

Facultad: Ingeniería  
Escuela: Computación  
Asignatura: Programación IV

Tema: Recorrido de Grafos. Aplicaciones de Grafos.

### Objetivos Específicos

- A partir de un ejemplo de grafos identificar el recorrido en profundidad y el recorrido en anchura y aplicaciones de grafos.
- Implementar el recorrido en profundidad en grafos utilizando Visual C# .NET.
- Implementar el recorrido en anchura en grafos utilizando Visual C# .NET.
- Resolver problemas propuestos, a través de la implementación de soluciones haciendo uso de la estructura de datos grafos.
- Implementar las soluciones utilizando Visual C#.NET.

### Materiales y Equipo

- Guía Número 9.
- Computadora con programa Microsoft Visual C#.

### Introducción Teórica

#### **Recorrido de Grafos.**

La operación de recorrer una estructura de datos consiste en visitar (procesar) cada uno de los nodos a partir de uno dado. Así, para recorrer un árbol se parte del nodo raíz y según el orden se visitan todos los nodos. De igual forma, recorrer un grafo consiste en visitar todos los vértices alcanzables a partir de uno dado.

Hay dos formas de recorrer un grafo: recorrido en profundidad y recorrido en anchura.

Si el conjunto de nodos marcados se trata como una **cola**, entonces el recorrido es en anchura; si se trata como una **pila**, el recorrido es en profundidad.

#### **Recorrido en Anchura.**

El recorrido de búsqueda en anchura, en amplitud o expansión, es una estrategia aplicable indistintamente al caso de grafos dirigidos y no dirigidos.

El recorrido en anchura es una generalización del recorrido por niveles de un árbol.

Se trata de visitar un nodo inicial y luego a todos los nodos que están a un arco de distancia de éste, luego a todos los nodos que están a dos arcos de distancia de éste y así sucesivamente, hasta alcanzar a todos los nodos a los que se pueda llegar desde el nodo inicial.

Una aplicación típica de este recorrido es la **resolución de problemas de planificación**.

La búsqueda en amplitud se puede utilizar para hallar la distancia más corta entre algún nodo inicial y los nodos restantes del grafo. Esta distancia más corta es el mínimo número de aristas que hay que recorrer para pasar desde el nodo inicial hasta el nodo concreto que se esté examinando.

Comenzando en el nodo "s", esta distancia se calcula examinando todas las aristas incidentes en el nodo "s", y pasando después a un nodo adyacente "w", repitiéndose entonces todo el proceso. El recorrido continúa hasta que se hayan examinado todos los nodos del grafo.

En una búsqueda en amplitud, cada nodo se visita o es procesado en algún sentido, dependiendo de la aplicación concreta. La búsqueda comienza en un nodo concreto del grafo. A continuación la búsqueda se extiende a los nodos del grafo que estén más próximos al nodo inicial antes de visitar ningún otro. Estos nodos están relacionados de alguna forma con el nodo especificado, y forman parte de un grupo que depende de la aplicación concreta.

Inicialmente, se visitan todos aquellos nodos que sean adyacentes al nodo inicial. A continuación se visitan todos los nodos que estén a una distancia "2" del nodo inicial. Este proceso se repite hasta que se haya visitado todos los nodos posibles.

Sin embargo, este enfoque de búsqueda puede dar lugar a problemas. Los nodos ya visitados no deben visitarse de nuevo. Se puede evitar esta situación marcando aquellos nodos que se hayan visitado. Si un nodo ya ha sido marcado con anterioridad (esto es, si ya ha sido alcanzado o visitado), nunca vuelve a ser visitado.

Este método utiliza una cola como estructura auxiliar en la que se mantienen los vértices que se vayan a procesar posteriormente.

El recorrido en anchura trabaja de la siguiente manera:

1. Se visita el nodo inicial.
2. Después de visitar el nodo inicial se visitan todos los sucesores.
3. Después los sucesores de los sucesores.
4. Se repiten los pasos 1 y 2 hasta encontrar un nodo sin arcos salientes o ya visitados.
5. Si después de visitar todos los descendientes del primer nodo, todavía quedan más nodos por visitar, se repite el proceso reiniciando.

Este tipo de recorrido es muy usado para problemas de planificación (problema de laberinto).

Al igual que el recorrido en profundidad, en el recorrido en anchura no existe un único recorrido de grafo, sino un conjunto de ellos.

