

NIVEL 3

ESTRUCTURA DE DATOS COMPUESTOS



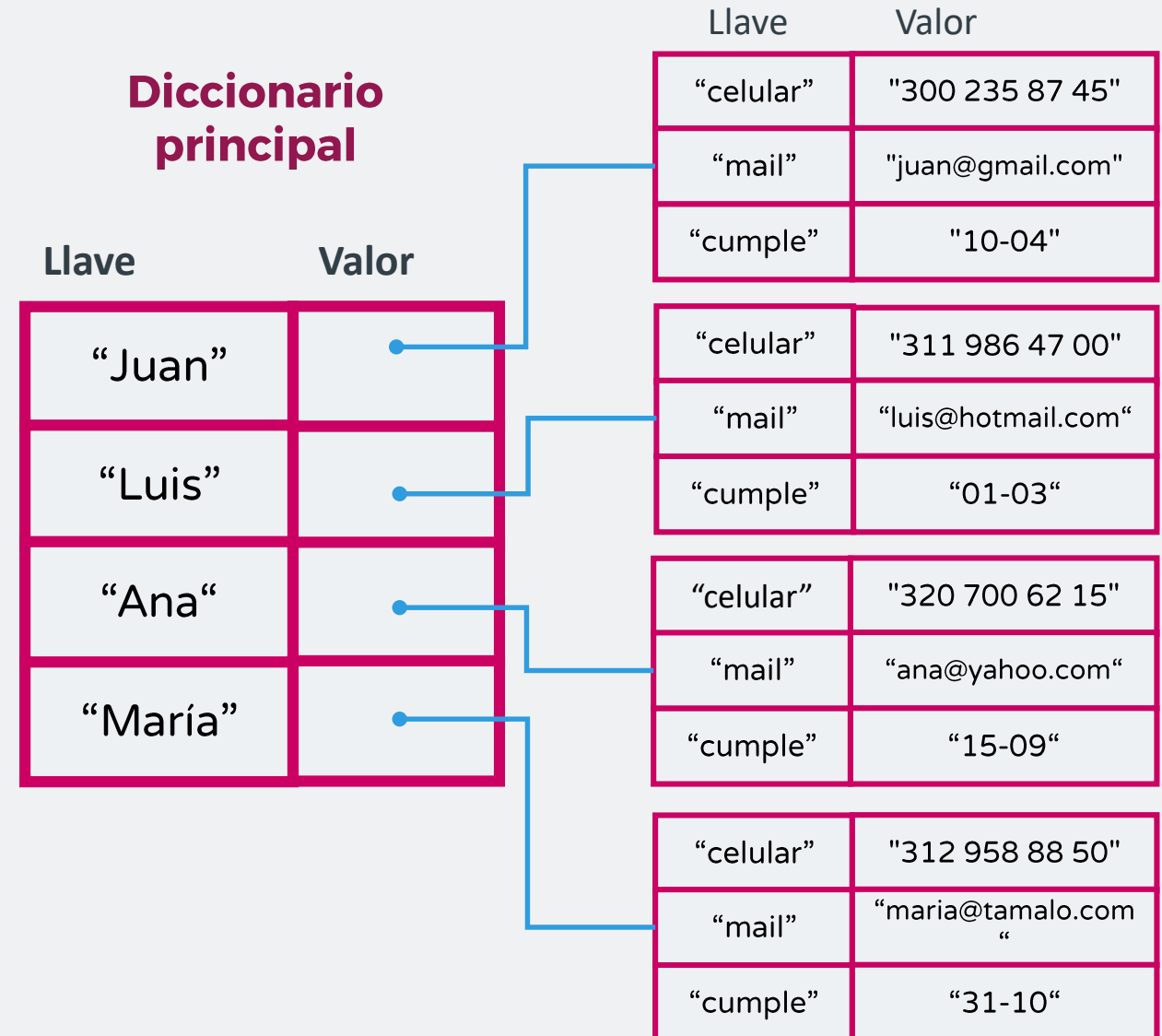
¿CUÁNDO NECESITAMOS ESTRUCTURAS MÁS COMPLEJAS?

En este diccionario sólo podemos guardar el número telefónico de un contacto. ¿Qué pasa si queremos guardar más información: dirección, correo electrónico, fecha de nacimiento?

```
In [1]: d = {}  
  
In [2]: d["Juan"] = "300 235 87 45"  
  
In [3]: d["Luis"] = "311 986 47 00"  
  
In [4]: d["Ana"] = "320 700 62 15"  
  
In [5]: d["María"] = "312 958 88 50"  
  
In [6]: print(d)  
{'Juan': '300 235 87 45', 'Luis': '311 986 47 00',  
'Ana': '320 700 62 15', 'María': '312 958 88 50'}
```

DICCIONARIO DE DICCIONARIOS

Esto se logra con un diccionario de diccionarios, donde la llave es el nombre del contacto y el valor es otro diccionario con los datos que queremos guardar!



ASÍ CREAMOS EL DICCIONARIO DE DICCIONARIOS EN PYTHON

```
Terminal 2/A ✕  
In [1]: directorio = {}  
  
In [2]: directorio["Juan"] = {"celular": "300 235 87 45", "mail": "juan@gmail.com", "cumple": "10-04"}  
  
In [3]: directorio["Luis"] = {"celular": "311 986 47 00", "mail": "luis@hotmail.com", "cumple": "01-03"}  
  
In [4]: directorio["Ana"] = {"celular": "320 700 62 15", "mail": "ana@yahoo.com", "cumple": "15-09"}  
  
In [5]: directorio["Maria"] = {"celular": "312 958 88 50", "mail": "maria@tamalo.com", "cumple": "31-10"}  
  
In [6]: print(directorio)  
{'Juan': {'celular': '300 235 87 45', 'mail': 'juan@gmail.com', 'cumple': '10-04'}, 'Luis': {'celular':  
'311 986 47 00', 'mail': 'luis@hotmail.com', 'cumple': '01-03'}, 'Ana': {'celular': '320 700 62 15',  
'mail': 'ana@yahoo.com', 'cumple': '15-09'}, 'Maria': {'celular': '312 958 88 50', 'mail':  
'maria@tamalo.com', 'cumple': '31-10'}}
```

