

Proyecto 3

Matemática Discreta sección 30



Excelencia que trasciende

DELVALLE
GRUPO EDUCATIVO

Paula de León
Isaías García
Natanael Girón
Jimberth Maldonado
Pablo Méndez
Karen Pineda

Ciudad de Guatemala, Guatemala
Octubre de 2025

Contents

1	Introducción	3
2	Estructura del Programa	3
3	Fundamento Teórico: Técnicas de Conteo	3
3.1	Permutaciones	3
3.2	Combinaciones	4
4	Descripción de la Implementación	4
4.1	Gestión de Conjuntos y Relaciones	4
4.2	Funciones de Conteo	4
4.3	Interfaz Interactiva	4
5	Resultados y Ejemplos	5
5.1	Ejemplo 1: Permutaciones de Objetos Diferentes	5
5.2	Ejemplo 2: Permutaciones de Objetos Iguales	6
5.3	Ejemplo 3: Combinaciones de Objetos Diferentes	7
5.4	Ejemplo 4: Combinaciones de Objetos Iguales	8
6	Detalles Técnicos	8
7	Conclusiones	9
8	Referencias	9

1 Introducción

El documento corresponde al proyecto 3 del curso **Matemática Discreta 1 (MM2015)** de la Universidad del Valle de Guatemala. El objetivo principal es implementar, mediante un programa en Python, las diferentes técnicas de conteo fundamentales, específicamente **permutaciones y combinaciones**, tanto para objetos diferentes como para objetos iguales.

El programa desarrollado en consola permite que el usuario defina conjuntos, cree relaciones y calcule automáticamente todas las combinaciones y permutaciones posibles. Los resultados se muestran paso a paso y se explican con las fórmulas utilizadas, lo que facilita la comprensión del tema.

2 Estructura del Programa

El programa está diseñado de forma modular e interactiva, integrando los temas previos de conjuntos y relaciones con las nuevas funcionalidades de conteo. Desde el menú principal, el usuario puede acceder a las siguientes opciones:

1. Definir conjunto referencial (universo)
2. Crear nuevos conjuntos o relaciones
3. Ejecutar operaciones de conjuntos (unión, intersección, diferencia, complemento)
4. Calcular producto cartesiano
5. Verificar propiedades de relaciones (reflexiva, simétrica, transitiva)
6. Calcular permutaciones y combinaciones:
 - Permutaciones de objetos diferentes
 - Permutaciones de objetos iguales
 - Combinaciones de objetos diferentes
 - Combinaciones de objetos iguales

3 Fundamento Teórico: Técnicas de Conteo

3.1 Permutaciones

El programa implementa dos tipos principales de permutaciones:

- **Permutaciones de objetos diferentes:**

$$P(n, r) = \frac{n!}{(n - r)!}$$

Donde n es el número total de elementos y r el número de elementos a permutar.

- **Permutaciones de objetos iguales:**

$$P(n; n_1, n_2, \dots, n_k) = \frac{n!}{n_1! n_2! \dots n_k!}$$

Esta fórmula se utiliza cuando existen elementos repetidos dentro del conjunto.

3.2 Combinaciones

Se implementaron dos tipos de combinaciones:

- **Combinaciones de objetos diferentes (sin repetición):**

$$C(n, r) = \binom{n}{r} = \frac{n!}{r!(n-r)!}$$

- **Combinaciones de objetos iguales (con repetición):**

$$C_R(n, r) = \binom{n+r-1}{r} = \frac{(n+r-1)!}{r!(n-1)!}$$

4 Descripción de la Implementación

4.1 Gestión de Conjuntos y Relaciones

El código utiliza clases en Python para mantener una estructura organizada. El objeto principal `RelacionesApp` permite:

- Crear y modificar conjuntos
- Definir un universo referencial
- Crear relaciones entre conjuntos
- Ejecutar operaciones lógicas sobre conjuntos

4.2 Funciones de Conteo

Los métodos `combinaciones_objetos_diferentes`, `combinaciones_objetos_iguales`, `permutaciones_objetos_diferentes` y `permutaciones_objetos_iguales` implementan las fórmulas descritas usando el módulo `itertools` y las funciones `factorial()` y `Counter()` de Python.

Cada cálculo imprime:

- Todas las combinaciones o permutaciones generadas
- La fórmula aplicada
- El total de resultados obtenidos

4.3 Interfaz Interactiva

El programa ejecuta un menú en consola con 18 opciones numeradas. El usuario ingresa el conjunto, el valor de k (cuando aplica) y elige si desea realizar una combinación o permutación, con o sin repetición.

5 Resultados y Ejemplos

5.1 Ejemplo 1: Permutaciones de Objetos Diferentes

```
=====
OPERACIONES CON RELACIONES Y CONTEO
=====
1. Definir conjunto referencial (universo)
2. Crear nuevo conjunto
3. Crear nueva relación
4. Operaciones básicas de conjuntos (U, n, -, ')
5. Producto cartesiano
6. Verificar si relación es reflexiva
7. Verificar si relación es simétrica
8. Verificar si relación es transitiva
9. Composición de relaciones (RoS)
10. Potencia de una relación (R^n)
11. Verificar operación binaria
12. Mostrar conjuntos y relaciones
13. Ejecutar ejemplo del proyecto
14. Calcular combinaciones de objetos iguales
15. Calcular combinaciones de objetos diferentes
16. Calcular permutaciones de objetos iguales
17. Calcular permutaciones de objetos diferentes
18. Salir
=====

Seleccione una opción (1-18): 17
Conjuntos disponibles: A, B, C, N, L, M
Seleccione el conjunto: M
Número de elementos a permutar (k): 3

Permutaciones SIN repetición (Objetos Diferentes) de M (k=3):
('1', '2', '3')
('1', '3', '2')
('2', '1', '3')
('2', '3', '1')
('3', '1', '2')
('3', '2', '1')

Total de permutaciones: 6
```

