

## PROGRAMACIÓN 2 - Unidad 5: Árboles

### Ejercicios entregables semana 2

21. Escriba el pseudocódigo de un algoritmo que reciba un árbol binario y cree un nuevo árbol binario copia del anterior. Dar, a continuación, el código C del algoritmo propuesto. Suponga, para ello, los tipos y estructuras de datos definidas en el ejercicio 3.
22. Escriba el pseudocódigo de un algoritmo que acepte un árbol binario y lo modifique de forma que sea la imagen refleja del original (es decir, un árbol en el que todos los subárboles derechos del árbol original son ahora subárboles izquierdos y viceversa). Dar, a continuación, el código C del algoritmo propuesto.

### Instrucciones de entrega:

Se os proporciona un programa de prueba *trees.zip* que contiene una biblioteca de árboles **libtree.a** con la que podéis probar vuestro código. El código se puede compilar usando la Makefile proporcionada, que genera un ejecutable `test_tree`.

La biblioteca `libtree.a` ha sido probada en Linux 64 bits, <http://pc-virtual.uam.es> (Ubuntu) y en la app Ubuntu de Windows 10. (Para mac, usar `libtree-mac.a`)

De forma **excepcional**, solo para que podáis realizar los ejercicios, el fichero de cabecera *tree.h* contiene las estructuras de `BTNode` y `BSTree`, que deberían ser privadas y estar definidas en *tree.c* (fichero no proporcionado). Esto os permite acceder a estas estructuras en las funciones que defináis.

El fichero `readme.md` (en formato markdown) contiene información sobre la biblioteca y el programa proporcionado. Podéis usar su funcionalidad para crear árboles a partir de ficheros y así probar fácilmente vuestro código. El zip incluye también algunos ficheros de prueba.

Debéis **rellenar las partes de `test_tree.c` que se indican el fichero con el pseudocódigo y código** para ambas preguntas, definiendo las funciones *tree\_copy* y *tree\_mirror* en el espacio indicado. **Subid exclusivamente el fichero `test_tree.c` modificado.**