

NIVEL 3

DATOS MÁS COMPLEJOS – DICCIONARIOS
DE DICCIONARIOS Y ARCHIVOS



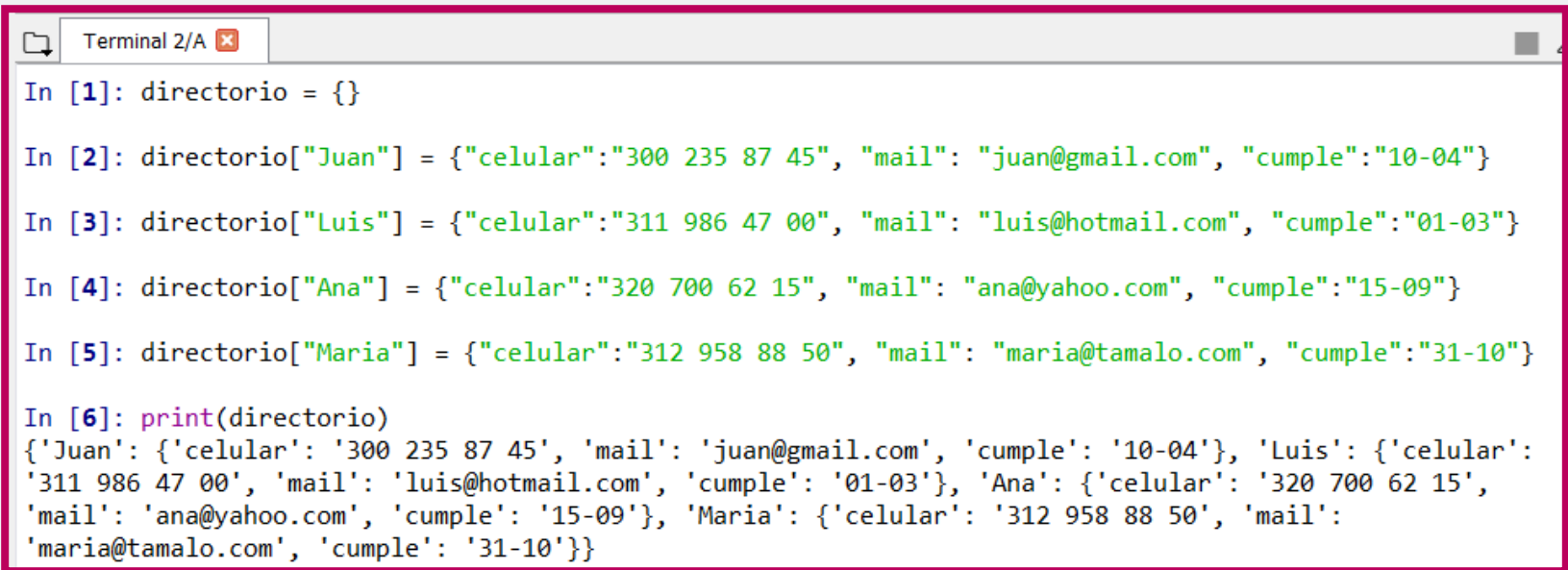
DICCIONARIOS DE DICCIONARIOS

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

```
In [10]: d = {}  
  
In [11]: d["Juan"] = "300 235 87 45"  
  
In [12]: d["Luis"] = "311 986 47 00"  
  
In [13]: d["Ana"] = "320 700 62 15"  
  
In [14]: d["María"] = "312 958 88 50"  
  
In [15]: print(d)  
{'Juan': '300 235 87 45', 'Luis': '311 986 47 00', 'Ana': '320 700 62 15', 'María': '312 958 88 50'}
```

DICCIONARIOS 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!



```
Terminal 2/A x
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"}

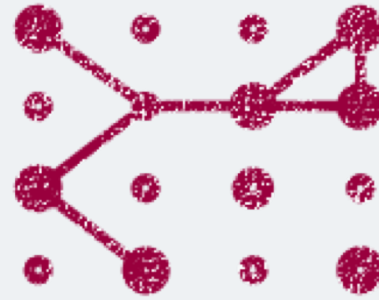
In [6]: print(directorio)
{'Juan': {'celular': '300 235 87 45', 'mail': 'juan@gmail.com', 'cumple': '10-04'}, 'Luis': {'celular': '311 986 47 00', 'mail': 'luis@hotmail.com', 'cumple': '01-03'}, 'Ana': {'celular': '320 700 62 15', 'mail': 'ana@yahoo.com', 'cumple': '15-09'}, 'Maria': {'celular': '312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

RECORRIENDO UN DICCIONARIO DE DICCIONARIOS



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'}
```

