

# Proyecto 1: Rutas Optimas (Algoritmo de Floyd)

Emily Sanchez  
Viviana Vargas

Curso: Investigación de Operaciones  
II Semestre 2025

11 de septiembre de 2025

## 1. Introducción

El algoritmo de Floyd-Warshall es un algoritmo para encontrar los caminos más cortos en un grafo ponderado. Fue publicado por Robert Floyd en 1962.

El algoritmo de Floyd se basa en el principio de la Programación Dinámica.

**Complejidad espacial:**  $O(n^2)$

**Complejidad temporal:**  $O(n^3)$

---

## 2. Descripción del Problema

Grafo con 3 nodos:

- Nodo A: A
- Nodo B: B
- Nodo C: C

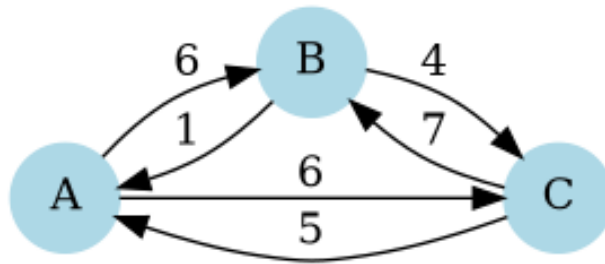


Figura 1: Representación del grafo original

### 3. Procedimiento del Algoritmo

#### 3.1. Matriz de Distancias Inicial $D(0)$

	A	B	C
A	0	6	6
B	1	0	4
C	5	7	0

Cuadro 1: Matriz de distancias inicial  $D(0)$

### 3.2. Matriz de Caminos Inicial $P(0)$

	A	B	C
A	-	A	A
B	B	-	B
C	C	C	-

Cuadro 2: Matriz de caminos inicial  $P(0)$

### 3.3. Iteraciones del Algoritmo

**3.3.1. Iteración 1 ( $k = 1$ ) - Nodo intermedio: A**

	A	B	C
A	0	6	6
B	1	0	4
C	5	7	0

Cuadro 3: Matriz de distancias  $D(1)$  - Cambios resaltados en verde

	A	B	C
A	-	A	A
B	B	-	B
C	C	C	-

Cuadro 4: Matriz de caminos  $P(1)$  - Cambios resaltados en azul

**3.3.2. Iteración 2 ( $k = 2$ ) - Nodo intermedio: B**

	A	B	C
A	0	6	6
B	1	0	4
C	5	7	0

Cuadro 5: Matriz de distancias  $D(2)$  - Cambios resaltados en verde

	A	B	C
A	-	A	A
B	B	-	B
C	C	C	-

Cuadro 6: Matriz de caminos  $P(2)$  - Cambios resaltados en azul

**3.3.3. Iteración 3 ( $k = 3$ ) - Nodo intermedio: C**

	A	B	C
A	0	6	6
B	1	0	4
C	5	7	0

Cuadro 7: Matriz de distancias  $D(3)$  - Cambios resaltados en verde

	A	B	C
A	-	A	A
B	B	-	B
C	C	C	-

Cuadro 8: Matriz de caminos  $P(3)$  - Cambios resaltados en azul



## 4. Resultados Finales

### 4.1. Matriz de Distancias Final D(3)

	A	B	C
A	0	6	6
B	1	0	4
C	5	7	0

Cuadro 9: Matriz de distancias final D(3)

## 4.2. Matriz de Caminos Final P(3)

	A	B	C
A	-	A	A
B	B	-	B
C	C	C	-

Cuadro 10: Matriz de caminos final P(3)

### 4.3. Rutas Óptimas

- **A → B:** Distancia: 6, Ruta: A → B
  - **A → C:** Distancia: 6, Ruta: A → C
  - **B → A:** Distancia: 1, Ruta: B → A
  - **B → C:** Distancia: 4, Ruta: B → C
  - **C → A:** Distancia: 5, Ruta: C → A
  - **C → B:** Distancia: 7, Ruta: C → B
-