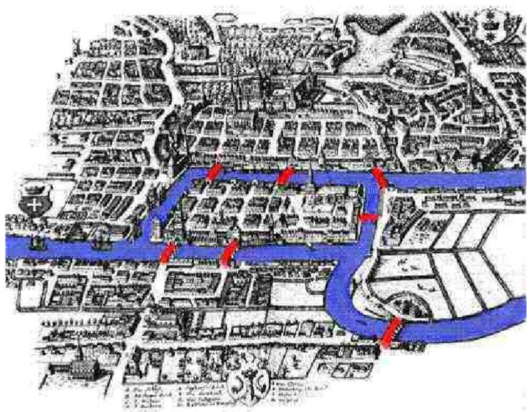
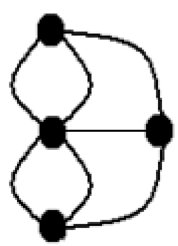
## Grafos

La teoría de grafos fue iniada por Euler. Se cuenta que vivía la ciudad de Königsberg , en Prusia (actualmente, Kalingrado, en Rusia) en la cual habíados pequeñas islas entre las orillas del río Pregel, con siete puentes alrededor, tal como se muestra en la siguiente ilustración:



Se refiere que la gente del lugar intentaba hacer un recorrido que permitiera atravesar cada puente una sola vez regresando al punto de partida. Como nadie lo lograba, este problema atrajo la atención de Euler, quien lo analizó empleando una técnica de graficado por la cual redujo a puntos la representación de las islas y las orillas, y a arcos los siete puentes.
El grafico resulto del siguiente modo. A los puntos se los denomino ‘nodos’ o ‘vértices’. Los enlaces son ‘aristas’ o arcos.

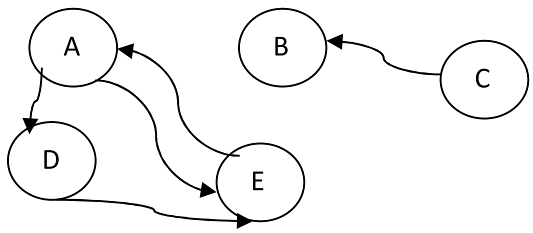


El problema ahora es análogo al de intentar dibujar el grafo anterior partiendo de un vértice, sin levantar el lápiz y sin pasar dos veces por el mismo arco. (A un recorrido con estas características actualmente se lo denomina camino euleriano).Euler demostró que en la condiciones dadas, el probema no tenía solución y además, sus desarrollos dieron origen a la *teoría de Grafos*.

Los grafos constituyen una muy útil herramienta matemática para modelizar situaciones referidas a cuestiones tan diferentes como mapas de interrelación de datos, carreteras, cañerías, circuitos eléctricos, diagrama de dependencia, etc.

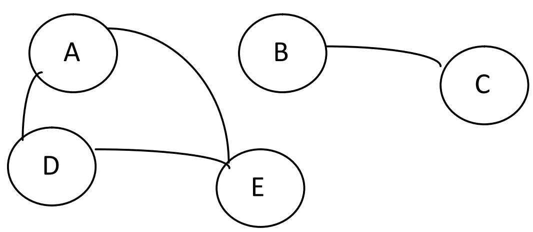
**Definición matematica de Grafo**

Un grafo dirigido o digrafo o grafo orientado consiste en una dupla formada por un conjunto V de vértices o nodos del grafo, y un conjunto de pares ordenados A (aristas orientadas) pertenecientes a VxV.
En símbolos el grafo dirigido G es
G = (V ; A ) donde A es un subconjunto de VxV (Aristas orientadas o dirigidas)
Ejemplo:




V={A, B, C, D, E}
A={ (A,D), (A,E), (E,A), (D,E), (C,B)}

Un grafo no dirigido (o no orientado) es una dupla formada por un conjunto V de vértices o nodos del grafo, y un conjunto de pares NO ordenados A (aristas no orientadas) pertenecientes a VxV. (La relación establecida entre los vértices es simétrica).
En símbolos el grafo dirigido G es
G = (V ; A ) donde A es un conjunto de pares no ordenados de VxV (Aristas NO orientadas o NO dirigidas)
Ejemplo:




V={A, B, C, D, E}
A={ (A,D), (A,E), (D,E), (C,B)} (pares NO ordenados)

En ambos casos, si (v,w) pertenece a A se dice que w es adyacente a v

Algunas definiciones mas:

Camino : serie alternada de vértices y aristas que inicia y finaliza con vértices y donde cada arista conecta el vértice que le precede con el que le sucede

Longitud de Camino: Cantidad de Aristas del camino

Camino Abierto: Camino donde el vértice inicial y final difieren.

Camino Cerrado: Camino donde el vértice inicial y final coinciden.

Recorrido: Camino que no repite aristas

Recorrido Euleriano: Recorrido que contiene todas las aristas del grafo

Circuito: Recorrido Cerrado

Circuito Euleriano: Recorrido Euleriano Cerrado <=> Todos los vértices de grado par

Camino Simple: Camino que no repite vértices (salvo inicial y final).