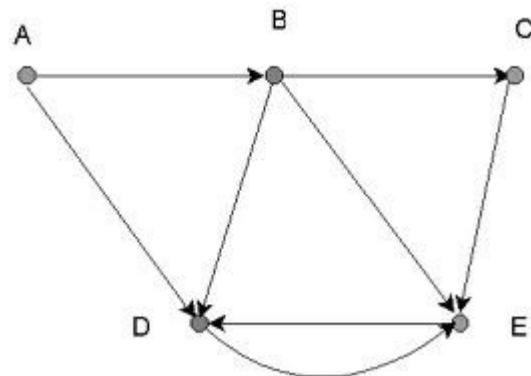
Para el siguiente grafo, habría varios recorridos en anchura posibles, entre los cuales podríamos mencionar:

Visitando primero el nodo izquierdo sería:

$A \rightarrow B \rightarrow D \rightarrow C \rightarrow E$

Visitando primero el derecho sería:

$A \rightarrow D \rightarrow B \rightarrow E \rightarrow C$



### Recorrido en Profundidad.

Este método es una forma básica de recorrer un grafo implementando recursividad.

En este recorrido se basan los recorridos preorden y postorden para árboles binarios.

Supóngase que una persona se encuentra en algún sistema de cuevas interconectadas, o en un laberinto, en una cierta intersección (o nodo), y que se le pide a esta persona que busque la salida, que se encuentra en un determinado nodo. En esta búsqueda se podrían emplear varias opciones.

Una posibilidad que probablemente no se utilizase sería la búsqueda en amplitud. Intuitivamente, una estrategia que comienza en algún nodo de la cueva y que después visita todos y cada uno de los nodos adyacentes siguientes de la cueva, no parece demasiado prometedora.

Una estrategia de aspecto mucho mejor es la que consiste en comenzar en un cierto punto y seguir, digamos, la cueva situada más a la derecha desde esa cueva hasta la próxima intersección. Al llegar a esa intersección, se sigue de nuevo la salida situada más a la derecha, y así sucesivamente hasta llegar al nodo deseado, si este proceso de seguimiento del camino situado más a la derecha no tiene éxito, porque no se hallan nuevas intersecciones, entonces el individuo deshace el camino andado hasta la última intersección, y sigue por la salida contigua a la situada más a la derecha, y así sucesivamente.

El retroceso podría muy bien devolver a la persona al nodo de partida, y entonces se sigue el proceso con la próxima salida de las situadas a la derecha. Los movimientos hacia delante, que van seguidos en ocasiones por retrocesos, son en esencia un método de búsqueda en profundidad que se estudiará a continuación con más detalle.

Así se puede utilizar una búsqueda en profundidad en un grafo arbitrario para realizar el recorrido de un grafo general. A medida que se encuentra cada nuevo nodo, se marca el mismo para evitar volver a visitarlo de nuevo.

La estrategia de recorrido en profundidad es la siguiente:

1. Se toma un nodo "s" como comienzo, y se marca.
2. A continuación se toma y se marca un nodo no marcado adyacente a "s", y ese nodo pasa a ser el nuevo nodo de partida, dejando posiblemente por el momento al nodo inicial original con nodos no explorados.
3. La búsqueda continúa por el grafo hasta que el camino en curso finalice con un grafo de salida igual a cero, o bien en un nodo en que todos los nodos adyacentes estén marcados.
4. A continuación la búsqueda vuelve al último nodo que todavía tenga nodos adyacentes sin marcar, y continúa marcando todos los nodos de forma recursiva hasta que ya no queden nodos sin marcar.

Es preciso marcar los nodos en la búsqueda en profundidad. Si no se hiciera esto, los grafos que contuvieran ciclos darían lugar a bucles infinitos.

Un algoritmo de búsqueda o recorrido en profundidad consta de una única rutina principal que invoca a un procedimiento recursivo de la manera siguiente:

#### *Principal*

1. Se marcan todos los nodos del grafo como no visitados.

2. Se invoca a `recorrido_en_profundidad` (s) para algún nodo inicial "s".

#### *Procedimiento recorrido\_en\_profundidad*

1. Se marca y visita "s".

2. Para cada vecino "w" de "s"

Si el vecino "w" no está marcado

Se invoca a `recorrido_en_profundidad` (w).

El recorrido en profundidad persigue el mismo objetivo que el recorrido en anchura: visitar todos los vértices del grafo alcanzables desde un vértice dado.

La búsqueda en profundidad empieza por un vértice "V" del grafo "G"; "V" se marca como visitado.

Después se recorre en profundidad cada vértice adyacente a "V" no visitado; así hasta que no haya más vértices adyacentes no visitados.

