**), además de la lista de autos. Además se deberá implementar la función **insertar()** que insertará un nuevo auto a la lista. El nodo actual (Pagani Zonda), se obtiene invocando la función **irA(saltos-respecto-del-primero)** que retorna el elemento seleccionado como se muestra en la figura. Implementar las funciones de soporte que considere necesarias, por ejemplo: **tamano()**, **vacía()**, **inicio()**, **ultimo()**, **obtener(nombre)**.