



- Nombre: Apellidos:
- DNI: Grupo (GII / GISI / GIC):
- Duración del examen: 2 horas (90 minutos ejercicios + 30 minutos test)
- Las respuestas a los ejercicios deberán escribirse en pseudocódigo y estar claramente justificadas, es decir, acompañadas de una breve y clara explicación.
- En todas las preguntas pueden incluirse operaciones auxiliares, debidamente justificadas, si se considera necesario.
- En el apartado a) de las preguntas deben especificarse claramente las operaciones básicas del TAD (tipos de entrada y salida de las mismas y ecuaciones de definitud) y describir su funcionamiento.

### EJERCICIOS (7,6 PUNTOS)

1. (2,5 PUNTOS) En una oficina de correos se almacenan los documentos certificados en dos pilas diferentes, situadas en cada una de las dos mesas de atención al cliente. Cada certificado se identifica con un número de registro por orden de llegada, independientemente de la mesa, de forma que ambas pilas están ordenadas (en la cima está el documento que ha llegado el último a la mesa). Al finalizar el día, se necesita una estructura que facilite el reparto de los certificados por orden de llegada.

Para facilitar la gestión de los documentos respetando el orden de llegada, suponiendo conocida la función *es\_registro\_menor*:  $\text{doc doc} \rightarrow \text{boolean}$ , se pide:

- a) Dar las especificaciones de los TAD's PILA[DOCUMENTO], COLA[DOCUMENTO] y LISTA[DOCUMENTO]. (0,75 pts)
- b) Crear una cola con todos los documentos ordenados según el número de registro, es decir, comenzando por el que llegó antes. Pueden utilizarse estructuras auxiliares. (1 pto)
- c) Crear una lista con todos los documentos ordenados según el número de registro, es decir, comenzando por el que llegó antes. No pueden utilizarse estructuras auxiliares. (0,75 pto)

Ejemplo: *Pila\_mesa\_1*: doc1-doc3-doc4-doc6-doc7-doc9 ← cima  
*Pila\_mesa\_2*: doc2-doc5-doc8-doc10 ← cima

a) Cola: doc1-doc2-doc3-doc4-doc5-doc6-doc7-doc8-doc9-doc10

↑  
Inicio

↑  
fin

b) Lista: doc1-doc2-doc3-doc4-doc5-doc6-doc7-doc8-doc9-doc10

↑  
Inicio

↑  
fin



2. **(2,5 PUNTOS)** Dar la especificación del tipo abstracto de datos `ARBOL_GENERAL[ENTEROS]` (0,25 pts) y escribir las siguientes operaciones:
- a) *esSumaRaices*: *árbol\_general*  $\rightarrow$  *boolean*, comprueba que, en todo el *árbol\_general*, cada una de las raíces es suma de las raíces de los árboles del bosque o árboles hijos. (1,25 pts)
  - b) *fila*: *árbol\_general* natural  $\rightarrow$  lista, que forma una lista con todos los enteros del árbol que se encuentran en la profundidad indicada por el natural. (1 pto)
3. **(2,6 PUNTOS)** Dar la especificación del tipo abstracto de datos `ARBOL_BINARIO[OBJETO]` (0,25 pts), y suponiendo conocida la operación `_>=_`: objeto objeto  $\rightarrow$  bool, implementa las siguientes operaciones (pueden ser parciales, o necesitar otras funciones auxiliares):
- a) *es\_ABB?*: *a\_bin*  $\rightarrow$  bool, que comprueba si un árbol binario cumple los requisitos para ser un árbol binario de búsqueda. (0,75 pts)
  - b) *semicompleto*: *árbol\_bin*  $\rightarrow$  bool, comprueba si un árbol binario es semicompleto. (0,75 pts)
  - c) *monticulo\_maximos*: *árbol\_bin*  $\rightarrow$  bool, comprueba si un árbol binario es un montículo de máximos. (0,85 pts)