**Pseudocodigo:**

comprovarExpressio le pasamos un array de símbolos

**¿Es el primer símbolo abierto ( o { ?**

**SI :** Lo introducimos en la pila (push)

**NO:** Devolvemos falso (la expresión no es correcta, ya no hace falta seguir).

Recorremos el array de símbolos

element = Símbolo del array (este va aumentando a cada vuelta)

**¿Es abierto element?**

**SI:** Lo introducimos en la pila (push)

**NO:** **¿La pila no está vacía?**

**SI:** buffer coge el elemento el ultimo elemento de la pila.

**¿Es correcto el paréntesis o la llave del buffer?**

**SI:** Sácalo de la pila (pop).

Devolvemos la pila vacía.

**Función auxiliar CheckOpen** (Le pasamos el símbolo a comprobar)

Creamos un OpenParentesis y un OpenClaudator

Preguntamos si el símbolo es de la misma clase que el OpenParentesis o el OpenClaudator, si es True es que es el mismo símbolo, de ser falso consideraremos que no es open.

**Función auxiliar CorrectParentesi** (Se le pasa el top de la pila, que será un símbolo open y el símbolo a comprobar)

Crea un OpenParentesi y un ClosingParentesi

Preguntamos si el primer elemento es de la clase openParentesi y el segundo elemento es de la clase ClosingParentesi.

**Función auxiliar CorrectClaudator.** Exactamente lo mismo que CorretParentesi pero tratando con llaves donde se tratan paréntesis.

**Ejemplo y funcionamiento de la pila**

