

4. Estructuras de datos simples

Objetivos: Usar estructuras de datos simples como listas, tuplas, conjuntos y diccionarios en python. Refuerzo de bucles y funciones. Aumento entrada/salida con ficheros. Retocar cadenas de caracteres.

1. Crea un fichero con una lista de asignaciones de pasajeros de asientos donde en cada línea aparece primero nombre y luego identificador del asiento separados por coma, por ejemplo:

```
Maria, 2A
Manuel, 21D
Dorotea, 14F
Samuel, 10C
```

Escribe un programa que lea tal fichero y lo convierta automáticamente a un fichero donde en la primera línea aparecen todos los asientos ocupados y en la segunda línea los nombres en el orden según sus asientos.

2. Aumenta el programa del apartado anterior a un pequeño sistema de reservas para un avión [A320](#). El programa debe actuar con un menú principal donde la entrada de una letra selecciona un módulo de actuación, ejemplos `r` activa el módulo de reserva, `l` activa el módulo de generar lista, `q` sale del programa.

El menú debe contener por lo menos los siguientes módulos:

- leer un fichero (con formato como en el apartado arriba)
- listar todos los asientos acupados
- añadir un reserva con un nombre y un asiento (ojo, solo un pasajero por asiento, y solo identificadores de asientos posibles)
- guardar las reservas en un fichero
- borrar una reserva (o bien con nombre o bien con asiento)
- mostar número de asientos todavía libres
- aumenta con funcionalidades que te resultan útiles (tales pueden ayudarte también para la depuración del programa)

Para no complicarte la vida al principio, puedes asumir que las filas de los asientos tienen números consecutivos y que siempre hay la misma cantidad por fila. Emplea bien y con sentido las estructuras de datos ya conocidas, por ejemplo, tuplas para unir pasajero con asiento, diccionarios para la ocupación del avión, conjuntos para los caracteres del menú, etc. Notarás que cuando más modular desarrollas el programa, más fácil es completar la tarea.