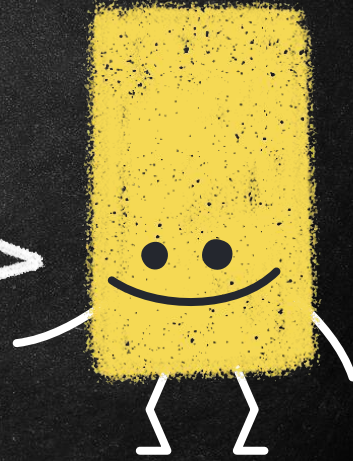


DANIEL  
PEDROSA  
MONTES



# PROBLEMA DE LA MÍNIMA DISPERSIÓN DIFERENCIAL

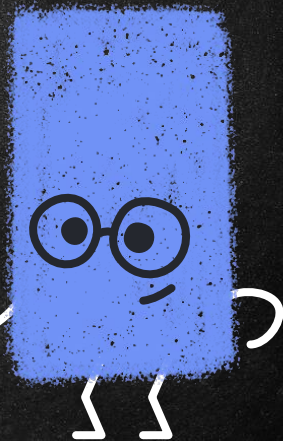
PALOMA  
CANDOCIA  
GONZÁLEZ





# ÍNDICE CHUPIGUAY

- ¿Qué es?
- Aplicaciones
- Paralelizar un algoritmo y encontrar concurrencia
- Representación de los datos de entrada
- Paralelización por datos
- Paralelización por tareas
- Estructuras de datos comunes
- Bibliografía



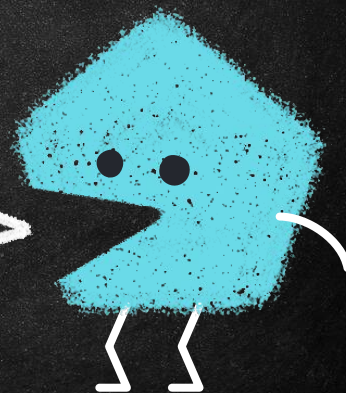
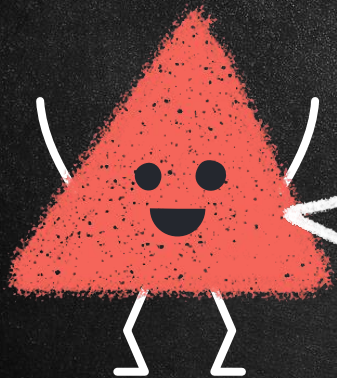


## ¿QUÉ ES?

Un problema de optimización combinatoria NP-Completo.

Consiste en, dado un grupo finito de puntos, escoger un subconjunto de forma que los puntos de este estén más o menos a la misma distancia.

$$\text{Minimizar } \text{Max}_{i \in M} \left\{ \sum_{j \in M} d_{ij} \right\} - \text{Min}_{i \in M} \left\{ \sum_{j \in M} d_{ij} \right\}$$

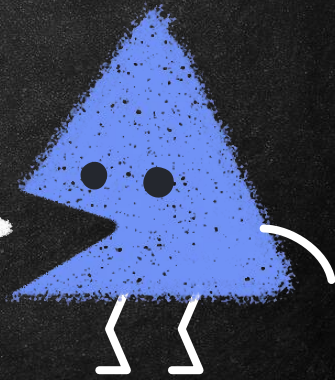
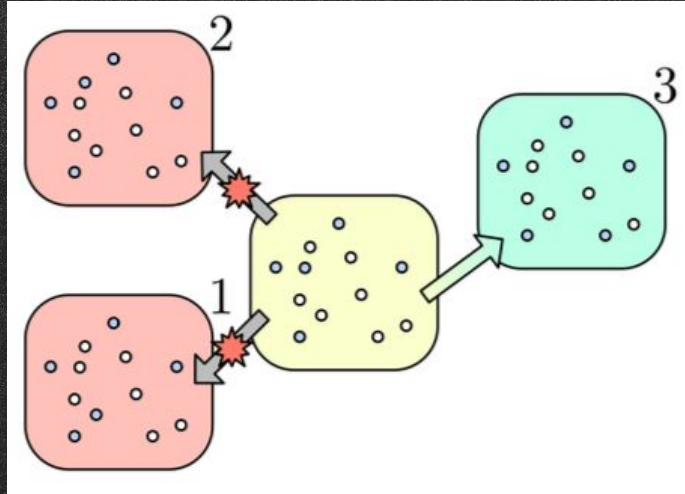




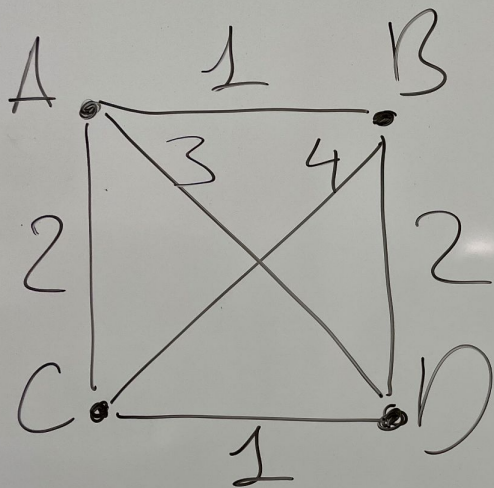
# APLICACIONES

Entre otros,

- Localización de farmacias u hospitales.
- Reparto equitativo en problemas de flujo de red.