Camino de Hamilton: Camino Simple que contiene todos los vértices del grafo

Ciclo: Camino Simple Cerrado (o bien , camino que conteniendo al menos tres vértices distintos tales que el primer vértice es adyacente al ultimo )

Ciclo de Hamilton: Camino Hamiltoniano Cerrado

Sub Grafo G’ de G(V,A): es el grafo G’ = (V’, A’) donde V’ es un subconjunto dẹ V y A’ es un subconjunto dẹ A

Grafo no dirigido conexo: un grafo no orientado es conexo sii para todo vértice del grafo hay un camino que lo conecte con otro vértice cualquiera del grafo.

Árbol libre: es un grafo no dirigido conexo sin ciclos.

Grafo dirigido fuertemente conexo: un grafo dirigido es fuertemente conexo sii entre cualquier par de vértices hay un camino que los une.

Grafo subyacente de un grafo: es el grafo no dirigido que se obtiene reemplazando cada arista (orientada) del mismo, por una arista no orientada.

Un grafo dirigido débilmente conexo: es aquel grafo dirigido que no es fuertemente conexo y cuyo grafo subyacente es conexo.

Grado de un vértice: es el número de aristas incidentes en el.

Grado de entrada de un vértice: es el número de aristas del cual el vértice dado es destino.

Grado de entrada de un vértice: es el número de aristas de las cuales el vértice dado es origen.

Vértice fuente: vértice cuyo Grado de Salida es 0 y no es aislado (esto ultimo significa que es primera o segunda componente de al menos un par del conjunto de aristas)

Vértice sumidero: vértice cuyo Grado de Entrada es 0 y no es aislado (esto ultimo significa que es primera o segunda componente de al menos un par del conjunto de aristas)

**El TDA Grafo:**

Es un TDA contenedor de un conjunto de datos llamados nodos y de un conjunto de aristas cada una de las cuales se determina mediante un par de nodos.

Este es un listado de posibles primitivas para el TDA Grafo:

Crear grafo: esta primitiva genera un grafo vacío.
Precondición: ------
Poscondición: grafo generado vacío

Destruir grafo: esta primitiva destruye el grafo.
Precondición: que el grafo exista previamente.
Poscondición: ------

Insertar nodo: esta primitiva inserta un nodo nuevo, recibido como argumento, en el grafo
Precondición: que el grafo exista previamente
Poscondición: el grafo queda modificado por el agregado del nuevo nodo

Insertar arista: esta primitiva inserta una arista nueva, recibida como argumento, en el grafo
Precondición: que el grafo exista previamente y que existan en el grafo los nodos origen y destino de la arista
Poscondición: el grafo queda modificado por el agregado de la nueva arista

Eliminar nodo: esta primitiva elimina un nodo, recibido como argumento, del grafo
Precondición: que el grafo exista previamente y que el nodo a eliminar no tenga aristas incidentes en el
Poscondición: el grafo queda modificado por la eliminación del nodo

Eliminar arista: esta primitiva elimina una arista, recibida como argumento, del grafo
Precondición: que el grafo exista previamente
Poscondición: el grafo queda modificado por la eliminación de la arista
Recorrer grafo: esta primitiva permite visitar cada uno de los nodos del grafo
Precondición: que el grafo exista
Poscondición: ------

Existe arista: esta primitiva recibe una arista y retorna un valor logico indicando si la arista existe en el grafo
Precondición: que el grafo exista
Poscondición: ------

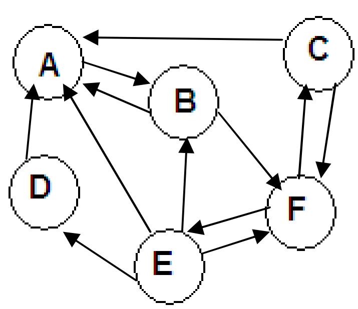
Existe nodo: esta primitiva recibe una arista y retorna un valor logico indicando si el nodo existe en el grafo.
Precondición: que el grafo exista
Poscondición: ------

## Estructuras de datos utilizadas para implementar grafos

Se puede utilizar una Matriz de Adyacencia, la cual, para N nodos es de NxN.
Si M es la matriz de adyacencia del; grafo G entonces verifica que M[i][j] es true (o 1) si la arista (i,j) pertenece al grafo, o false (o bien 0 ) si no pertenece.

Otra posibilidad es usar listas de adyacencia. En este caso, la representación es mediante una lista de listas.
De este modo, se tiene una lista con nodos; cada uno de ellos conteniendo la información sobre un vértice y un puntero a una lista ligada a los vértices adyacentes al vértice indicado.

Ejemplo: para este grafo




La representación de la Matriz de Adyacencia es:

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | 0 | 1 | 0 | 0 | 0 | 0 |
| B | 1 | 0 | 0 | 0 | 0 | 1 |
| C | 1 | 0 | 0 | 0 | 0 | 1 |
| D | 1 | 0 | 0 | 0 | 0 | 0 |
| E | 1 | 1 | 0 | 1 | 0 | 1 |
| F | 0 | 0 | 1 | 0 | 1 | 0 |

La representación con listas de adyacencia se muestra a continuación.
Observar que cada lista de adyacencia (las listas horizontales apuntadas desde el nodo que contiene los datos de un vértice) esta formada por nodos que tienen información sobre la dirección de un nodo adyacente, y el puntero al siguiente nodo de la lista.