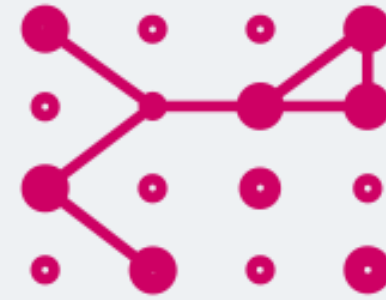
Y ASÍ LO RECORREMOS

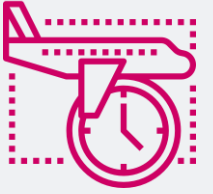


Terminal 2/A

```
In [9]: for cada_clave in directorio:
...:     print(cada_clave,"->",directorio[cada_clave])
...:
Juan -> {'celular': '300 235 87 45', 'mail': 'juan@gmail.com', 'cumple': '10-04'}
Luis -> {'celular': '311 986 47 00', 'mail': 'luis@hotmail.com', 'cumple': '01-03'}
Ana -> {'celular': '320 700 62 15', 'mail': 'ana@yahoo.com', 'cumple': '15-09'}
Maria -> {'celular': '312 958 88 50', 'mail': 'maria@tamalo.com', 'cumple': '31-10'}
```

**¡LISTOS PARA UN
DICCIONARIO CON
MAS INFORMACIÓN!**





LISTADO DE VUELOS

Tenemos a nuestra disposición la información de **7142** vuelos que ocurrieron un mismo día dentro de Norte América. Para cada vuelo conocemos:

- Código del vuelo: combinación del código de la aerolínea y un número (ej. DL1802)
- Aerolínea: código IATA de una aerolínea (ej. AA, DL, AV)
- Origen: código del aeropuerto de origen del vuelo (ej. ATL, SFO, BOG, BAQ)
- Destino: código del aeropuerto de destino del vuelo (ej. ATL, SFO, BOG, BAQ)
- Distancia: distancia en millas entre el aeropuerto origen y el destino
- Hora de salida: hora de salida real del vuelo (hhmm). Para un vuelo que salió a las 3:30pm aparecerá 1530
- Duración: duración en minutos del vuelo
- Retraso: retraso en minutos del vuelo con respecto a la hora de llegada programada. Cuando es negativo indica que el vuelo llegó antes de lo previsto

USAREMOS UN DICCIONARIO DE DICCIONARIOS

Para manejar la información de los vuelos, usaremos un diccionario de diccionarios, donde las llaves son los códigos de los vuelos y los valores correspondientes son a su vez diccionarios con el resto de la información de cada vuelo, así:

Diccionario principal

Llave	Valor
"XE2679"	•
...	
...	
"UA63"	•

Llave	Valor
"aerolinea"	"XE"
"origen"	"EWR"
"destino"	"MYR"
"distancia"	"550"
"salida"	"1110"
"duracion"	"113"
"retraso"	"18"

"aerolinea"	"UA"
"origen"	"MSP"
"destino"	"DEN"
"distancia"	"680"
"salida"	"1439"
"duracion"	"147"
"retraso"	"34"

ASÍ CREAMOS EL DICCIONARIO DE DICCIONARIOS EN PYTHON

```
Terminal 2/A x
```

```
In [11]: vuelos = {}

In [12]: vuelos["XE2679"] = {"aerolinea": "XE", "origen": "EWR", "destino": "MYR",
                             "distancia": "550", "salida": "110", "duracion": "113", "retraso": "18"}

In [13]: vuelos["YV7138"] = {"aerolinea": "YV", "origen": "ORD", "destino": "ABE",
                             "distancia": "654", "salida": "1426", "duracion": "144", "retraso": "43"}

In [14]: vuelos["002880"] = {"aerolinea": "00", "origen": "MKE", "destino": "EWR",
                             "distancia": "725", "salida": "1830", "duracion": "154", "retraso": "104"}

In [15]: vuelos["UA63"] = {"aerolinea": "UA", "origen": "MSP", "destino": "DEN",
                           "distancia": "680", "salida": "1439", "duracion": "147", "retraso": "34"}
```

¿CÓMO CARGAMOS LA INFORMACIÓN DE LOS 7142 VUELOS?

No podemos hacerlo «a mano» uno por uno:

- Nos tomaría demasiado tiempo
- Estaría sujeta a errores de tecleo
- Si la base de información cambia, necesitamos poder cargarla igualmente

SOLUCIÓN: leer la información de un archivo !!!



VEAMOS EL ARCHIVO CSV "EN BRUTO"

La primera línea del archivo CSV es diferente: contiene las etiquetas de las columnas. A partir de la segunda línea está la información de los vuelos. Hay un vuelo por línea



```
Archivo Edición Formato Ver Ayuda
aerolinea,codigo_vuelo,origen,destino,distancia,salida,duracion,retraso
XE,XE2679,EWR,MYR,550,1110,113,18
YV,YV7138,ORD,ABE,654,1426,144,43
OO,OO1976,ORD,CVG,264,647,80,44
UA,UA31,DEN,SFO,967,1125,158,1
US,US1409,CLT,MSY,651,1649,155,44
WN,WN1107,LAX,BNA,1797,1548,234,17
EV,EV4606,ATL,HSV,151,1822,56,3
F9,F9105,MSP,DEN,680,2007,117,8
FL,FL658,ATL,HPN,780,1840,112,47
HA,HA11,SFO,HNL,2398,848,325,23
MQ,MQ3005,LAX,SBP,155,1301,58,7
NW,NW1448,LAN,DTW,74,1340,65,117
OH,OH5096,CVG,ORD,264,1805,103,187
9E,9E2949,LAN,DTW,74,1745,61,75
AA,AA407,ORD,STL,258,659,66,15
AS,AS89,SEA,ANC,1449,2210,196,0
B6,B61087,JFK,CLT,541,2219,124,25
CO,CO334,RDU,EWR,416,1901,83,47
```