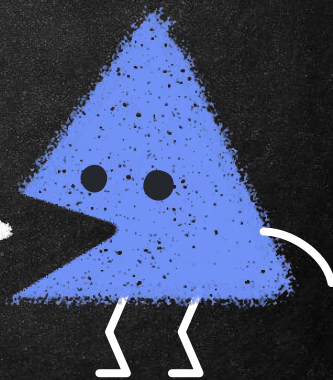


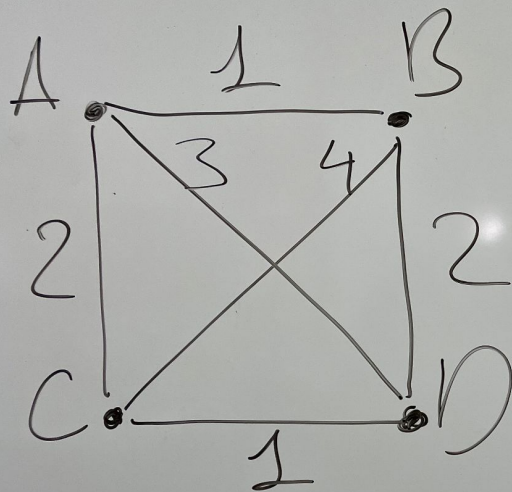
$$\sum_{j \in V} d_{Aj} = 6$$

$$\sum_{j \in V} d_{Bj} = 7$$

$$\sum_{j \in V} d_{Cj} = 7$$

$$\sum_{j \in V} d_{Dj} = 6$$



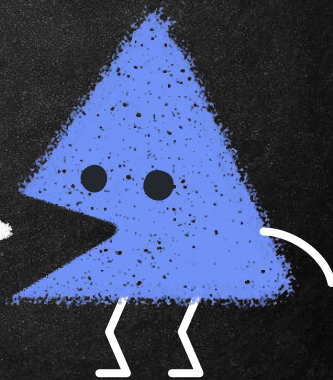


$$\sum_{j \in M} d_{Aj} = 6$$

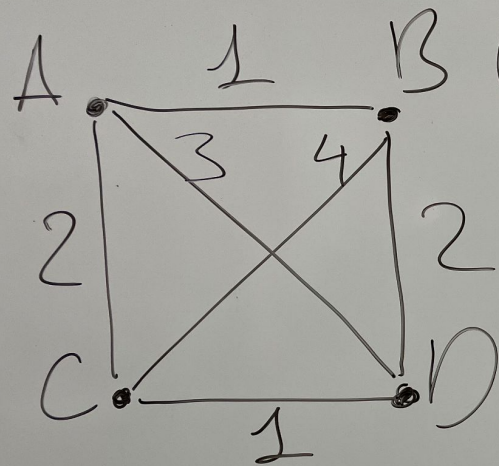
$$\sum_j d_{Bj} = 7 \quad \leftarrow \text{MAX}$$

$$\sum_j d_{Cj} = 7$$

$$\sum_j d_{Dj} = 6$$





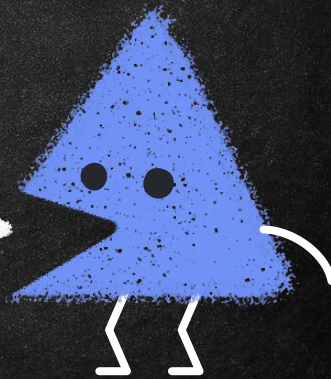


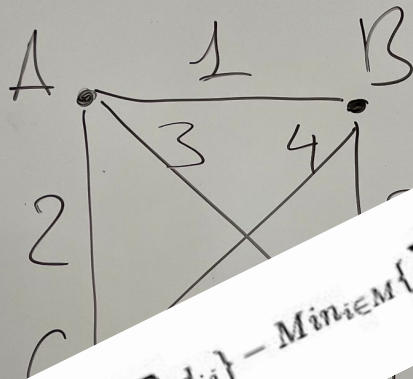
$$\sum_{j \in M} d_{Aj} = 6 \quad \leftarrow \text{MIN}$$

