



## **Estructuras de Datos**

### **Actividad de Laboratorio 3, Sección C3**

*20 de noviembre de 2023*

#### **Objetivos de la evaluación:**

La siguiente evaluación tiene como objetivo medir los aprendizajes de los estudiantes en los siguientes aspectos:

- Resolver problemas computacionales en lenguaje C usando estructuras de datos no lineales.

#### **Instrucciones:**

Las instrucciones que debe respetar fielmente en esta evaluación son las siguientes:

1. La actividad debe ser desarrollada de manera individual
2. Puede consultar sus apuntes y material personal
3. NO puede usar internet para realizar búsquedas.
4. Solo usar C ANSI
5. Debe utilizar el TDA Grafo (S11\_TDAGrafo\_C3.h) entregado para desarrollar la actividad con las operaciones que se indican.
6. Utilice el archivo S11\_ActLab3\_C3.c y los TDA disponibles en la actividad para responder esta evaluación y que usted debe renombrar agregando su apellido y nombre para realizar la entrega.
7. Subir la actividad de este laboratorio en buzón en Uvirtual, en archivos nombrados S11\_ActLab3\_C3\_APELLIDO\_NOMBRE.c y S11\_TDAGrafo\_C3\_APELLIDO\_NOMBRE.h
8. Plazo: Hoy dentro del horario de laboratorio, en el buzón correspondiente a su sección
9. No está permitido usar variables globales.

#### **ENUNCIADO:**

Descargue los archivos S11\_ActLab3\_C3.c, los TDA y grafo1.in de Uvirtual para desarrollar la actividad. Considere lo siguiente:

S11\_TDAGrafo\_C3.h contiene la estructura de datos y operaciones elementales del TDA Grafo que utilizará para desarrollar su actividad. La definición de la estructura de este TDA no debe ser modificada. Recuerde que debe renombrarlo especificando apellido y nombre para realizar la entrega en la Uvirtual

S11\_ActLab3\_C3.c es el código fuente donde usted debe responder su actividad. Recuerde que debe renombrarlo especificando apellido y nombre para realizar la entrega en la Uvirtual. Lea los problemas que debe resolver (descritos a continuación) y responda cada uno respetando a cabalidad las indicaciones.



Se solicita resolver los siguientes problemas:

### Problema 1:

- a. implementar la función “int **GradodeVertice**(TDAGrafoC3\* grafo, int vertice)” que devuelve un entero que corresponde al grado del vértice, si el vértice no existe, debe retornar un -1
- b. Usando la función anterior implementar “int **GradoPromedio**(TDAGrafoC3\* grafo)” devuelve un número que corresponde al grado promedio del grafo

Probar ambas funcionalidades en el main()

### Problema 2:

- a. Construir una función C que realice el recorrido en amplitud (anchura)  
int \*bfs(TDAgradoC2 \*grafo, int vertice)  
retorna un arreglo de enteros de largo n, donde en cada posición i contiene un 1 si el nodo fue visitado, o un 0 en caso contrario.
- b. int **esConexo**(TDAGrafo\* grafo)  
retorna un 1 si grafo es conexo, 0 en caso contrario

Realizar un recorrido en amplitud sobre el grafo1.in (paso del nombre del archivo por línea de comando)

Modificar el archivo grafo1.in para que no sea conexo (grafo2.in).

En el main aplicar la función **esConexo** a ambos grafos y mostrar el resultado.

### Problema 3:

Usando como base la implementación de recorrido en amplitud (anchura) realizada, modificarla y construir un función en C que calcule la **distancia** desde un nodo i a un nodo j, “int **distancia**(TDAGrafoC3 grafo, int i, int j)”, retorna -1 si no se puede llegar de i a j;