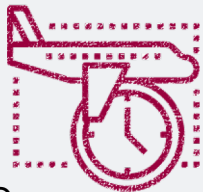


**¡LISTOS PARA UN DICCIONARIO
CON MAS INFORMACIÓN!**

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 lo siguiente:

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



VEAMOS ALGUNOS DATOS

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

USAREMOS UN DICCIONARIO DE DICCIONARIOS

Para manejar la información de los vuelos, usaremos un diccionario de diccionarios, donde la llave es el código del vuelo y el valor es a su vez un diccionario con el resto de la información, así:

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



```
Terminal 2/A
In [11]: vuelos = {}

In [12]: vuelos["XE2679"] = {"aerolinea": "XE", "origen": "EWR", "destino": "MYR",
"distancia": "550", "salida": "1110", "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["OO2880"] = {"aerolinea": "OO", "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 !!!



ARCHIVOS

Desde el punto de vista de la programación, los archivos son objetos en los que podemos escribir y/o leer información. El trabajo con archivos obliga a seguir siempre un protocolo de **3 pasos**:

1. Abrir el archivo
2. Leer o escribir la información que deseamos
3. Cerrar el archivo



PASO 1: ABRIR EL ARCHIVO

- Se hace con la instrucción **open**
- Debemos indicar su nombre incluyendo la ruta (relativa o absoluta) y el modo de trabajo:
 - ✓ **Lectura (r)**: es posible leer información del archivo, pero no modificarla ni añadir nueva información
 - ✓ **Escritura (w)**: solo es posible escribir información en el archivo. La apertura de un archivo en modo escritura borra todo el contenido previo del mismo
 - ✓ **Adición(a)**: permite añadir nueva información al archivo, conservando la información existente

PASOS 2 Y 3

2: Leer o escribir la información que deseamos

- Para leer usamos la instrucción **readline**
- Para escribir usamos la instrucción **write**



3: cerrar el archivo

- Se hace con la instrucción **close**
- Es MUY importante cerrar todos los archivos tan pronto hayamos acabado de trabajar con ellos, especialmente si los hemos modificado
- Si no cerramos el archivo, es posible que los cambios que hayamos efectuado se pierdan o, peor aún, que el archivo se dañe

EJEMPLO – LECTURA ARCHIVO VUELOS

Este método lee la información de los vuelos de un archivo cuyo nombre entra por parámetro y retorna el diccionario de diccionarios con la información

Se crea el diccionario que será retornado al final de la función

```
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
```

EJEMPLO – LECTURA ARCHIVO VUELOS

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`

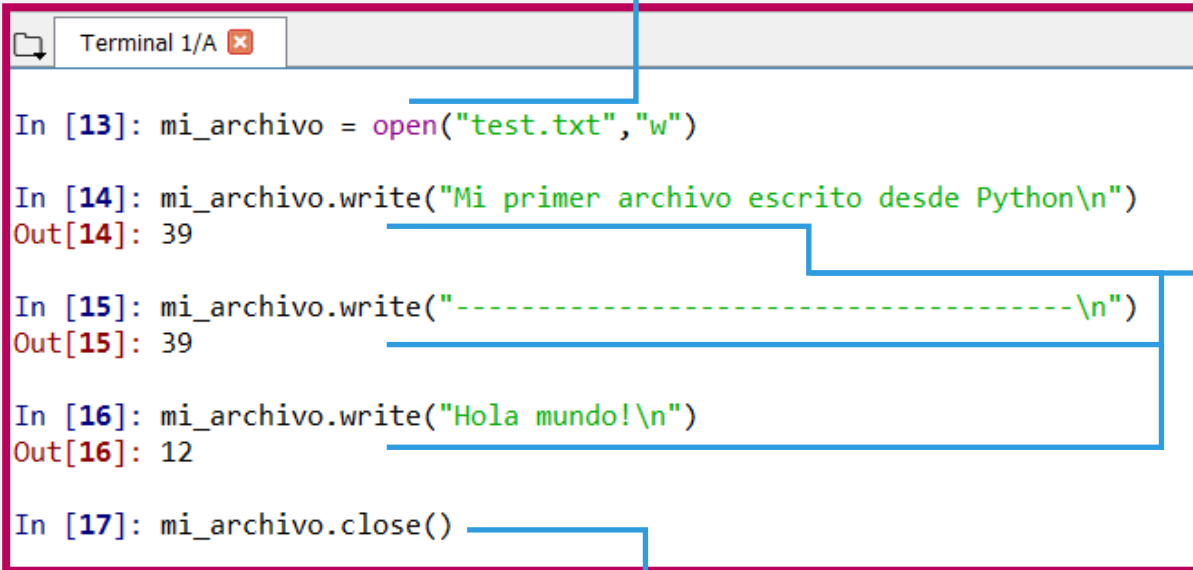
```
EjemploCargarVuelos.py x
1 def cargar_vuelos(nombre_archivo: str) -> dict:
2     vuelos = {}
3     archivo = open(nombre_archivo, "r")
4     titulos = archivo.readline()
5     print(titulos)
6
7     linea = archivo.readline()
8     while len(linea) > 0:
9         datos = linea.split(";")
10        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
```

Resultado de la ejecución

