Metodología:

Nos dividimos los distintos bloques de código de esta sección. En mi caso lo que hice fueron los siguientes puntos:

1. Importación de Módulos Esenciales

Mi primera tarea fue asegurar la correcta funcionalidad del programa al importar el módulo itertools. Esta importación es fundamental, ya que itertools nos proporciona la función product, una herramienta altamente eficiente y optimizada para el cálculo del producto cartesiano entre dos o más colecciones de datos, lo cual es esencial para una de las operaciones finales del proyecto.

2. Función para la Recopilación de Años de Nacimiento

Desarrollé la función principal obtener_anios_nacimiento(). Esta función está diseñada para interactuar con el usuario y recopilar los años de nacimiento de los integrantes del grupo de forma dinámica.

- Solicita al usuario que ingrese los años uno por uno, validando que sean valores numéricos y estén dentro de un rango realista (entre 1900 y el año actual, que se ajusta automáticamente para mayor precisión).
- Maneja posibles errores de entrada (por ejemplo, si el usuario ingresa texto en lugar de números).
- Permite ingresar años ficticios si hay duplicados, asegurando que cada "integrante" tenga un año asociado, aunque dos personas reales compartan el mismo año de nacimiento.
- Acumula los años válidos en una lista y lleva un conteo del número total de integrantes ingresados.
- El proceso de entrada finaliza cuando el usuario escribe 'FIN'.
- Finalmente, la función devuelve tanto la lista de años de nacimiento recopilados como la cantidad total de integrantes, lo cual es crucial para las fases posteriores del análisis.

```python
def obtener_anios_nacimiento():

    anios_nacimiento_grupo = []
    contador_integrantes = 0
    ingreso_anios = True

    while ingreso_anios == True:
        entrada = input(f"Ingrese el año de nacimiento del integrante {contador_integrantes+1} o 'FIN' para terminar: ").upper()

        if entrada == 'FIN':
            ingreso_anios = False
        else:
            try:
                anio = int(entrada)
                if anio >= 1900 and anio <= 2025:
                    anios_nacimiento_grupo.append(anio)
                    contador_integrantes += 1
                else:
                    print("Por favor ingresar un año de nacimiento realista (Entre 1900 y 2025)")
            except ValueError:
                print("Entrada invalida. Por favor ingrese un número entero o 'FIN'")

    return anios_nacimiento_grupo, contador_integrantes
```

3. Ejecución Principal del Script

Me responsabilicé de la estructura del bloque de ejecución principal (if __name__ == "__main__":). Este bloque es el punto de entrada de nuestro programa.

Cuando el script se ejecuta directamente, es lo primero que se activa.
Su función inicial es imprimir un mensaje de bienvenida.
Acto seguido, realiza la llamada a la función obtener_anios_nacimiento(), almacenando los datos de los años y la cantidad de integrantes que el usuario ha proporcionado. Esto asegura que tengamos la información base necesaria para proceder con todos los análisis subsiguientes.

```python
if __name__ == "__main__":
    print("Iniciando prueba")

    anios_nacimiento, cantidad_integrantes = obtener_anios_nacimiento()
```

4. Verificación y Resumen de Datos Ingresados

Implementé el segmento de código encargado de la verificación inicial de los datos recopilados.

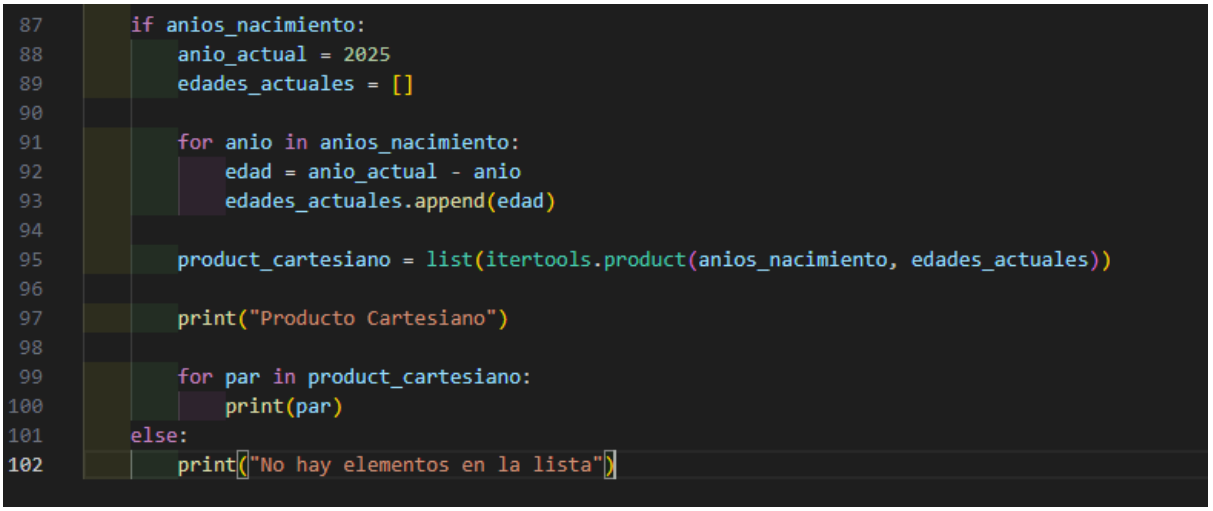
Este bloque evalúa si la lista de anios_nacimiento contiene algún elemento.
Si la lista está vacía (es decir, no se ingresaron años), se notifica al usuario.
En caso contrario, se muestra un resumen claro: la lista completa de los años de nacimiento ingresados y la cantidad total de integrantes, proporcionando una confirmación inmediata de los datos recibidos.

```python
    if anios_nacimiento == []:
        print("No se ingresaron años de nacimiento")
    else :
        print(f"Años de nacimiento recopilados: {anios_nacimiento}")
        print(f"Cantidad de integrantes: {cantidad_integrantes}")
```

5. Cálculo del Producto Cartesiano

Finalmente, desarrollé el bloque para el cálculo y la presentación del producto cartesiano. Esta es una operación avanzada que permite ver todas las combinaciones posibles entre dos conjuntos de datos:

- Primero, el programa determina el año actual (de forma dinámica).
- Con el año actual, se calcula la edad actual de cada integrante basándose en su año de nacimiento, almacenando estas edades en una nueva lista.
- Luego, se utiliza la función itertools.product() para generar todas las posibles tuplas donde cada tupla contiene un año de nacimiento y una edad actual. Esto crea una relación de "todos con todos" entre ambos conjuntos de datos.
- Para facilitar la visualización, el resultado se convierte a una lista y se imprime de manera ordenada, mostrando cada par (año de nacimiento, edad actual) en una línea separada. Este resultado es valioso para comprender la interrelación de estos dos atributos dentro del grupo.



```
87    if anios_nacimiento:
88        anio_actual = 2025
89        edades_actuales = []
90
91        for anio in anios_nacimiento:
92            edad = anio_actual - anio
93            edades_actuales.append(edad)
94
95        product_cartesiano = list(itertools.product(anios_nacimiento, edades_actuales))
96
97        print("Producto Cartesiano")
98
99        for par in product_cartesiano:
100           print(par)
101   else:
102       print("No hay elementos en la lista")
```