

# Fundamentos de Programación

## Curso 23-24. 2ª Convocatoria. PYTHON

### Contexto

La empresa Festi World S.A. pone a su disposición un dataset con algunos datos de los festivales de música que organiza. La compañía necesita que se exploten estos datos y, para ello, le solicita una serie de funciones para extraer información relevante. Los datos que le suministra en un fichero CSV son los siguientes:

- **Nombre del festival.**
- **Fecha de comienzo del festival.**
- **Fecha de finalización del festival.**
- **Estado del festival**, que puede tomar los valores **PLANIFICADO**, **CELEBRADO** o **CANCELADO**, en función del estado actual del festival.
- **Precio de la entrada**, que puede tomar valores decimales.
- **Entradas vendidas**: número de entradas vendidas.
- **Artistas**, que es un listado de artistas que actúan en el festival y de ellos tenemos:
  - **Nombre**
  - **Hora de actuación**
  - **Caché**: cantidad en miles de euros que cobra el artista por actuar.
- **Top**, que toma como valor "sí" o "no" en función de si el festival es considerado top mundial o no.

Una línea del fichero que nos proporcionan tiene el siguiente aspecto:

```
Tomorrowland,2024-07-19,2024-07-21,PLANIFICADO,280.99,70000,David  
Guetta_20:00_700-Tiësto_21:30_750-Calvin Harris_23:00_800,sí
```

e indica que:

- El festival Tomorrowland se celebra desde el 19 de julio de 2024 al 21 de julio de 2024, y su estado actual es PLANIFICADO. La entrada tiene un precio de 280.99€ y se han vendido un total de 70.000 entradas. En este festival actúan David Guetta a las 20:00 cobrando 700 mil euros, después actúa Tiësto a las 21:00 cobrando 750 mil, y Calvin Harris a las 23:00 cobrando 800 mil euros. Por último, este festival sí está considerado como uno de los tops.

Para facilitar los encargos de la empresa se han dividido sus peticiones en diferentes ejercicios, que tiene a continuación.

### Ejercicios

Para la realización de los ejercicios se usarán las siguientes definiciones de **namedtuple** y su uso será obligatorio. Además puede encontrar los tipos asociados a cada atributo:

```
from typing import NamedTuple, List, Dict, Tuple, Optional, Set

Artista = NamedTuple("Artista",
                     [("nombre", str),
                      ("hora_comienzo", time),
                      ("cache", int)])

Festival = NamedTuple("Festival",
                      [("nombre", str),
                       ("fecha_comienzo", date),
                       ("fecha_fin", date),
                       ("estado", str),
                       ("precio", float),
                       ("entradas_vendidas", int),
                       ("artistas", List[Artista]),
                       ("top", bool)
                      ])
```

Como ejemplo, para el festival anterior obtendremos la siguiente tupla. Fíjese con detalle en el tipo de cada uno de los campos:

```
Festival(nombre='Tomorrowland', fecha_comienzo=datetime.date(2024, 7, 19),
fecha_fin=datetime.date(2024, 7, 21), estado='PLANIFICADO', precio=280.99,
entradas_vendidas=70000, artistas=[Artista(nombre='David Guetta',
hora_comienzo=datetime.time(20, 0), cache=700), Artista(nombre='Tiësto',
hora_comienzo=datetime.time(21, 30), cache=750), Artista(nombre='Calvin Harris',
hora_comienzo=datetime.time(23, 0), cache=800)], top=True)
```

Implemente las siguientes funciones en un módulo `festivales.py`:

1. `lee_festivales`: recibe una cadena de texto con la ruta de un fichero `csv`, y devuelve una lista de tuplas `Festival` con la información contenida en el fichero. La lista de festivales devuelta debe estar ordenada de manera ascendente por fecha de comienzo de los mismos. **(1.25 punto)**

```
def lee_festivales (archivo:str)->List[Festival]
```