```
Terminal 3/A ✕  
  
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',
```

EJEMPLO – ESCRITURA ARCHIVO

Paso 1: Se abre el archivo con la instrucción `open` y en modo de escritura (`w`)

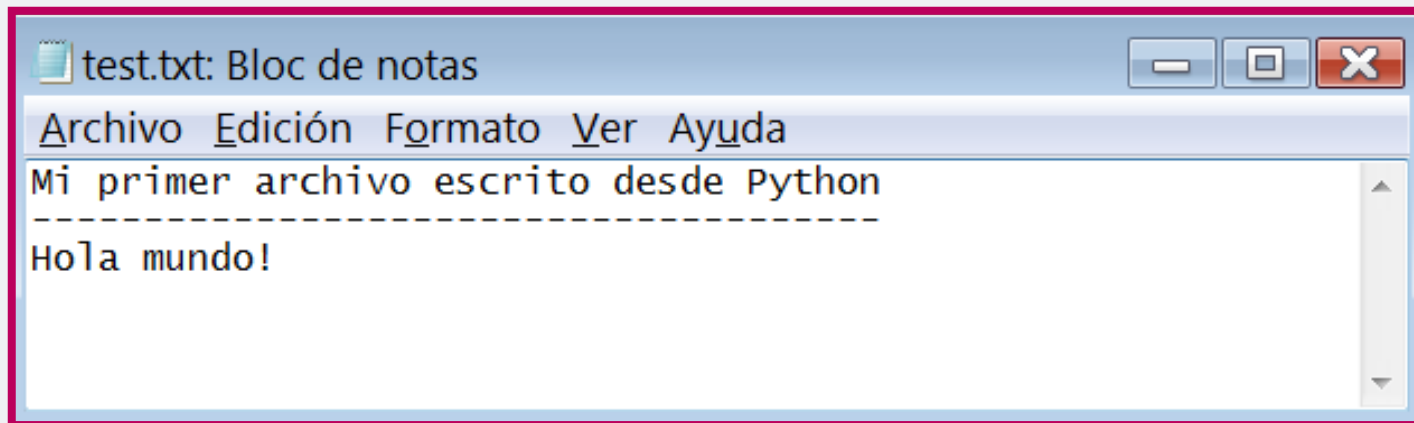
A terminal window titled 'Terminal 1/A' with a red close button. It contains five lines of Python code and their corresponding outputs. Blue lines with dots at the end point from the text blocks to specific parts of the code: from 'open' to the first line, from 'write' to the second and third lines, and from 'close' to the fifth line.

```
In [13]: mi_archivo = open("test.txt", "w")
In [14]: mi_archivo.write("Mi primer archivo escrito desde Python\n")
Out[14]: 39
In [15]: mi_archivo.write("-----\n")
Out[15]: 39
In [16]: mi_archivo.write("Hola mundo!\n")
Out[16]: 12
In [17]: mi_archivo.close()
```

Paso 2: Se escribe la información del archivo línea por línea con la instrucción `write`

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

Resultado de la ejecución



Un nuevo archivo llamado **test.txt** con la información que escribimos