VEAMOS LA ESTRUCTURA DEL ARCHIVO CSV

aerolinea	codigo_vuelo	origen	destino	distancia	salida	duracion	retraso
XE	XE2679	EWR	MYR	550	1110	113	18
YV	YV7138	ORD	ABE	654	1426	144	43
OO	OO2880	MKE	EWR	725	1830	154	104
UA	UA63	MSP	DEN	680	1439	147	34
US	US311	CLT	PHX	1774	1801	279	20
WN	WN1283	SLC	BOI	291	2102	61	68
EV	EV4838	BMI	ATL	533	1437	96	25
FL	FL434	ATL	PNS	272	1440	68	16
MQ	MQ3576	CID	DFW	685	1540	114	-11
NW	NW433	DTW	SFO	2079	1213	316	22
OH	OH5399	CVG	MSP	596	2157	103	38
9E	9E4733	MSP	CLE	622	1934	180	79
AA	AA522	PDX	DFW	1616	1502	213	10
AS	AS333	SFO	SEA	679	1855	111	3
B6	B6505	EWR	FLL	1065	942	186	-12
CO	CO1764	IAH	ORD	925	1158	155	6
DL	DL420	SEA	JFK	2421	1309	318	12

LECTURA DEL ARCHIVO Y CONSTRUCCIÓN DEL DICCIONARIO

Esta función lee la línea por línea la información de los vuelos de un archivo cuyo nombre entra por parámetro y retorna el diccionario de diccionarios

Analicémosla detenidamente ...

```
def cargar_vuelos(ruta_archivo: str)->dict:

    vuelos = {}
    archivo = open(ruta_archivo)
    titulos = archivo.readline().split(",")
    print(titulos)

    linea = archivo.readline()
    while len(linea) > 0:
        datos = linea.split(",")
        codigo_vuelo = datos[1]
        vuelo = {}
        vuelo["aerolinea"] = datos[0]
        vuelo["origen"] = datos[2]
        vuelo["destino"] = datos[3]
        vuelo["distancia"] = datos[4]
        vuelo["salida"] = datos[5]
        vuelo["duracion"] = datos[6]
        vuelo["retraso"] = datos[7]
        vuelos[codigo_vuelo] = vuelo
        linea = archivo.readline()

    archivo.close()
    return vuelos
```

LECTURA DEL ARCHIVO Y CONSTRUCCIÓN DEL DICCIONARIO

Se crea el diccionario principal que será retornado al final de la función y lo guardamos en la variable llamada «vuelos»

```
def cargar_vuelos(ruta_archivo: str)->dict:

    vuelos = {}
    archivo = open(ruta_archivo)
    titulos = archivo.readline().split(",")
    print(titulos)

    linea = archivo.readline()
    while len(linea) > 0:
        datos = linea.split(",")
        codigo_vuelo = datos[1]
        vuelo = {}
        vuelo["aerolinea"] = datos[0]
        vuelo["origen"] = datos[2]
        vuelo["destino"] = datos[3]
        vuelo["distancia"] = datos[4]
        vuelo["salida"] = datos[5]
        vuelo["duracion"] = datos[6]
        vuelo["retraso"] = datos[7]
        vuelos[codigo_vuelo] = vuelo
        linea = archivo.readline()

    archivo.close()
    return vuelos
```

LECTURA DEL ARCHIVO Y CONSTRUCCIÓN DEL DICCIONARIO

Paso 1: Se abre el archivo con la instrucción **open** y en modo de lectura (**r**)

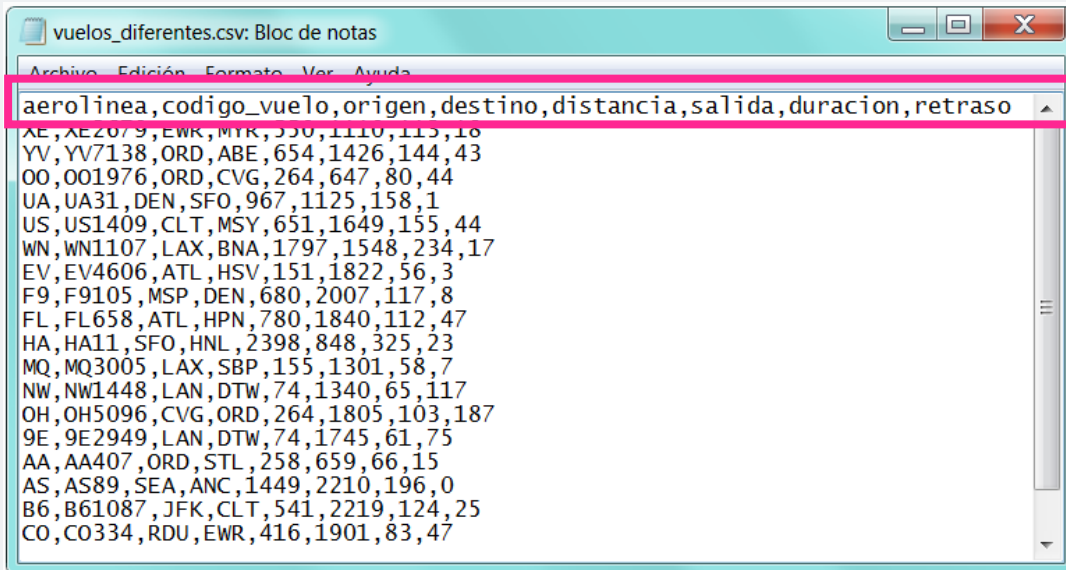
Paso 2: Se lee la información del archivo línea por línea con la instrucción **readline**. La lectura termina cuando la línea leída es vacía

