



Universidad Autónoma de Zacatecas

Unidad Académica de Ingeniería Eléctrica

Programa Académico de Ingeniería de Software

Práctica 31

Datos generales:

Nombre de la Práctica	Grafos con listas de adyacencia
Nombre de la carrera	Ingeniería de Software
Nombre de la materia	Laboratorio de Estructuras de Datos
Número y nombre de Unidad(es) temática(s)	IV. Estructuras no lineales.
Docente que imparte la materia	Aldonso Becerra Sánchez
Fecha de entrega para los alumnos	23-noviembre-2021
Fecha de entrega con penalización	24-noviembre-2021
Fecha de elaboración	22-noviembre-2021

Objetivo de la Práctica	Practicar con las operaciones sobre grafos que emplean listas de adyacencia.
Tiempo aproximado de realización	2 horas
Introducción	Los grafos son útiles en el manejo de mapas y planos donde se defina un amplia gama de posibilidades de conexión entre diferentes entidades con el fin de estar comunicadas o por el hecho de ser dependientes unas de otras.

Referencias que debe consultar el alumno (si se requieren):

Referencia 1:

1.Cairo, Osvaldo; Guardati, Silvia. Estructura de Datos, Tercera Edición. McGraw-Hill, México, Tercera Edición, 2006.

Referencia 2:

2.Mark Allen Weiss. Estructura de datos en Java. Ed. Addison Wesley.



Universidad Autónoma de Zacatecas

Unidad Académica de Ingeniería Eléctrica

Programa Académico de Ingeniería de Software

Referencia 3:

3. Joyanes Aguilar, Luis. Fundamentos de Programación. Algoritmos y Estructuras de Datos. Tercera Edición, 2003. McGraw – Hill.

Actividades que debe realizar el alumno:

Actividad inicial:

Generar el reporte en formato IDC.

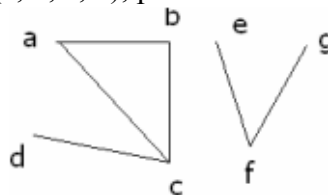
Actividad 1:

Primero genere la **Introducción**.

Actividad 2:

Usando la clase GrafoListaAdyacencia:

- Determine los componentes conexos de un grafo. Cada componente conexo de un grafo es un subgrafo que no pudo recorrerse por completo usando un solo recorrido. De tal forma que tiene que recorrerse nuevamente, pero ahora comenzando desde los vértices que no fueron visitados en los recorridos anteriores; de esta manera se repite el procedimiento hasta terminar de visitar todos los nodos. Puede apoyarse en el procedimiento explicado en clase que usa los recorridos (profundidad o anchura).
- a) Por ejemplo, aquí habría dos componentes conexos (dado que si recorremos el grafo una sola vez, no se pueden visitar todos los vértices al no haber enlaces con los demás, o porque las flechas no permitan llegar a los otros nodos): partiendo de a (a, b, c, d), partiendo de e (e, f, g):



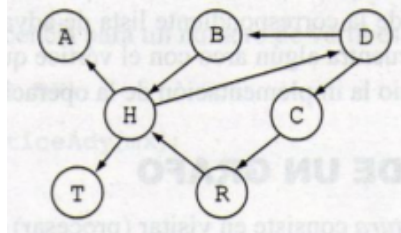


Universidad Autónoma de Zacatecas

Unidad Académica de Ingeniería Eléctrica

Programa Académico de Ingeniería de Software

- b) Por ejemplo, aquí habría dos componentes conexos: partiendo de A (A), partiendo de D (D, B, C, H, R, T)



Complete la etapa de **desarrollo** del reporte con base a esta actividad.

Actividad 3:

Pruebe el funcionamiento del programa de las actividades con todo y sus capturas de pantalla.

Actividad 4:

Realice la sección de **Código agregado** (diagrama de clases UML).

Actividad 5:

Realice la sección de **Pre-evaluación** (use los lineamientos establecidos).

Actividad 6:

Finalmente haga las **Conclusiones**.

Actividad 7:

Enviar en <http://ingsoftware.reduaz.mx/moodle>

Archivo anexo que se requiere para esta tarea (opcional):

Dudas o comentarios: a7donso@gmail.com