**UNIDAD TEMÁTICA 7 – GRAFOS DIRIGIDOS– Trabajo de Aplicación 6**

# Ejercicio 1 (total 30 minutos)

1. Desarrollar en lenguaje natural y seudocódigo de alto nivel un algoritmo sencillo que permita determinar si un grafo contiene ciclos (15 minutos)

**Boolean TVertice.tieneCiclo(unCamino);**

Lenguaje Natural

Recorro en bpf los nodos del grafo, si el adyacente ya está visitado y el vértice está en unCamino, entonces el grafo tiene un ciclo. Si tiene retorna true sino false.

Seudocodigo

Boolean tieneCiclo(unCamino)

visitar

agregar al camino

para cada adyacente w

if w está visitado y está en UnCamino

unCamino.agregar(w);

imprimir unCamino;

retornarTrue;

sino

sigo iterando

finParaCada

retornar false;

**Boolean TGrafoDirigido.tieneCiclo();**

LenguajeNatural

si es nulo retorno false, si el número de vértices es menor a 3 tambien retorno false

sino, llamo a tiene Ciclo de Vértice y retorno su resultado.

2. Dado el código de TDA GrafoDirigido ya implementado y las clases auxiliares distribuidas, crear un método sencillo para determinar si el grafo tiene ciclos o no. (implementar tanto el método de Grafo como el de Vértice), respetando las firmas (20 minutos):

* Boolean TGrafoDirigido.tieneCiclo(); // indica si el grafo tiene ciclo y si es así, imprime las etiquetas de los vértices involucrados en el ciclo encontrado.
* Boolean TVertice.tieneCiclo (TCamino unCamino ); Precondiciones: El camino está creado y vacío. Postcondiciones: El camino contendrá todas las etiquetas del ciclo.