Paso 3: Se cierra el archivo con la instrucción **close**

```
EjemploCargarVuelos.py x
1 def cargar_vuelos(nombre_archivo: str) -> dict:
2     vuelos = {}
3     archivo = open(nombre_archivo, "r")
4     titulos = archivo.readline()
5     print(titulos)
6
7     linea = archivo.readline()
8     while len(linea) > 0:
9         datos = linea.split(";")
10        codigo_vuelo = datos[1]
11        vuelo = {}
12        vuelo["aerolinea"] = datos[0]
13        vuelo["origen"] = datos[2]
14        vuelo["destino"] = datos[3]
15        vuelo["distancia"] = datos[4]
16        vuelo["salida"] = datos[5]
17        vuelo["duracion"] = datos[6]
18        vuelo["retraso"] = datos[7]
19        vuelos[codigo_vuelo] = vuelo
20        linea = archivo.readline()
21
22    archivo.close()
23    return vuelos
```


DETENGÁMONOS EN LA CONSTRUCCIÓN DEL DICCIONARIO

Se lee la primera línea del archivo que sabemos que solo contiene los títulos de las columnas



```
def cargar_vuelos(ruta_archivo: str)->dict:

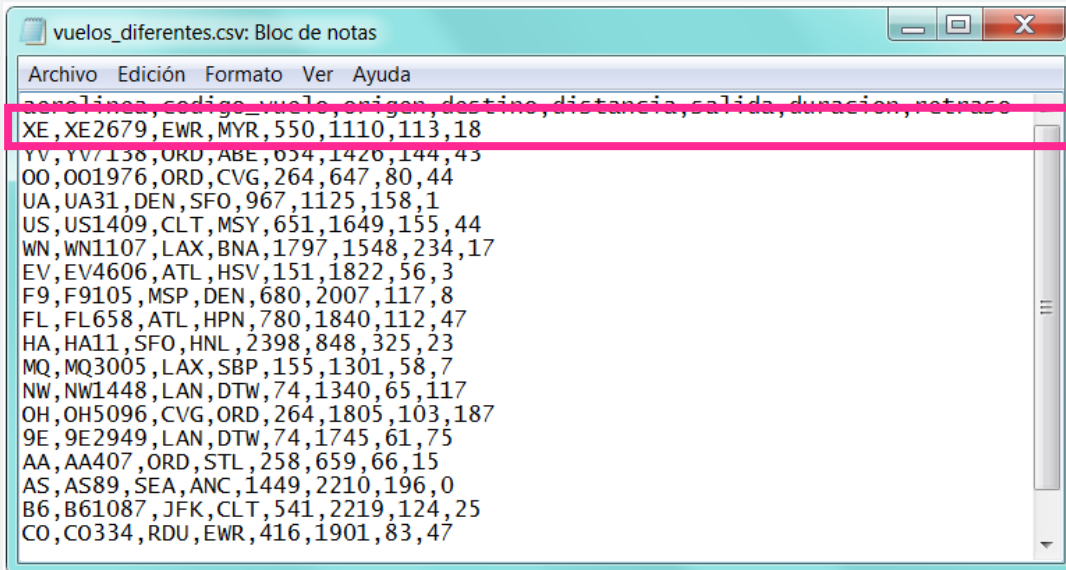
    vuelos = {}
    archivo = open(ruta_archivo)
    titulos = archivo.readline().split(",")
    print(titulos)

    linea = archivo.readline()
    while len(linea) > 0:
        datos = linea.split(",")
        codigo_vuelo = datos[1]
        vuelo = {}
        vuelo["aerolinea"] = datos[0]
        vuelo["origen"] = datos[2]
        vuelo["destino"] = datos[3]
        vuelo["distancia"] = datos[4]
        vuelo["salida"] = datos[5]
        vuelo["duracion"] = datos[6]
        vuelo["retraso"] = datos[7]
        vuelos[codigo_vuelo] = vuelo
        linea = archivo.readline()

    archivo.close()
    return vuelos
```

DETENGÁMONOS EN LA CONSTRUCCIÓN DEL DICCIONARIO

Se lee la segunda línea del archivo para empezar el ciclo que irá armando el diccionario a medida que va leyendo línea por línea del archivo



```
def cargar_vuelos(ruta_archivo: str)->dict:

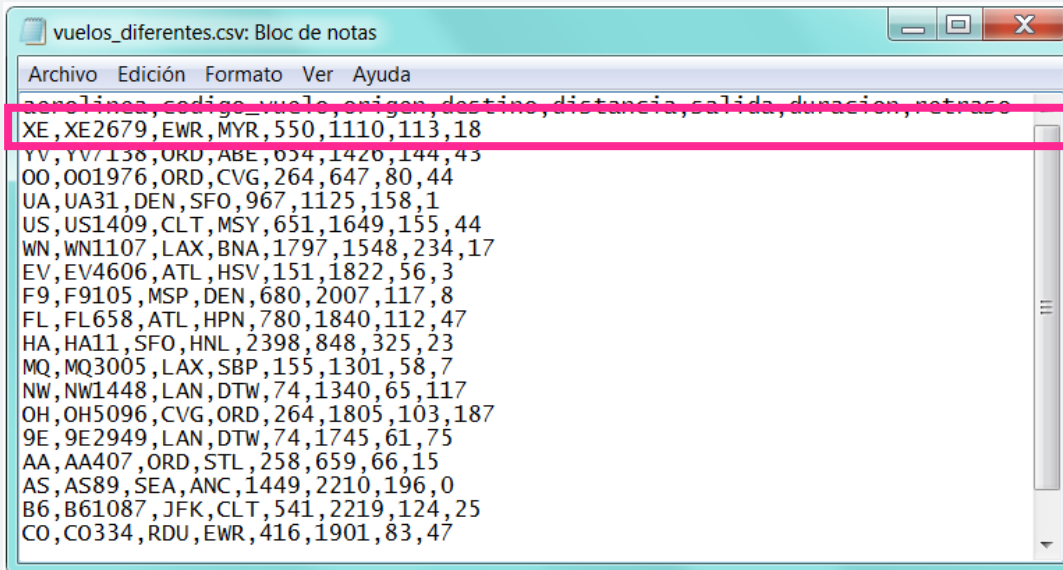
    vuelos = {}
    archivo = open(ruta_archivo)
    titulos = archivo.readline().split(",")
    print(titulos)

    linea = archivo.readline()
    while len(linea) > 0:
        datos = linea.split(",")
        codigo_vuelo = datos[1]
        vuelo = {}
        vuelo["aerolinea"] = datos[0]
        vuelo["origen"] = datos[2]
        vuelo["destino"] = datos[3]
        vuelo["distancia"] = datos[4]
        vuelo["salida"] = datos[5]
        vuelo["duracion"] = datos[6]
        vuelo["retraso"] = datos[7]
        vuelos[codigo_vuelo] = vuelo
        linea = archivo.readline()

    archivo.close()
    return vuelos
```

