

Universidad Nacional Autónoma de  
México

FACULTAD DE CIENCIAS

COMPLEJIDAD COMPUTACIONAL

*Práctica 1*  
*No Determinismo*

*Lucy Gasca Soto*  
*Brenda Margarita Becerra Ruiz*

AUTOR

*Delgado Díaz Hermes Alberto*



26 de febrero de 2025



Ejemplo de resultado:

- Cuando no se acepta la solución de los candidatos.

```
Intento 1: Elemento candidato: 3
El elemento 3 no es mayoritario (aparece 1 veces en 46 elementos).
Intento 2: Elemento candidato: 3
El elemento 3 no es mayoritario (aparece 1 veces en 46 elementos).
Intento 3: Elemento candidato: 12
El elemento 12 no es mayoritario (aparece 0 veces en 46 elementos).
Intento 4: Elemento candidato: 4
El elemento 4 no es mayoritario (aparece 2 veces en 46 elementos).
Intento 5: Elemento candidato: 9
El elemento 9 no es mayoritario (aparece 1 veces en 46 elementos).
No se encontró un candidato mayoritario válido en los intentos realizados.
```

- Cuando se acepta la solución.

```
Intento 1: Elemento candidato: 6
El elemento 6 es mayoritario (aparece 25 veces en 46 elementos).
Resultado: 6
```

Comando de ejecución:

Dentro de la carpeta *src*

```
?- Python Problema01.py
```

## Problema 02: 3-Dimensional Matching

Consideremos tres conjuntos:  $B$ ,  $G$  y  $H$  cada conjunto tiene  $n$  elementos y existe una relación  $T \subseteq B \times G \times H$ .

**Problema 3DM:** Encontrar el conjunto de  $n$  tripleteas, de tal manera que ningún par de ellas tenga componentes en común.

### Versión como problema de decisión

Dados tres conjuntos  $B$ ,  $G$  y  $H$ , cada conjunto tiene  $n$  elementos y existe una relación  $T \subseteq B \times G \times H$ , ¿existe un subconjunto  $M \subseteq T$  de  $n$  triplas tal que ninguna pareja de triplas en  $M$  comparta algún elemento?

## Algoritmo No Determinista

- Fase Adivinadora.

Se utiliza la primitiva *nd-choice* para 'adivinar' un subconjunto  $M$  de  $T$  formado por  $n$  triplas.

- Fase Verificadora.

Se verifica que  $M$  contenga  $n$  triplas. Después se comprueba que, para cada par de triplas en  $M$ , no exista ningún elemento en común.

## Programa

Dentro del código se encuentra un ejemplo de ejecución, donde hace 5 intentos de candidatos para saber si el candidato es el subconjunto  $M$  de  $n$  triplas que cumpla con las condiciones.

```
1 B = ['Fernanda', 'Julieta', 'Ana' ]
2 G = ['Juan', 'Alan', 'Gustavo']
3 H = ['Casa1', 'Casa2', 'Casa3']
4 T = [
5     ('Fernanda', 'Juan', 'Casa1'),
6     ('Fernanda', 'Juan', 'Casa2'),
7     ('Fernanda', 'Juan', 'Casa3'),
8     ('Fernanda', 'Alan', 'Casa1'),
9     ('Fernanda', 'Alan', 'Casa2'),
10    ('Fernanda', 'Alan', 'Casa3'),
11    ('Fernanda', 'Gustavo', 'Casa1'),
12    ('Fernanda', 'Gustavo', 'Casa2'),
13    ('Fernanda', 'Gustavo', 'Casa3'),
14    ('Julieta', 'Juan', 'Casa1'),
15    ('Julieta', 'Juan', 'Casa2'),
16    ('Julieta', 'Juan', 'Casa3'),
17    ('Julieta', 'Alan', 'Casa1'),
18    ('Julieta', 'Alan', 'Casa2'),
19    ('Julieta', 'Alan', 'Casa3'),
20    ('Julieta', 'Gustavo', 'Casa1'),
21    ('Julieta', 'Gustavo', 'Casa2'),
22    ('Julieta', 'Gustavo', 'Casa3'),
23    ('Ana', 'Juan', 'Casa1'),
24    ('Ana', 'Juan', 'Casa2'),
25    ('Ana', 'Juan', 'Casa3'),
26    ('Ana', 'Alan', 'Casa1'),
27    ('Ana', 'Alan', 'Casa2'),
28    ('Ana', 'Alan', 'Casa3'),
29    ('Ana', 'Gustavo', 'Casa1'),
30    ('Ana', 'Gustavo', 'Casa2'),
31    ('Ana', 'Gustavo', 'Casa3'),
32
33 ]
```

Ejemplo de resultado:

- Cuando no se acepta la solución de los candidatos.

```
Intento 1: Conjunto candidato (M): [('Ana', 'Alan', 'Casa2'), ('Fernanda', 'Juan', 'Casa3'), ('Ana', 'Juan', 'Casa2')]
La verificación falla: se repite algún elemento en las tripletas.
Intento 2: Conjunto candidato (M): [('Fernanda', 'Juan', 'Casa2'), ('Ana', 'Gustavo', 'Casa1'), ('Fernanda', 'Gustavo', 'Casa1')]
La verificación falla: se repite algún elemento en las tripletas.
Intento 3: Conjunto candidato (M): [('Fernanda', 'Alan', 'Casa2'), ('Fernanda', 'Juan', 'Casa1'), ('Julieta', 'Juan', 'Casa2')]
La verificación falla: se repite algún elemento en las tripletas.
Intento 4: Conjunto candidato (M): [('Julieta', 'Gustavo', 'Casa2'), ('Julieta', 'Juan', 'Casa3'), ('Julieta', 'Alan', 'Casa1')]
La verificación falla: se repite algún elemento en las tripletas.
Intento 5: Conjunto candidato (M): [('Julieta', 'Juan', 'Casa2'), ('Julieta', 'Gustavo', 'Casa1'), ('Julieta', 'Gustavo', 'Casa3')]
La verificación falla: se repite algún elemento en las tripletas.
Intento 6: Conjunto candidato (M): [('Ana', 'Alan', 'Casa3'), ('Julieta', 'Alan', 'Casa2'), ('Ana', 'Gustavo', 'Casa1')]
La verificación falla: se repite algún elemento en las tripletas.
Intento 7: Conjunto candidato (M): [('Ana', 'Alan', 'Casa2'), ('Ana', 'Gustavo', 'Casa1'), ('Julieta', 'Alan', 'Casa1')]
La verificación falla: se repite algún elemento en las tripletas.
Intento 8: Conjunto candidato (M): [('Ana', 'Alan', 'Casa1'), ('Julieta', 'Alan', 'Casa1'), ('Ana', 'Juan', 'Casa3')]
La verificación falla: se repite algún elemento en las tripletas.
Intento 9: Conjunto candidato (M): [('Fernanda', 'Alan', 'Casa3'), ('Ana', 'Alan', 'Casa1'), ('Julieta', 'Gustavo', 'Casa1')]
La verificación falla: se repite algún elemento en las tripletas.
Intento 10: Conjunto candidato (M): [('Fernanda', 'Gustavo', 'Casa1'), ('Ana', 'Alan', 'Casa3'), ('Fernanda', 'Alan', 'Casa3')]
La verificación falla: se repite algún elemento en las tripletas.
No se encontró un conjunto candidato válido en los intentos realizados.
```

- Cuando se acepta la solución.

```
Intento 1: Conjunto candidato (M): [('Ana', 'Gustavo', 'Casa1'), ('Julieta', 'Gustavo', 'Casa2'), ('Julieta', 'Juan', 'Casa3')]
La verificación falla: se repite algún elemento en las tripletas.
Intento 2: Conjunto candidato (M): [('Julieta', 'Alan', 'Casa2'), ('Fernanda', 'Juan', 'Casa1'), ('Ana', 'Juan', 'Casa1')]
La verificación falla: se repite algún elemento en las tripletas.
Intento 3: Conjunto candidato (M): [('Ana', 'Alan', 'Casa3'), ('Julieta', 'Alan', 'Casa2'), ('Fernanda', 'Gustavo', 'Casa1')]
La verificación falla: se repite algún elemento en las tripletas.
Intento 4: Conjunto candidato (M): [('Ana', 'Juan', 'Casa2'), ('Ana', 'Juan', 'Casa1'), ('Fernanda', 'Alan', 'Casa3')]
La verificación falla: se repite algún elemento en las tripletas.
Intento 5: Conjunto candidato (M): [('Fernanda', 'Gustavo', 'Casa1'), ('Fernanda', 'Gustavo', 'Casa2'), ('Fernanda', 'Alan', 'Casa2')]
La verificación falla: se repite algún elemento en las tripletas.
Intento 6: Conjunto candidato (M): [('Fernanda', 'Gustavo', 'Casa2'), ('Ana', 'Alan', 'Casa1'), ('Julieta', 'Juan', 'Casa3')]
La verificación es exitosa. M es un 3DM válido.
Resultado: [('Fernanda', 'Gustavo', 'Casa2'), ('Ana', 'Alan', 'Casa1'), ('Julieta', 'Juan', 'Casa3')]
```

Comando de ejecución:

Dentro de la carpeta *src*

```
?- Python Problema02.py
```

La idea de hacer intentos dentro del código es que haya una alta probabilidad de que al menos una vez seleccionemos al candidato que cumpla las condiciones de los problemas.