Esta técnica se denomina en profundidad porque la dirección de visitar es hacia adelante mientras que sea posible; al contrario que la búsqueda en anchura, que primero visita todos los vértices posibles en amplitud.

La definición recursiva del recorrido en profundidad nos indica que tenemos que utilizar una "pila" como estructura para guardar los vértices adyacentes no visitados a uno dado. De igual forma que en el recorrido en anchura, hacemos uso de una lista de vértices para controlar los ya visitados.

Los usos que pueden darse a este tipo de recorrido son:

- ✚ Simular una red, a partir de un grafo y analizar su robustez para transferencia de información.
- ✚ Identificar bucles, los cuales son potenciales causantes de elevación de tráfico en recorridos.

Hay que tomar en cuenta que, para recorridos primero en profundidad, no existe un único recorrido de grafo, sino un conjunto de ellos.

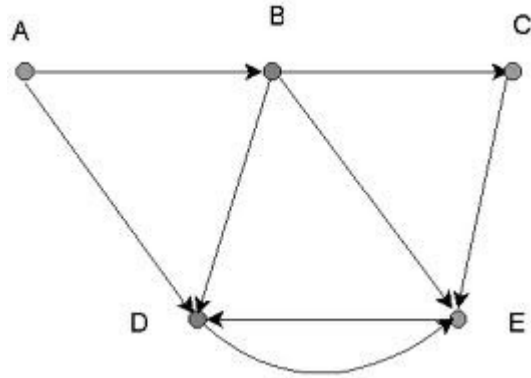
Para el siguiente grafo, habría varios recorridos en profundidad posibles, entre los cuales podríamos mencionar:

Visitando primero el nodo izquierdo sería:

$A \rightarrow D \rightarrow E \rightarrow B \rightarrow C$

Visitando primero el nodo derecho sería:

$A \rightarrow B \rightarrow C \rightarrow E \rightarrow D$



### Aplicaciones de Grafos.

Prácticamente cualquier red puede ser modelada con un grafo: una red de carreteras que conecta ciudades, una red eléctrica o un sistema de alcantarillados, una red de computadoras.

Al visitar una página web y hacer clic a un enlace, visto como un grafo los vértices son los sitios, y cuyas aristas son lógicamente los enlaces.

Gracias a la teoría de Grafos se pueden resolver diversos problemas como por ejemplo la síntesis de circuitos secuenciales.

Los grafos se utilizan también para modelar trayectos como el de una línea de autobús a través de las calles de una ciudad, en el que podemos obtener caminos óptimos para el trayecto aplicando diversos algoritmos como puede ser el algoritmo de Floyd.

Para la administración de proyectos, utilizamos técnicas como PERT en las que se modelan los proyectos utilizando grafos y optimizando los tiempos para concretar las distintas tareas asociadas al desarrollo de un proyecto en particular.

## Análisis de resultados

### Ejercicio 1.

Para el simulador elaborado en la guía anterior, se le agregará más funcionalidad.

Agregar las opciones de menú siguientes:

- ✚ Recorrido en Anchura. Se debe solicitar el nodo inicial al usuario.
- ✚ Recorrido en Profundidad. Se debe solicitar el nodo inicial al usuario.

## Ejercicio 2.

Desarrollar el método **BuscarVertice** y agregarlo como opción del menú.

- ✚ Parámetro: el vértice a buscar debe ser especificado por el usuario.
- ✚ Retorno: booleano, devolver verdadero en caso de encontrar el vértice, falso en caso contrario.

## Ejercicio 3.

Deben agregarse las siguientes opciones de menú al simulador:

- Desarrollar un método para encontrar la distancia entre dos nodos, a partir de la suma de los pesos de los arcos que conforman el camino entre el nodo origen y destino, basado en el recorrido en anchura. Agregar este método como opción en el menú a la solución que se ha ido elaborando desde la Guía No.8.

## Investigación Complementaria

Para la siguiente semana:

Resolver los ejercicios propuestos a continuación utilizando interfaz gráfica de formulario (Windows Forms), para implementar la simulación del Grafo.

Cada ejercicio debe tener toda la funcionalidad básica del simulador de Grafos que se ha venido construyendo desde la Guía No.8, es decir: crear grafo, insertar nodo, eliminar nodo, insertar arco (con peso asociado), eliminar arco.