DETENGÁMONOS EN LA CONSTRUCCIÓN DEL DICCIONARIO

Mientras la línea leída no esté vacía (es decir, que se haya leído información del archivo)



```
def cargar_vuelos(ruta_archivo: str)->dict:

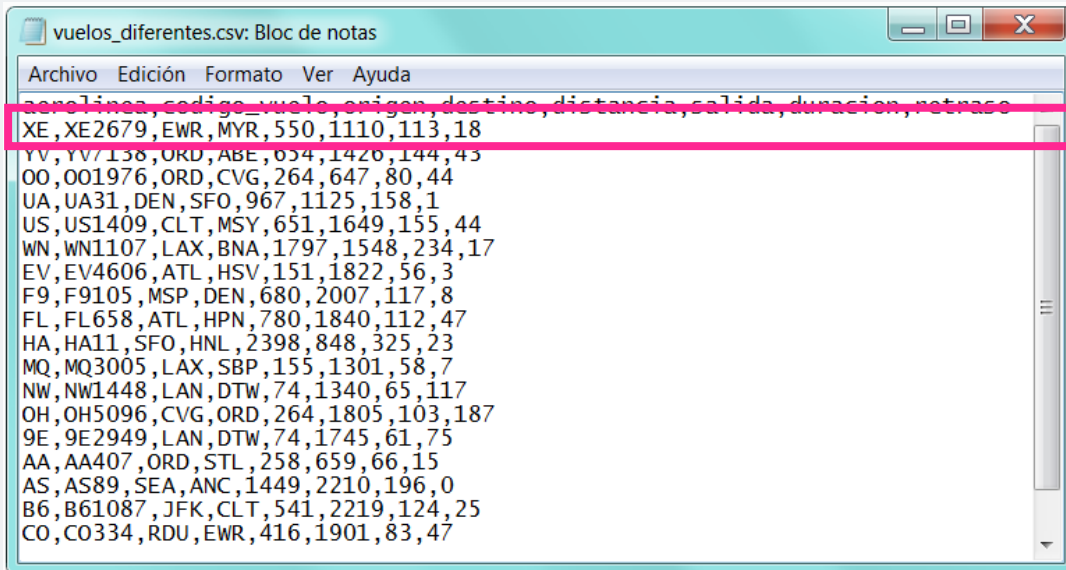
    vuelos = {}
    archivo = open(ruta_archivo)
    titulos = archivo.readline().split(",")
    print(titulos)

    linea = archivo.readline()
    while len(linea) > 0:
        datos = linea.split(",")
        codigo_vuelo = datos[1]
        vuelo = {}
        vuelo["aerolinea"] = datos[0]
        vuelo["origen"] = datos[2]
        vuelo["destino"] = datos[3]
        vuelo["distancia"] = datos[4]
        vuelo["salida"] = datos[5]
        vuelo["duracion"] = datos[6]
        vuelo["retraso"] = datos[7]
        vuelos[codigo_vuelo] = vuelo
        linea = archivo.readline()

    archivo.close()
    return vuelos
```

DETENGÁMONOS EN LA CONSTRUCCIÓN DEL DICCIONARIO

Se descompone la línea leída en sus palabras, tomando como separador la coma y dejando los valores en la lista «datos»



```
def cargar_vuelos(ruta_archivo: str)->dict:

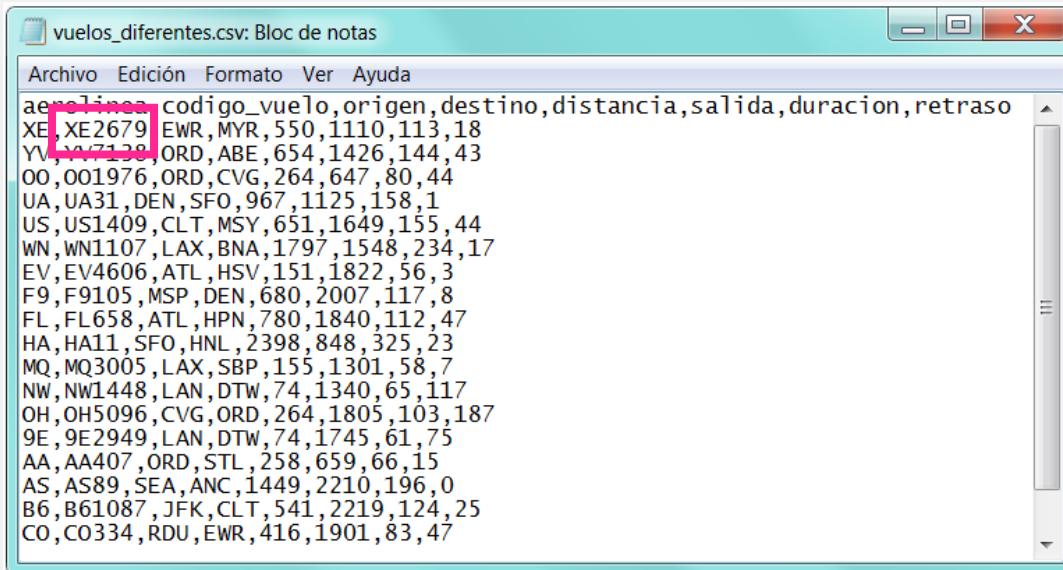
    vuelos = {}
    archivo = open(ruta_archivo)
    titulos = archivo.readline().split(",")
    print(titulos)

    linea = archivo.readline()
    while len(linea) > 0:
        datos = linea.split(",")
        codigo_vuelo = datos[1]
        vuelo = {}
        vuelo["aerolinea"] = datos[0]
        vuelo["origen"] = datos[2]
        vuelo["destino"] = datos[3]
        vuelo["distancia"] = datos[4]
        vuelo["salida"] = datos[5]
        vuelo["duracion"] = datos[6]
        vuelo["retraseo"] = datos[7]
        vuelos[codigo_vuelo] = vuelo
        linea = archivo.readline()

    archivo.close()
    return vuelos
```

