

# Programación II, 2016-2017

Escuela Politécnica Superior, UAM

## Práctica 1: Estructuras de Datos y Tipos Abstractos de Datos

### PARTE 2: PREGUNTAS SOBRE LA PRÁCTICA

---

1. ¿Sería posible implementar la función de copia de nodos empleando el siguiente prototipo `STATUS Node_copy(Node pDest, const Node pOrigin);` ? ¿Por qué?

R.1 No porque si no pasas los puntos como punteros a la estructura, es decir, paso por referencia, estas pasando por valor y cambiaras los valores de la estructura dentro de la función únicamente, pero al salir de la función tendrás los mismos valores en el main.

2. ¿Es imprescindible el puntero `Node*` en `int Node_print(FILE * pf, const Node* p);` o podría ser `int Node_print(FILE * pf, const Node p);` ?

R2. Es imprescindible que al momento de recibir un puntero al Nodo ya que este consta de una estructura la cual sera usada en diferentes funciones dentro de Nodo.c por lo tanto para poder leer un Nodo creado por la function `Node_ini` sera imprescindible recibirlo por puntero.

3. ¿Qué cambios habría que hacer en la función de copiar nodos si quisiéramos que recibiera un nodo como argumento donde hubiera que copiar la información? Es decir, ¿cómo se tendría que implementar si en lugar de `Node* Node_copy(const Node* pOrigin)`, se hubiera definido como `STATUS Node_copy(const Node * pSource, Node * pDest)`? ¿Lo siguiente sería válido: `STATUS Node_copy(const Node * pSource, Node ** pDest)`? Discute las diferencias.

R.3 Para ambos casos, lo único que cambiaría en la función es que te ahorras crear un nodo dentro y reservarle memoria porque ya lo hiciste en el main. Lo nuevo que habría que añadir a la función es una comprobación en el control de errores para comprobar si el nodo destino existe, además de que esta función nos devuelve un STATUS de ERROR u OK en caso de que falla.

Dentro de la función `Node_copy(const Node * pSource, Node * pDest)` se copiaría los nodos, su nombre y su id, por ejemplo `pDest->id = pSource->id` además se debe observar que el const solo se aplica al Source ya que este no se modificara.

Respecto a la `Node_copy(const Node * pSource, Node ** pDest)` sera igual que el caso anterior solo que esta vez pDest es un puntero que apunta a otro puntero por lo que al momento de copiar los campos de los nodos deberemos acceder al contenido del puntero pDest, por ejemplo `(*pDest)->id = pSource->id`.

4. Indica qué se tendría que cambiar en `graph.c/h` para tener grafos que pudieran almacenar cualquier estructura de datos, es decir, que no estuviera limitado a almacenar nodos definidos como el TAD Node.

R.4 Una solución sería crear un TAD llamado elemento el cual se vaya modificando con los tipos de datos que queramos usar, este se deberá poner como cabecera en el graph.h y graph.c.