

## UT7-TA2

### ✓ ESCENARIO - Tabla de vuelos

La tabla puede leerse como una **matriz de adyacencias** con distancias (en km):

Origen / Destino	Montevideo	Porto Alegre	San Pablo	Punta del Este
Montevideo	x	300	400	-
Porto Alegre	-	x	200	-
San Pablo	-	-	x	410
Punta del Este	150	390	-	x

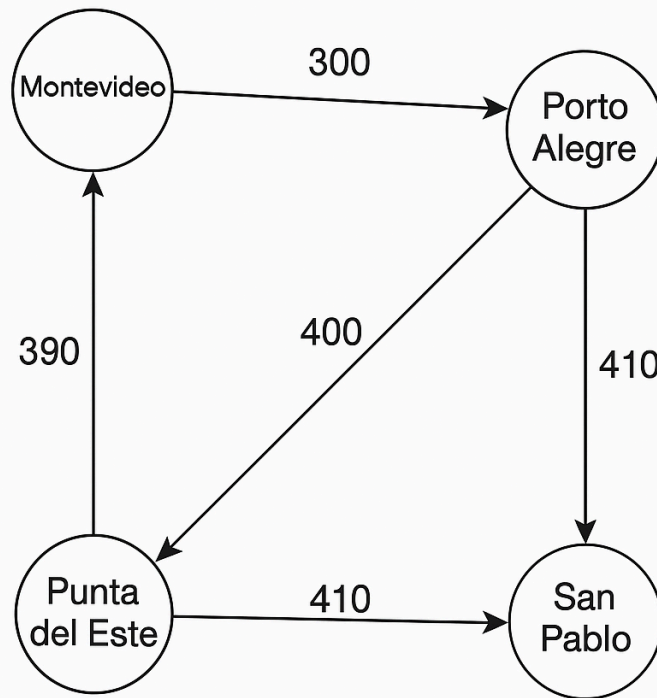
---

### ✓ EJERCICIO 1

#### 1. Dibujar el grafo

Nodos: Montevideo, Porto Alegre, San Pablo, Punta del Este  
Aristas dirigidas con los siguientes pesos:

- Montevideo → Porto Alegre (300)
- Montevideo → San Pablo (400)
- Porto Alegre → San Pablo (200)
- Punta del Este → Montevideo (150)
- Punta del Este → Porto Alegre (390)
- San Pablo → Punta del Este (410)



## 2. Análisis del algoritmo de Floyd-Warshall

### ❖ Complejidad:

- Tiempo:  $O(n^3)$
- Espacio:  $O(n^2)$

Para **4 ciudades**, esto es trivial (64 operaciones), pero:

### ❖ ¿Puede reducirse?

No en su forma básica. Pero:

- Para grafos muy esparcidos, usar **algoritmos más rápidos por demanda** (como Dijkstra desde cada nodo) puede ser más eficiente.
- **Floyd-Warshall** es ideal si necesitamos **todos los caminos más cortos entre todos los pares**, como en este caso.

### ❖ En operaciones reales:

- Para vuelos en todo un continente, el crecimiento cúbico **sí impacta**, aunque sigue siendo viable para decenas o pocos cientos de ciudades.

- Se justifica si el análisis es **previo** a búsquedas y consultas, no en tiempo real.

## ✓ EJERCICIO 2

### 1. Calcular distancias mínimas con Floyd-Warshall

Inicializamos la matriz con las distancias dadas y aplicamos el algoritmo:

📌 **Matriz inicial ( $\infty$  si no hay camino directo):**

	MON	POA	SP	PDE
MON	0	300	400	$\infty$
POA	$\infty$	0	200	$\infty$
SP	$\infty$	$\infty$	0	410
PDE	150	390	$\infty$	0

### 🔄 Aplicamos Floyd-Warshall

Después de completar el algoritmo, la **matriz de distancias mínimas** queda:

	MON	POA	SP	PDE
MON	0	300	400	810
POA	$\infty$	0	200	610
SP	$\infty$	$\infty$	0	410
PDE	150	390	590	0

### 2. ¿Dónde instalar el centro de mantenimiento?

Buscamos la **ciudad con menor suma de distancias mínimas al resto**:

- Montevideo:  $300 + 400 + 810 = 1510$
- Porto Alegre:  $200 + 610 = 810$
- San Pablo:  $410 = 410$
- Punta del Este:  $150 + 390 + 590 = 1130$

👉 **Resultado: San Pablo** tiene el menor costo total de conexión, pero **no tiene acceso entrante desde otras ciudades** (salvo desde Porto Alegre y Montevideo).

Si se busca **centralidad real**, **Porto Alegre** podría ser más adecuada.

---

## ✓ EJERCICIO 3 - Recuperación de caminos

### Matrices auxiliares:

Usamos dos matrices:

- `dist[i][j]`: distancia mínima de `i` a `j`
- `next[i][j]`: siguiente nodo en el camino mínimo de `i` a `j`

Para recuperación de camino entre `i` y `j`:

```
List<Integer> reconstruirCamino(int i, int j) {  
    if (next[i][j] == -1) return Collections.emptyList();  
    List<Integer> camino = new ArrayList<>();  
    camino.add(i);  
    while (i != j) {  
        i = next[i][j];  
        camino.add(i);  
    }  
    return camino;  
}
```

Se puede adaptar para usar nombres de ciudades.

---

## ✓ EJERCICIO 4 - Preguntas posibles

Ejemplos de preguntas tipo quiz proyectadas:

1. ¿Cuál es el camino más corto de Montevideo a San Pablo?
    - Respuesta: Montevideo → San Pablo (400 km)
  2. ¿Cuál es la distancia más corta de Punta del Este a San Pablo?
    - Respuesta: PDE → POA → SP = 390 + 200 = **590 km**
  3. ¿Cuál es la ciudad mejor conectada con el resto?
    - Respuesta discutible: **Porto Alegre** o **Punta del Este** (según criterios)
  4. ¿Cuál es la distancia mínima de Montevideo a Punta del Este?
    - Respuesta: 810 km (vía San Pablo)
-