DETENGÁMONOS EN LA CONSTRUCCIÓN DEL DICCIONARIO

Se extrae el código del vuelo, que sabemos, se encuentra en la posición 1 de la lista y lo guardamos en la variable «codigo_vuelo»



```
def cargar_vuelos(ruta_archivo: str)->dict:

    vuelos = {}
    archivo = open(ruta_archivo)
    titulos = archivo.readline().split(",")
    print(titulos)

    linea = archivo.readline()
    while len(linea) > 0:
        datos = linea.split(",")
        codigo_vuelo = datos[1]
        vuelo = {}
        vuelo["aerolinea"] = datos[0]
        vuelo["origen"] = datos[2]
        vuelo["destino"] = datos[3]
        vuelo["distancia"] = datos[4]
        vuelo["salida"] = datos[5]
        vuelo["duracion"] = datos[6]
        vuelo["retraso"] = datos[7]
        vuelos[codigo_vuelo] = vuelo
        linea = archivo.readline()

    archivo.close()
    return vuelos
```


DETENGÁMONOS EN LA CONSTRUCCIÓN DEL DICCIONARIO

Se crea el diccionario interno y lo guardamos en la variable «vuelo», este será añadido más adelante al diccionario principal

```
def cargar_vuelos(ruta_archivo: str)->dict:

    vuelos = {}
    archivo = open(ruta_archivo)
    titulos = archivo.readline().split(",")
    print(titulos)

    linea = archivo.readline()
    while len(linea) > 0:
        datos = linea.split(",")
        codigo_vuelo = datos[1]
        vuelo = {}
        vuelo["aerolinea"] = datos[0]
        vuelo["origen"] = datos[2]
        vuelo["destino"] = datos[3]
        vuelo["distancia"] = datos[4]
        vuelo["salida"] = datos[5]
        vuelo["duracion"] = datos[6]
        vuelo["retraso"] = datos[7]
        vuelos[codigo_vuelo] = vuelo
        linea = archivo.readline()

    archivo.close()
    return vuelos
```

DETENGÁMONOS EN LA CONSTRUCCIÓN DEL DICCIONARIO

Se llena el diccionario interno «vuelo» con toda la información del vuelo, asociando a cada llave, el valor correspondiente que se encuentra en la lista «datos»

```
def cargar_vuelos(ruta_archivo: str)->dict:

    vuelos = {}
    archivo = open(ruta_archivo)
    titulos = archivo.readline().split(",")
    print(titulos)

    linea = archivo.readline()
    while len(linea) > 0:
        datos = linea.split(",")
        codigo_vuelo = datos[1]
        vuelo = {}
        vuelo["aerolinea"] = datos[0]
        vuelo["origen"] = datos[2]
        vuelo["destino"] = datos[3]
        vuelo["distancia"] = datos[4]
        vuelo["salida"] = datos[5]
        vuelo["duracion"] = datos[6]
        vuelo["retraso"] = datos[7]
        vuelos[codigo_vuelo] = vuelo
        linea = archivo.readline()

    archivo.close()
    return vuelos
```


DETENGÁMONOS EN LA CONSTRUCCIÓN DEL DICCIONARIO

Se añade el diccionario interno «vuelo» al diccionario principal asociándolo como valor de la llave que guardamos previamente en la variable `codigo_vuelo`

```
def cargar_vuelos(ruta_archivo: str)->dict:

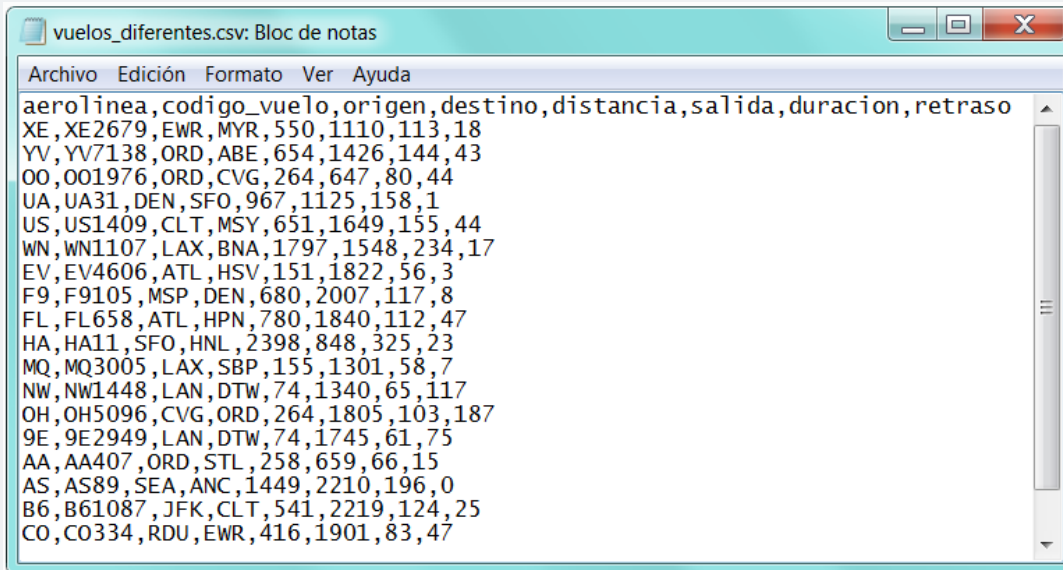
    vuelos = {}
    archivo = open(ruta_archivo)
    titulos = archivo.readline().split(",")
    print(titulos)

    linea = archivo.readline()
    while len(linea) > 0:
        datos = linea.split(",")
        codigo_vuelo = datos[1]
        vuelo = {}
        vuelo["aerolinea"] = datos[0]
        vuelo["origen"] = datos[2]
        vuelo["destino"] = datos[3]
        vuelo["distancia"] = datos[4]
        vuelo["salida"] = datos[5]
        vuelo["duracion"] = datos[6]
        vuelo["retraso"] = datos[7]
        vuelos[codigo_vuelo] = vuelo
        linea = archivo.readline()

    archivo.close()
    return vuelos
```

