	<b>UNIVERSIDAD DE LOS LLANOS</b>		<b>CÓDIGO: FO-DOC-112</b>	
	<b>PROCESO GESTIÓN DE APOYO A LA ACADEMIA</b>		<b>VERSIÓN: 01</b>	<b>PÁGINA: 1 de 4</b>
	<b>FORMATO GUÍA PARA PRÁCTICAS DE LABORATORIO</b>		<b>FECHA: 02/09/2016</b>	
			<b>VIGENCIA: 2016</b>	

**UNIDAD ACADÉMICA:** INGENIERÍA DE SISTEMAS

**CURSO:** OPTIMIZACIÓN

**PRÁCTICA Nº 03:** Teoría de Grafos

## 1. OBJETIVO

- *Analizar el concepto y propiedades fundamentales de los grafos, así como su representación matemática mediante matrices de adyacencia.*
- *Generar la representación gráfica de un grafo, facilitando la visualización de sus componentes y relaciones.*

## 2. CONSULTA PREVIA

*Teoría de grafos, representación mediante matrices de adyacencia, programación en Python, caminos y ciclos en grafos, representación gráfica de grafos, validación de datos, visualización matemática y desarrollo de software con enfoque gráfico.*

## 3. FUNDAMENTO TEÓRICO

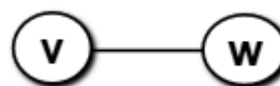
### **Definición y estructura de un grafo**

*La teoría de grafos estudia estructuras matemáticas utilizadas para modelar relaciones entre objetos. Un grafo está compuesto por un conjunto no vacío de vértices y un conjunto de aristas, donde cada arista conecta dos vértices del grafo.*

*Existen dos tipos principales de grafos:*

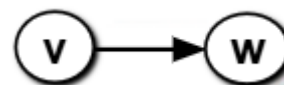
- **Grafos no dirigidos:** Las aristas no tienen un orden, es decir, .

$$(v, w) = (w, v)$$




- **Grafos dirigidos (o digrafos):** Las aristas son pares ordenados, es decir, . En este caso, es la cola y es la cabeza de la arista.

$$\langle v, w \rangle \neq \langle w, v \rangle$$



$$\langle v, w \rangle \Rightarrow w = \text{cabeza de la arista}, v = \text{cola.}$$

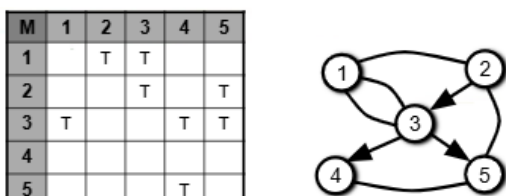
	UNIVERSIDAD DE LOS LLANOS		CÓDIGO: FO-DOC-112
	PROCESO GESTIÓN DE APOYO A LA ACADEMIA		VERSIÓN: 01   PÁGINA: 2 de 4
	FORMATO GUÍA PARA PRÁCTICAS DE LABORATORIO		FECHA: 02/09/2016
			VIGENCIA: 2016

## Terminología fundamental

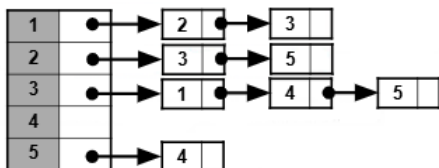
- **Nodos adyacentes a un nodo  $v$**  : Todos los nodos unidos a  $v$  mediante una arista.
  - En grafos dirigidos:
    - Nodos adyacentes a  $v$ : todos los  $w$  con  $\langle v, w \rangle \in A$ .
    - Nodos adyacentes de  $v$ : todos los  $u$  con  $\langle u, v \rangle \in A$ .
- **Camino**: Secuencia de vértices tal que todas las aristas .
- **Longitud de un camino**: Número de aristas en el camino =  $n^\circ$  de nodos - 1.
- **Camino simple**: Camino en el que todos los vértices son distintos, salvo el primero y el último que pueden ser iguales.
- **Ciclo**: Es un camino en el cual el primer y el último vértice son iguales. En grafos no dirigidos las aristas deben ser diferentes.
  - Se llama **ciclo simple** si el camino es simple.
- **Subgrafo**: Grafo cuyos vértices y aristas son subconjuntos del grafo original.
- **Grafo conexo (o conectado)**: Existe un camino entre cualquier par de vértices.
  - Si es un grafo dirigido, se llama **fuertemente conexo**.
- **Grafo completo**: Existe una arista entre cualquier par de vértices.

## Representación de grafos

1. **Matriz de adyacencia**: Una matriz cuadrada donde el elemento es verdadero si existe una arista entre el nodo y el nodo .




2. **Lista de adyacencia**: Para cada nodo, se mantiene una lista de nodos adyacentes.



## Operaciones elementales con grafos

- Creación de grafos vacíos o con un número fijo de vértices.
- Inserción y eliminación de nodos o aristas.
- Consulta de existencia de aristas.
- Iteración sobre aristas adyacentes a un nodo.