$$\sum_j d_{Bj} = 7 \quad \leftarrow \text{MAX}$$

$$\sum_j d_{Cj} = 7$$

$$\sum_j d_{Dj} = 6$$





$$\sum_{j \in M} d_{Aj} = 6 \quad \leftarrow \text{MIN}$$

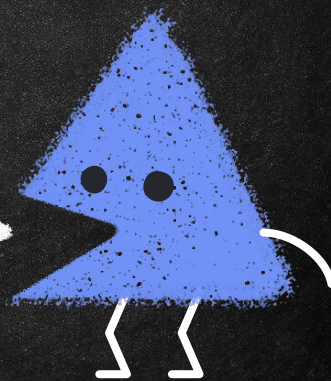
$$\sum_{j \in M} d_{Bj} = 7 \quad \leftarrow \text{MAX}$$

$$\sum_{j \in M} d_{Cj} = 7$$

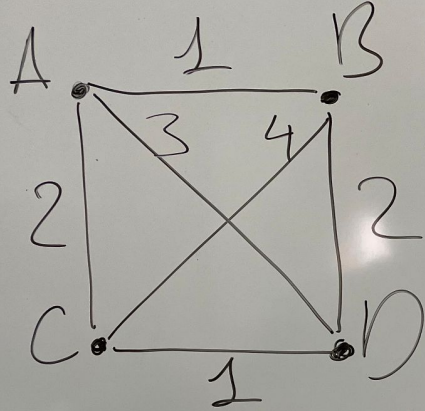
$$\sum_{j \in M} d_{Dj} = 6$$

Minimizar  $\text{Max}_{i \in M} \{ \sum_{j \in M} d_{ij} \} - \text{Min}_{i \in M} \{ \sum_{j \in M} d_{ij} \}$

$$h(M) = 7 - 6 = 1$$







$$\sum_{j \in M} d_{Aj} = 6 \quad \leftarrow \text{MIN}$$

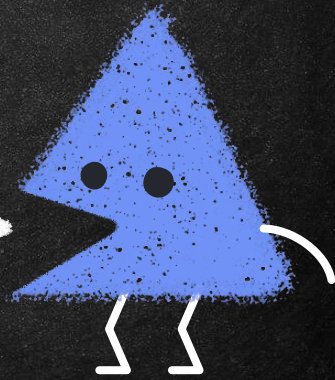
$$\sum_{j \in N} d_{Bj} = 7 \quad \leftarrow \text{MAX}$$

$$\sum_{j \in C} d_{Cj} = 7$$

$$\sum_{j \in D} d_{Dj} = 6$$

$$h(M) = 7 - 6 = 1$$

MINIMIZAR  
 $h(X)$





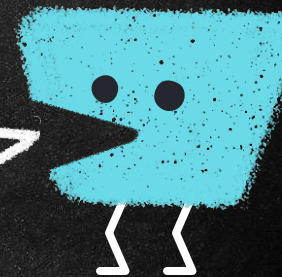
## PARALELIZAR UN ALGORITMO

1. Encontramos la concurrencia.
2. Decidimos su estructura.
3. Damos soporte a las estructuras de datos.
4. Implementamos.