DETENGÁMONOS EN LA CONSTRUCCIÓN DEL DICCIONARIO

Se lee la siguiente línea del archivo y se repite el ciclo hasta que se hayan leído todas las líneas del archivo



```
def cargar_vuelos(ruta_archivo: str)->dict:

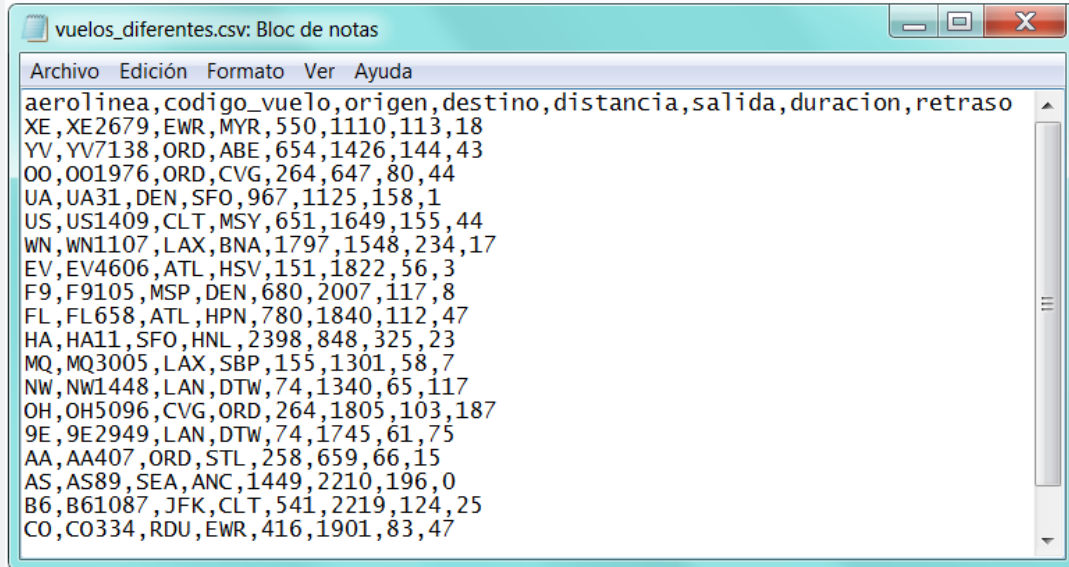
    vuelos = {}
    archivo = open(ruta_archivo)
    titulos = archivo.readline().split(",")
    print(titulos)

    linea = archivo.readline()
    while len(linea) > 0:
        datos = linea.split(",")
        codigo_vuelo = datos[1]
        vuelo = {}
        vuelo["aerolinea"] = datos[0]
        vuelo["origen"] = datos[2]
        vuelo["destino"] = datos[3]
        vuelo["distancia"] = datos[4]
        vuelo["salida"] = datos[5]
        vuelo["duracion"] = datos[6]
        vuelo["retraso"] = datos[7]
        vuelos[codigo_vuelo] = vuelo
        linea = archivo.readline()

    archivo.close()
    return vuelos
```

DETENGÁMONOS EN LA CONSTRUCCIÓN DEL DICCIONARIO

Finalmente, cerramos el archivo y retornamos la estructura de datos



```
def cargar_vuelos(ruta_archivo: str)->dict:

    vuelos = {}
    archivo = open(ruta_archivo)
    titulos = archivo.readline().split(",")
    print(titulos)

    linea = archivo.readline()
    while len(linea) > 0:
        datos = linea.split(",")
        codigo_vuelo = datos[1]
        vuelo = {}
        vuelo["aerolinea"] = datos[0]
        vuelo["origen"] = datos[2]
        vuelo["destino"] = datos[3]
        vuelo["distancia"] = datos[4]
        vuelo["salida"] = datos[5]
        vuelo["duracion"] = datos[6]
        vuelo["retraso"] = datos[7]
        vuelos[codigo_vuelo] = vuelo
        linea = archivo.readline()

    archivo.close()
    return vuelos
```