	<b>UNIVERSIDAD DE LOS LLANOS</b>		<b>CÓDIGO: FO-DOC-112</b>	
	<b>PROCESO GESTIÓN DE APOYO A LA ACADEMIA</b>		<b>VERSIÓN: 01</b>	<b>PÁGINA: 3 de 4</b>
	<b>FORMATO GUÍA PARA PRÁCTICAS DE LABORATORIO</b>		<b>FECHA: 02/09/2016</b>	
			<b>VIGENCIA: 2016</b>	

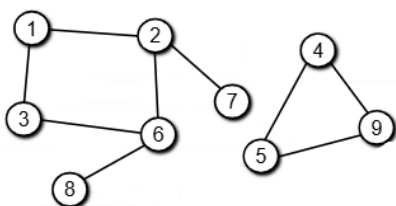
## Recorridos sobre grafos

Los recorridos son una herramienta útil para resolver muchos problemas sobre grafos.

1. **Búsqueda primero en profundidad (DFS):** Explora tan lejos como sea posible a lo largo de cada rama antes de retroceder, equivalente a un recorrido en preorden de un árbol.

El recorrido **no es único**: depende del nodo inicial y del orden de visita de los adyacentes

Ejemplo. Grafo no dirigido.



- Si aparecen varios árboles: **bosque de expansión en profundidad**.
2. **Búsqueda primero en amplitud (BFS):** Visita primero el nodo inicial, luego todos sus nodos adyacentes, después los adyacentes de estos, y así sucesivamente, equivalente a recorrer un árbol por niveles.

El algoritmo utiliza una **cola de vértices**.

- Operaciones básicas:
  - Sacar un nodo de la cola.
  - Añadir a la cola sus adyacentes no visitados.


## Algoritmos fundamentales

1. **Algoritmo de Prim:** Encuentra el árbol de expansión mínima de un grafo ponderado.
2. **Algoritmo de Dijkstra:** Resuelve el problema de caminos mínimos desde un nodo origen a todos los demás nodos en un grafo con pesos positivos.
3. **Problemas NP-completos:** Como el problema del ciclo hamiltoniano y el problema del agente viajero.

## Aplicaciones de los grafos

Los grafos tienen aplicaciones en múltiples áreas, como:

- **Redes de transporte:** Modelado de rutas y conexiones.
- **Redes sociales:** Representación de interacciones entre usuarios.
- **Análisis de circuitos eléctricos:** Donde los nodos representan componentes y las aristas conexiones.
- **Optimización de procesos:** Resolución de problemas de asignación y flujo.

	<b>UNIVERSIDAD DE LOS LLANOS</b>		<b>CÓDIGO: FO-DOC-112</b>	
	<b>PROCESO GESTIÓN DE APOYO A LA ACADEMIA</b>		<b>VERSIÓN: 01</b>	<b>PÁGINA: 4 de 4</b>
	<b>FORMATO GUÍA PARA PRÁCTICAS DE LABORATORIO</b>		<b>FECHA: 02/09/2016</b>	
			<b>VIGENCIA: 2016</b>	

#### 4. EQUIPOS, MATERIALES Y REACTIVOS

Equipos	Materiales	Sustancias y/o Reactivos
Computador		

#### 5. PROCEDIMIENTO O METODOLOGÍA

*Desarrollar un programa en lenguaje de programación Python que permita:*

1. Ingresar la matriz de adyacencia de un grafo específico.
2. Implementar las validaciones correspondientes para asegurar la integridad de los datos ingresados.
3. Generar:
  - La representación matemática del grafo.
  - Su representación gráfica.
  - Evidencias del concepto de:
    - Camino
    - Ciclo
    - Nodos adyacentes

*La entrega debe realizarse a través del aula virtual, incluyendo:*

- Un informe técnico siguiendo el formato IEEE Working Paper, donde se evidencie el proceso de desarrollo de software implementado.
- El código fuente del programa debe estar anexado al informe.

#### 6. RESULTADOS

*Todos los procedimientos y resultados obtenidos durante la práctica deben reflejarse en el documento con formato IEEE Working Paper con su respectivo código fuente. Estos archivos deben ser presentados de forma estructurada y posteriormente subido al aula virtual.*

#### 7. BIBLIOGRAFÍA

- [1]. Diestel, R. (2017). *Graph Theory* (5th ed.). Springer.
- [2]. Gersting, J. L. (2014). *Mathematical Structures for Computer Science* (7th ed.). W.H. Freeman.
- [3]. Hagberg, A. A., Schult, D. A., & Swart, P. J. (2008). *Exploring network structure, dynamics, and function using NetworkX*. In *Proceedings of the 7th Python in Science Conference* (pp. 11-15).
- [4]. Sedgewick, R., & Wayne, K. (2011). *Algorithms* (4th ed.). Addison-Wesley Professional.

ELABORADO POR: Marco Antonio Gutiérrez Cárdenas REVISADO POR:	CARGO: Profesor del curso	FECHA: 05/02/2025
---	---------------------------	-------------------