

Sprint 1 - Manejo eficiente de datos al programar aplicaciones distribuidas

Cantidad total de vuelos que existieron en cada día del mes. Denomina al mapper como **mapper_1.py** y al reducer como **reducer_1.py**).

Mapper_1.py

```
import sys

def main():
    """Función principal del mapper que procesa entradas de v
    for line in sys.stdin:
        # Dividir la línea por comas
        fields = line.strip().split(',')

        # Ignorar la línea de encabezado
        if fields[0] == 'Year':
            continue

        # Extraer el día del mes
        day_of_month = fields[3] # Columna "DayofMonth"

        # Emitir el día del mes con un conteo de 1
        print(f"{day_of_month}\t1")

if __name__ == "__main__":
    main()
```

Reducer_1.py

```

import sys

def main():
    """Función principal del reducer que suma el conteo de vu
    current_day = None
    total_flights = 0

    for line in sys.stdin:
        line = line.strip()
        day_of_month, count = line.split('\t')
        count = int(count)

        if current_day == day_of_month:
            total_flights += count
        else:
            if current_day is not None:
                print(f"{current_day}\t{total_flights}")
            current_day = day_of_month
            total_flights = count

    if current_day is not None:
        print(f"{current_day}\t{total_flights}")

if __name__ == "__main__":
    main()

```

Conteo total de vuelos que salieron de los diferentes orígenes (campo OriginAirportID). Denomina al mapper como **mapper_2.py** y al reducer como **reducer_2.py**).

Mapper_2.py

```

import sys

def mapper():

```

```

"""Función mapper que procesa entradas de datos de vuelos
for line in sys.stdin:
    # Eliminar espacios en blanco y dividir la línea en campos
    line = line.strip()
    fields = line.split(',')

    # Asegurarse de que la línea tenga suficientes campos
    if len(fields) > 15: # Ajusta según el número total de campos
        origin_airport_id = fields[12] # Suponiendo que el ID del aeropuerto está en el índice 12
        if origin_airport_id.isdigit():
            # Emitir la clave y el valor
            print(f"{origin_airport_id}\t1")

if __name__ == "__main__":
    mapper()

```

Reducer_2.py

```

import sys

def reducer():
    """Función reducer que suma el conteo de vuelos por ID de aeropuerto
    current_airport_id = None
    total_flights = 0

    for line in sys.stdin:
        line = line.strip()
        if not line:
            continue

        airport_id, count = line.split("\t")
        count = int(count)

        if current_airport_id == airport_id:
            total_flights += count
        else:
            if current_airport_id is not None:

```

```
        # Imprimir el resultado para el aeropuerto an
        print(f"{current_airport_id}\t{total_flights}")
    current_airport_id = airport_id
    total_flights = count

# Imprimir el resultado para el último aeropuerto
if current_airport_id is not None:
    print(f"{current_airport_id}\t{total_flights}")

if __name__ == "__main__":
    reducer()
```