VEAMOS EL DICCIONARIO CREADO

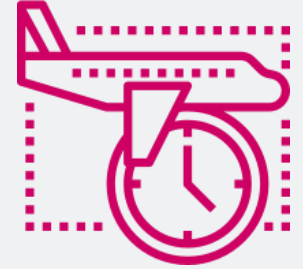
Resultado de la ejecución

```
Terminal 3/A x

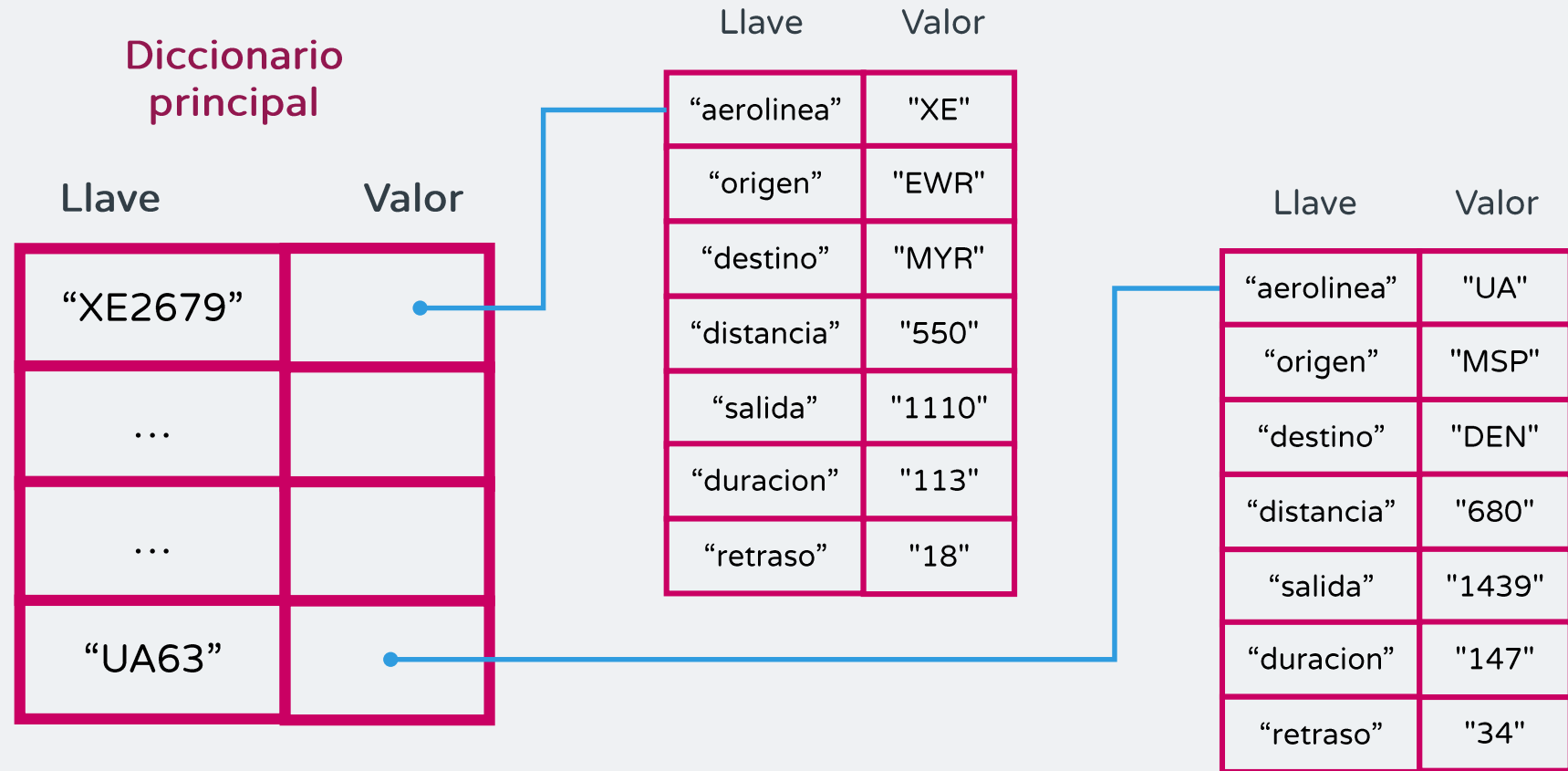
In [8]: cargar_vuelos("vueloss.csv")
aerolinea;codigo_vuelo;origen;destino;distancia;salida;duracion;retraso

Out[8]:
{'XE2679': {'aerolinea': 'XE',
            'origen': 'EWR',
            'destino': 'MYR',
            'distancia': '550',
            'salida': '1110',
            'duracion': '113',
            'retraso': '18\n'},
 'YV7138': {'aerolinea': 'YV',
            'origen': 'ORD',
            'destino': 'ABE',
            'distancia': '654',
            'salida': '1426',
            'duracion': '144',
            'retraso': '43\n'},
 '002880': {'aerolinea': '00',
            'origen': 'MKE',
            'destino': 'EWR',
            'distancia': '725',
            'salida': '1830',
```

PROCESEMOS EL DICCIONARIO DE DICCIONARIOS



Recordemos cómo es la estructura del diccionario de vuelos



AEROLÍNEA CON MAS VUELOS - ESTRATEGIA DE SOLUCIÓN

Vamos a utilizar un diccionario para ir contando los vuelos de cada aerolínea, al mismo tiempo que vamos calculando la que tiene más vuelos

```
def un_solo_recorrido(vuelos:dict)->str:
    #Se declaran las variables donde se van a guardar
    #la aerolínea con más vuelos y el número de vuelos de
    #dicha aerolínea
    aerolinea_con_mas_vuelos = None
    mas_vuelos = 0

    #Se construye un diccionario auxiliar en el cual
    #se van a guardar las aerolíneas (llaves) y la cantidad de
    #vuelos de cada una (valores)
    diccionario_conteo = {}

    #Se recorre el diccionario de vuelos
    for cada_vuelo in vuelos:
        #Se obtiene la aerolínea de cada vuelo
        aero = vuelos[cada_vuelo]["aerolinea"]
        #Se busca cuántas veces aparece esta aerolínea en
        #el diccionario de conteo que estamos armando.
        #Se usa la función get para esto.
        veces = diccionario_conteo.get(aero,0)
        #Se actualiza el diccionario de conteo con la cantidad
        #de veces incrementada en 1
        diccionario_conteo[aero] = veces + 1
        #Y de una vez se compara este valor con el valor máximo
        #que llevamos hasta el momento
        if diccionario_conteo[aero] > mas_vuelos:
            mas_vuelos = diccionario_conteo[aero]
            aerolinea_con_mas_vuelos = aero

    #Al final, se retorna la aerolínea con más vuelos
    return aerolinea_con_mas_vuelos
```

PODEMOS ORGANIZAR LA INFORMACIÓN DE MUCHAS OTRAS FORMAS

- Con una lista de diccionarios
- Con un diccionario de listas de diccionarios
- ...

Lista de todos los vuelos



Llave	Valor
"vuelo"	"XE2679"
"aerolinea"	"XE"
"origen"	"EWR"
"destino"	"MYR"
"distancia"	"550"
"salida"	"1110"
"duracion"	"113"
"retraso"	"18"

Llave	Valor
"vuelo"	"UA63"
"aerolinea"	"UA"
"origen"	"MSP"
"destino"	"DEN"
"distancia"	"680"
"salida"	"1439"
"duracion"	"147"
"retraso"	"34"

PODEMOS ORGANIZAR LA INFORMACIÓN DE MUCHAS OTRAS FORMAS

- Con una lista de diccionarios
- Con un diccionario de listas de diccionarios
- ...

Diccionario principal de aerolíneas

