

# Segundo Parcial de Laboratorio

## Algoritmos y Estructura de Datos II

### TEMA A

#### Ejercicio 3

En `balanced.c` hay una nueva implementación del algoritmo de chequeo de delimitadores balanceados. En esta ocasión se utilizará una cola para guardar los delimitadores de la expresión de entrada. Deben completar el algoritmo utilizando las funciones del TAD Cola listadas a continuación:

Función	Descripción
<code>queue queue_empty(void)</code>	Crea una cola vacía
<code>queue queue_enqueue(queue q, queue_elem e)</code>	Inserta un elemento a la cola
<code>bool queue_is_empty(queue q);</code>	Indica si la cola está vacía
<code>unsigned int queue_size(queue q)</code>	Obtiene el tamaño de la cola
<code>queue_elem queue_first(queue q)</code>	Obtiene el primer elemento de la cola (el próximo a atender).
<code>queue queue_dequeue(queue q)</code>	Quita un elemento de la cola
<code>queue queue_destroy(queue q)</code>	Destruye una instancia del TAD Cola

El programa **no debe tener *memory leaks***

#### Ejercicio 4

En `queue.c` se da una nueva implementación incompleta del TAD Cola que deben completar. La implementación utiliza listas enlazadas de una manera particular. Analizar cómo se almacenan los elementos en la cadena de nodos y terminar de implementar las funciones `queue_first()` y `queue_dequeue()`. **No pueden cambiar las implementaciones del resto de las funciones**, solo pueden tocar las funciones marcadas para completar. Para verificar que esta nueva implementación del TAD funciona correctamente, pueden utilizar el programa del ejercicio anterior.