

APELLIDOS:	NOMBRE:
TITULACIÓN: __GII __GII+ADE __GII+MAT __GIS __GIS+GII __GIS+MAT	
DNI:	
Duración: 1:45 h.	

Está terminantemente prohibida la tenencia de dispositivos electrónicos con capacidad de procesamiento y comunicación (móviles, tablets, smartwatches, etc.) durante la realización de la prueba. Cualquier indicio de su posesión durante la realización de la prueba será motivo de expulsión. Se sugiere dejar 1 hora de esfuerzo en la resolución del ejercicio 5. Contestar en la hoja de enunciados los 4 primeros ejercicios.

1) (Prueba escrita teoría, 0.5 puntos) Defina qué se entiende por Tipo Abstracto de Datos, y explique los tipos de abstracción que se han considerado en la asignatura.

2) (Prueba escrita teoría, 2 puntos) ¿Cuál es el orden de complejidad de las siguientes operaciones en las estructuras mencionadas? Se considera imprescindible justificar la respuesta.

a) Inserción de un elemento en una Pila implementada como un array.

b) Inserción de un elemento en una Cola de Prioridad implementada como una estructura dinámica con enlace simple apuntando a la cabecera.

c) Búsqueda de un elemento en un **Árbol Binario** implementado como una estructura dinámica con enlaces al hijo derecho e izquierdo

d) Búsqueda de un elemento en un Árbol Binario **de Búsqueda** implementado como una estructura dinámica con enlaces al hijo derecho e izquierdo.

- 3) (Prueba escrita teoría, 1.5 puntos) Considere las siguientes especificaciones algebraicas de operaciones sobre listas ¿qué operaciones son Misterio y Misterio2? Explique su respuesta

OPERACIONES

Misterio: TipoLista \rightarrow TipoElemento

Misterio2: TipoLista \rightarrow Booleano

...

ECUACIONES

$$\text{Misterio}(\text{Construir}(\text{elemento}, \text{lista})) = \text{SI EsVacia}(\text{lista}) \rightarrow \begin{matrix} \text{elemento} \\ | \text{Misterio}(\text{lista}) \end{matrix}$$

Misterio2(elem, CrearVacia) = FALSO

Misterio2(elem, Construir (elemento, lista)) = (elem = elemento) Ó Misterio2(elem, lista)

- 4) (Prueba escrita práctica, 3 puntos) Defina los tipos de datos para representar en una unidad de PASCAL un árbol binario. A partir de su representación, codifique las operaciones `NumNodos`, que calcula el número total de nodos de un árbol dado; `Altura`, que calcula la altura de un árbol; y `Frontera`, que, dado un árbol, devuelve una lista con los nodos hoja del árbol. Puede suponer que dispone de las operaciones necesarias de listas en una unidad llamada `ULista`.

5) (Prueba escrita práctica, 4 puntos) METRO MADRID

La red de Metro de Madrid, con 12 líneas, 267 estaciones, 283 km lineales es sexta del mundo por kilómetros, tercera de Europa y quinta del mundo por número de estaciones. La red de Metro se puede representar mediante una estructura de tipo grafo, en la que cada estación tiene una lista de estaciones adyacentes, que son aquéllas con las que está conectada directamente, una para la cabecera, dos para una estación típica y dos, o más de dos, para una estación en la que se pueda hacer transbordo entre líneas. Se pide:

- a) [0.5 puntos] Teniendo en cuenta las características del grafo conexo y no dirigido que vamos a generar, se debe justificar cuál es la implementación más adecuada. A continuación, se pide definir los tipos de datos que van a soportar la funcionalidad del TAD MetroMadrid, haciendo la descomposición de tipos en unidades que considere más adecuada, y teniendo en cuenta el grafo y el elemento de tipo *TStation*, que contendrá el nombre de la estación como una cadena de 25 caracteres máximo y una variable puntero a una estructura *TInfoStation* que ha desarrollado una empresa consultora externa, y que se unirá al proyecto a través de la unidad *uInfoStation*.
- b) [2 puntos] Implementar la funcionalidad *InsertarLínea*, que recibe una lista de estaciones, todas correspondientes a una misma línea, e inserta estas estaciones de manera adecuada en la estructura grafo del metro. Será necesario codificar cada uno

de los subprogramas de las unidades auxiliares donde se define cada tipo de dato (excepto los de la unidad creada por la empresa consultora). Considere el caso de que una estación pertenezca a varias líneas, y que para copiar la información de tipo *TInfoStation* se ofrece el subprograma *CopyInfoStation* que recibe 2 valores de tipo *TInfoStation* (no de tipo puntero a *TInfoStation*):

PROCEDURE *CopyInfoStation*(**VAR** *dest: TInfoStation*; *ori: TInfoStation*);

{dest es la variable de TInfoStation a la que se le asigna el valor de la variable ori}

- c) [1.5 puntos] Dada una estación que se toma de referencia, se desea saber a qué distancia de ella se encuentra el resto de estaciones del plano de metro. De esta manera, dada una estación de referencia, sus adyacentes se encuentran a distancia 1, los adyacentes a éstas a distancia 2, etc... Implementar la funcionalidad que permite calcular la distancia de cualquier estación del plano de metro a una estación dada de origen. Puede suponer que dispone de la funcionalidad de los TAD pila, cola y conjunto en las unidades UPila, UCola y UConjunto, respectivamente...