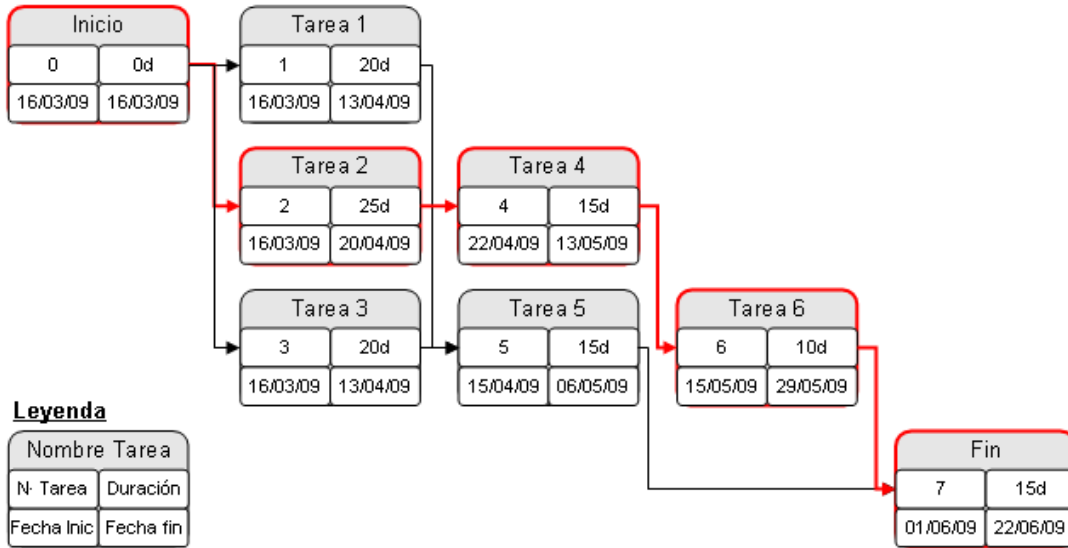
## Ejercicio 1.

Desarrollar una aplicación de grafo para simular gráficos de Pert, considerando las siguientes condiciones:

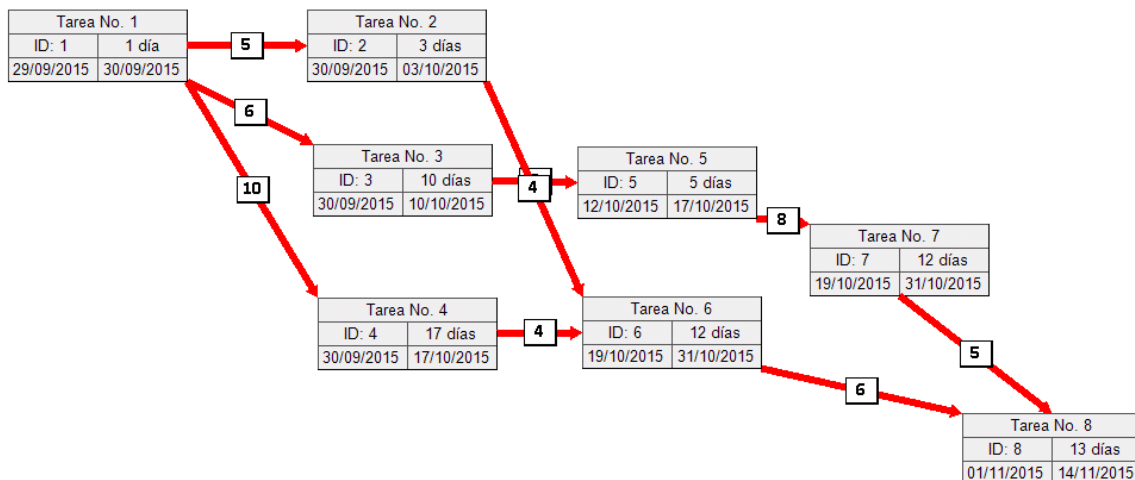
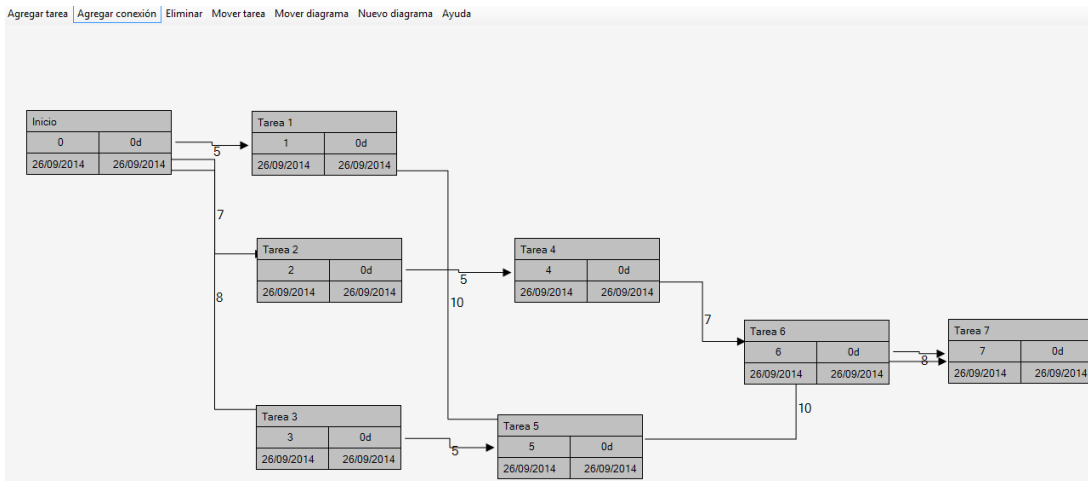
- Cada vértice constará de un número de fase, nombre de la fase (descripción), duración de la fase (en días), fecha inicio, fecha final.
- Las aristas contendrán el costo para llegar a la siguiente fase.