Figure 1: Ejecución en consola - Permutaciones de objetos diferentes

5.2 Ejemplo 2: Permutaciones de Objetos Iguales

```
=====
OPERACIONES CON RELACIONES Y CONTEO
=====
1. Definir conjunto referencial (universo)
2. Crear nuevo conjunto
3. Crear nueva relación
4. Operaciones básicas de conjuntos (U, n, -, ')
5. Producto cartesiano
6. Verificar si relación es reflexiva
7. Verificar si relación es simétrica
8. Verificar si relación es transitiva
9. Composición de relaciones (RoS)
10. Potencia de una relación (R^n)
11. Verificar operación binaria
12. Mostrar conjuntos y relaciones
13. Ejecutar ejemplo del proyecto
14. Calcular combinaciones de objetos iguales
15. Calcular combinaciones de objetos diferentes
16. Calcular permutaciones de objetos iguales
17. Calcular permutaciones de objetos diferentes
18. Salir
=====

Seleccione una opción (1-18): 16
Conjuntos disponibles: A, B, C
Seleccione el conjunto: A

--- PERMUTACIONES DE OBJETOS IGUALES ---
Para este cálculo, necesitamos la secuencia de elementos tal cual fueron ingresados, incluyendo repeticiones.
El conjunto 'A' solo tiene elementos únicos: ['1', 'a', 'b']
Ingrese la secuencia de elementos *con repetición* (separados por comas, ej: a,a,b,c): a,a,1

Secuencia ingresada: ['a', 'a', '1']
Total de elementos (N): 3
Frecuencias (n1, n2, ...): {'a': 2, '1': 1}

Fórmula:  $N! / (n1! * n2! * \dots)$ 
Total de Permutaciones de Objetos Iguales: 3
```

Figure 2: Ejecución en consola - Permutaciones de objetos iguales

5.3 Ejemplo 3: Combinaciones de Objetos Diferentes

```
=====
OPERACIONES CON RELACIONES Y CONTEO
=====
1. Definir conjunto referencial (universo)
2. Crear nuevo conjunto
3. Crear nueva relación
4. Operaciones básicas de conjuntos (U, n, -, ')
5. Producto cartesiano
6. Verificar si relación es reflexiva
7. Verificar si relación es simétrica
8. Verificar si relación es transitiva
9. Composición de relaciones (RoS)
10. Potencia de una relación (R^n)
11. Verificar operación binaria
12. Mostrar conjuntos y relaciones
13. Ejecutar ejemplo del proyecto
14. Calcular combinaciones de objetos iguales
15. Calcular combinaciones de objetos diferentes
16. Calcular permutaciones de objetos iguales
17. Calcular permutaciones de objetos diferentes
18. Salir
=====

Seleccione una opción (1-18): 14
Conjuntos disponibles: A, B, C
Seleccione el conjunto: A
Número de elementos a combinar (k): 2

Combinaciones con repetición de A (k=2):
('1', '1')
('1', 'a')
('1', 'b')
('a', 'a')
('a', 'b')
('b', 'b')

Total de combinaciones: 6
```

Figure 3: Ejecución en consola - Combinaciones de objetos diferentes

5.4 Ejemplo 4: Combinaciones de Objetos Iguales

```
=====
OPERACIONES CON RELACIONES Y CONTEO
=====
1. Definir conjunto referencial (universo)
2. Crear nuevo conjunto
3. Crear nueva relación
4. Operaciones básicas de conjuntos (U, n, -, '')
```

5. Producto cartesiano

6. Verificar si relación es reflexiva

7. Verificar si relación es simétrica

8. Verificar si relación es transitiva

9. Composición de relaciones ($R \circ S$)

10. Potencia de una relación (R^n)

11. Verificar operación binaria

12. Mostrar conjuntos y relaciones

13. Ejecutar ejemplo del proyecto

14. Calcular combinaciones de objetos iguales

15. Calcular combinaciones de objetos diferentes

16. Calcular permutaciones de objetos iguales

17. Calcular permutaciones de objetos diferentes

18. Salir

```
=====

Seleccione una opción (1-18): 14
Conjuntos disponibles: A, B, C
Seleccione el conjunto: A
Número de elementos a combinar (k): 2

Combinaciones con repetición de A (k=2):
('1', '1')
('1', 'a')
('1', 'b')
('a', 'a')
('a', 'b')
('b', 'b')

Total de combinaciones: 6
```

Figure 4: Ejecución en consola - Combinaciones de objetos iguales

6 Detalles Técnicos

- Lenguaje utilizado: Python 3
- Librerías: `itertools`, `math`, `collections`, `re`
- Entrada y salida: consola interactiva
- Validación: manejo de errores, vacíos y casos extremos

- Total de opciones en menú: 18

7 Conclusiones

1. La implementación en Python permitió aplicar los conceptos de permutaciones y combinaciones de forma práctica e interactiva.
2. El programa integra tanto objetos diferentes como iguales, abarcando todos los tipos de conteo vistos en clase.
3. Las fórmulas se validan con resultados generados por código, fortaleciendo la comprensión teórica.
4. La estructura modular del código facilita su extensión para incluir nuevos temas de Matemática Discreta.
5. Se resolvieron problemas de indentación y definición de métodos, asegurando la ejecución sin errores.

8 Referencias

- Rosen, K. (2019). *Discrete Mathematics and Its Applications* (8th ed.). McGraw-Hill.
- Universidad del Valle de Guatemala. (2025). *Matemática Discreta 1 (MM2015)*.