## ENCONTRAR CONCURRENCIA

1. Descomponemos el problema.
2. Análisis de dependencias.
3. Evaluamos el diseño elegido.

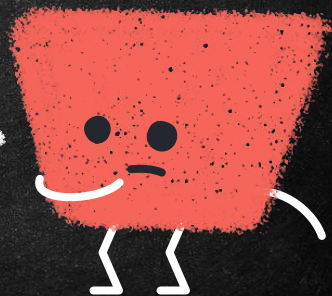




# REPRESENTACIÓN DE LOS DATOS DE ENTRADA

PUNTO A	PUNTO B	DISTANCIA
1	1	0
1	2	4
1	3	7
2	2	0
2	3	5
3	3	0

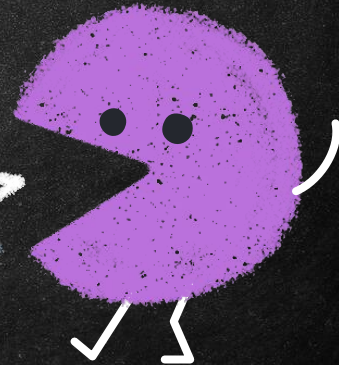
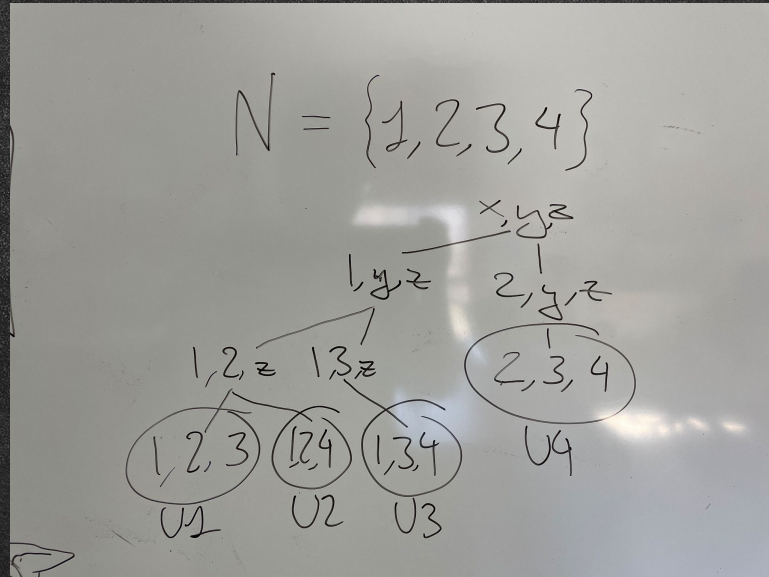
- Cada punto se representa por un índice.
- Se nos proporciona la distancia existente entre puntos.





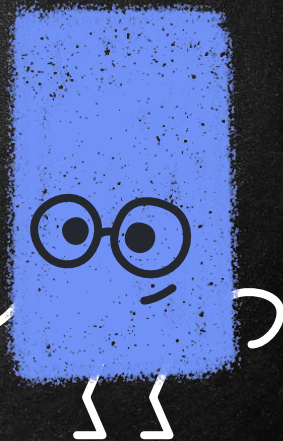
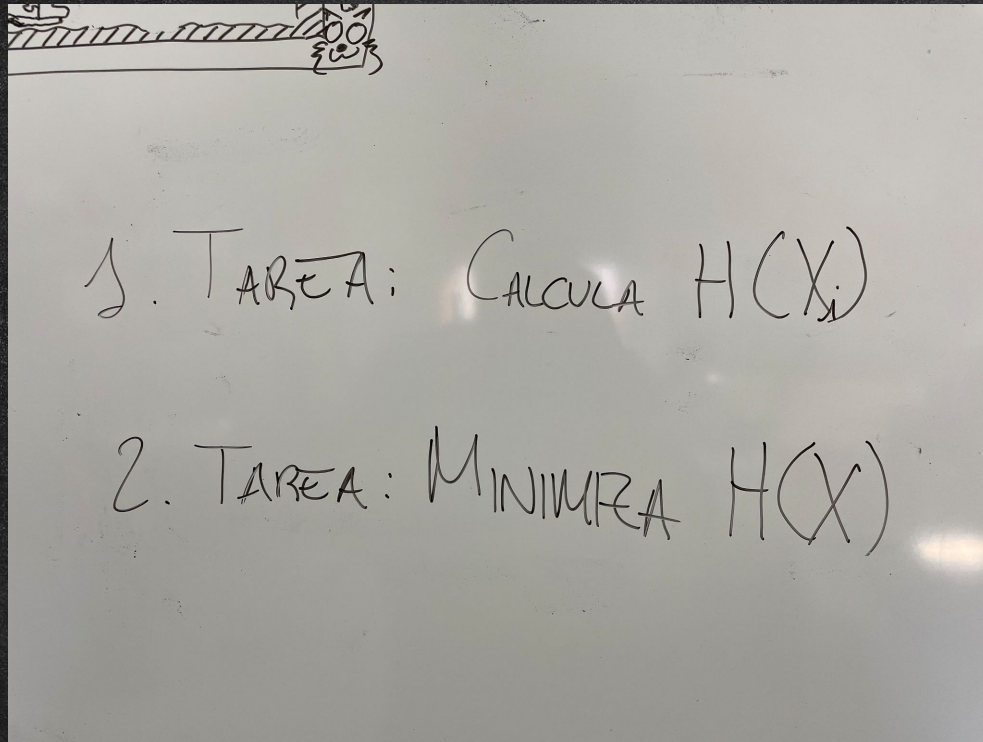
# PARALELIZACIÓN POR DATOS

Podríamos asignar a cada unidad de cómputo un número igual de soluciones a evaluar.





# PARALELIZACIÓN POR TAREAS





# ESTRUCTURAS DE DATOS COMUNES

```
struct solucion {  
    int indices[N];  
    int num_indices;  
}; ← Datos locales
```

```
struct distancias {  
    int d[N][N];  
    int num_indices;  
}; (Read-only)
```

```
struct solucion-evaluada {  
    int num-permutacion;  
    int sum_max;  
    int sum_min;  
}; (Envía el mínimo por  
unidad de cómputo)
```





# BIBLIOGRAFÍA

- Apuntes de la asignatura de Metaheurísticas (21/22)
- Apuntes de la asignatura Arquitectura y Computación de Altas Prestaciones (22/23)