A partir de lo anterior simular el siguiente gráfico de Pert:

## 8 Programación IV. Guía No. 9



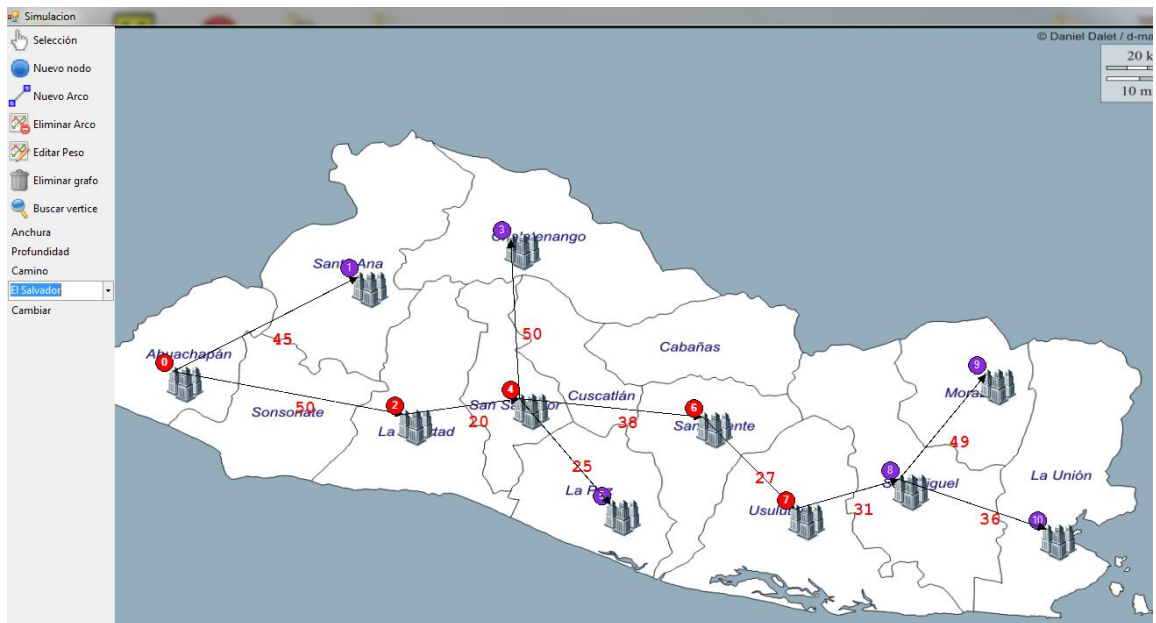
A continuación se muestran ejemplos de cómo podría lucir su aplicación:







10 Programación IV. Guía No. 9



Guía 9: Recorrido de Grafos.  
Aplicaciones de Grafos.

Hoja de cotejo: **9**

Alumno:

Máquina No:

Docente:

GL:

Fecha:

**EVALUACIÓN**

	%	1-4	5-7	8-10	Nota
<b>CONOCIMIENTO</b>	Del 20 al 30%	Conocimiento deficiente de los fundamentos teóricos	Conocimiento y explicación incompleta de los fundamentos teóricos	Conocimiento completo y explicación clara de los fundamentos teóricos	
<b>APLICACIÓN DEL CONOCIMIENTO</b>	Del 40% al 60%				
<b>ACTITUD</b>	Del 15% al 30%	No tiene actitud proactiva.	Actitud propositiva y con propuestas no aplicables al contenido de la guía.	Tiene actitud proactiva y sus propuestas son concretas.	
TOTAL	100%				