

# Práctica 1

Miguel Jarne Anguita, Pablo Jiménez Poyatos y María Puyol Romera

27 de febrero de 2024

El objeto de esta práctica es manejar una gran cantidad de datos almacenados en distintos archivos de una manera eficiente. A partir de un gran conjunto de datos, en el escenario más básico, se pretende establecer una condición y poder obtener los datos que la cumplen, así como los archivos en los que estaban almacenados esos datos.

Por otro lado, en un caso más avanzado, se pretende establecer una condición que pueda afectar a relaciones entre esos datos, partiendo de la base de que tomaremos la lista de los datos que hayan pasado el primer filtro como candidatos a pasar el segundo.

Para entender el funcionamiento del programa, proponemos el siguiente ejemplo:

Supóngase como primer filtro la condición de que **un número sea primo**. Obsérvese que se trata de una condición que se irá comprobando dato a dato. Mientras se va conformando la cola de elementos que cumplan esta condición (ya que iremos almacenando los que la cumplan en una cola), iremos comprobando la segunda condición:

Supóngase como segunda condición el hecho de que **dos números de una lista sean iguales**. Esta condición, en contraposición con la primera, no se va comprobando dato a dato, sino que recibirá una **lista** como parámetro de entrada, y devolverá un *booleano* en función de si hay dos números iguales en la lista o no. Para esta condición, a medida que se va conformando nuestra cola de números primos, se irá comprobando esta condición. Así, en el momento en el que encuentre dos iguales, se ejecutará la instrucción *ev.set()* y se detendrán los procesos, pues ya se ha encontrado lo que se buscaba.

Otro ejemplo de condición global pudiera ser que el **nr datos requeridos** fuera igual a un determinado número en particular, en cuyo caso bastará con comprobar la longitud de la lista que se va conformando y ejecutar la instrucción *ev.set()* en el momento en el que encontremos el número deseado.

Además, se debe tener en cuenta que para lo que hemos llamado segundas condiciones, deben ser condiciones que se comprueben para listas de datos y no para otro tipo. Además, el programa devolverá una tupla compuesta por la lista de elementos que verifican todas las condiciones impuestas y *True* si existe alguna solución y la lista vacía y *False* si no existe ninguna combinación de datos que satisfaga las condiciones. En el código se muestran varios ejemplos de posibles condiciones.

Por último, nótese que en la función *main* se pueden modificar en función del número de buscadores, las condiciones que se quieran imponer, etc. Además, se ha introducido un fragmento de código por si en el directorio utilizado no existiera ninguna carpeta con el nombre adecuado.