2. `total_facturado`: esta función devuelve el importe total facturado de los festivales que se han celebrado entre dos fechas dadas. La función recibe una lista de tuplas de tipo `Festival` y dos fechas, cuyos valores por defecto son `None`. La función devuelve un número real con el total facturado por los festivales celebrados entre las dos fechas dadas. Si la fecha inicial es `None` se hace el cálculo sin limitar la fecha mínima de los festivales. Si la fecha final es `None` se hace el cálculo sin limitar la fecha máxima de los festivales. Para calcular el total facturado por festival hay que multiplicar el número de entradas por el precio de la entrada del festival. **Nota:** tenga en cuenta que la función debe tomar la facturación de los festivales con estado `celebrado` en el rango de fechas, es decir, solo se tendrán en cuenta aquellos festivales que empiezan y acaban dentro del rango de fechas. **(1.25 punto)**

```
def total_facturado(festivales:List[Festival], fecha_ini:Optional[date]=None,
fecha_fin:Optional[date]=None)->float
```

3. **artista\_top**: recibe una lista de tuplas de tipo **Festival** y devuelve una tupla compuesta por un número entero y una cadena de texto, que representan el número de festivales y el nombre del artista que haya participado en más festivales que finalmente se han celebrado, respectivamente. **(1.75 puntos)**

```
def artista_top(festivales: List[Festival]) -> Tuple[int, str]
```

4. **mes\_mayor\_beneficio\_medio**: recibe una lista de tuplas de tipo **Festival** y devuelve una cadena de texto que será el nombre del mes, en español, de aquel que haya obtenido un mayor beneficio medio. Es decir, cada festival tiene un beneficio que se calcula a partir de las entradas vendidas menos el caché de los artistas. Pues esta función debe calcular el beneficio medio que se ha obtenido cada mes y devolver aquel cuyo beneficio haya sido el mayor. **Nota**: Si hubiera algún festival que se celebra entre dos meses, se imputará al mes en el que comienza. Por ejemplo, un festival que comience el 30 de junio y acabe el 4 de julio será imputado al mes de junio. **(2.25 puntos)**

```
def mes_mayor_beneficio_medio(festivales: List[Festival]) -> str
```

5. **artistas\_comunes**: recibe una lista de tuplas de tipo **Festival** y dos cadenas de texto **festi1** y **festi2**, y devuelve una lista con los nombres de aquellos artistas que se repitan entre **festi1** y **festi2**. **Nota**: se considera que no hay festivales repeidos. **(1.75 puntos)**

```
def artistas_comunes(festivales: List[Festival], festi1: str, festi2:str) ->
List[str]
```

6. **festivales\_top\_mejor\_ratio**: Cada festival tiene una duración de entre 2 y 8 días. Implemente una función que, recibiendo una lista de tuplas de tipo **Festival**, y un número **n** cuyo valor por defecto será 3, devuelva un diccionario en el que las claves son las duraciones de los festivales, y los valores listas con los nombres de los **n** festivales de más calidad (ordenados de más a menos calidad). La calidad de un festival viene dada por el ratio entre entradas vendidas y número de artistas participantes en el festival. Cuanto más alto es este ratio, más calidad tiene el festival. **(1.75 punto)**

```
def festivales_top_calidad_por_duracion(festivales: List[Festival], n: int=3) ->
Dict[int, List[str]]
```

Pruebe las funciones implementadas en un módulo **festivales\_test.py**. Use una función de test, con los parámetros adecuados para cada función a probar. Las funciones que implemente deben comprobar las diferentes casuísticas que pudieran darse en cada ejercicio. Se recomienda que lo vaya haciendo a medida

que vaya resolviendo los distintos apartados. Cada test puntuará con 0.25 para cada ejercicio, y para obtener la calificación la función deberá poder ejecutarse y los resultados deberán coincidir con los que se indican a continuación:

### Salida del fichero `festivales_test.py`

```
Test Ej1: lee_festivales
Test lee_festivales
Total festivales: 21
Los tres primeros:
    Festival(nombre='Ultra Music Festival',
fecha_comienzo=datetime.date(2024, 3, 29), fecha_fin=datetime.date(2024, 3, 31),
estado='CANCELADO', precio=300.89, entradas_vendidas=50000, artistas=
[Artista(nombre='Armin van Buuren', hora_comienzo=datetime.time(20, 0),
cache=700), Artista(nombre='Skrillex', hora_comienzo=datetime.time(21, 30),
cache=750), Artista(nombre='Steve Aoki', hora_comienzo=datetime.time(23, 0),
cache=800), Artista(nombre='Calvin Harris', hora_comienzo=datetime.time(0, 30),
cache=850), Artista(nombre='Martin Garrix', hora_comienzo=datetime.time(2, 0),
cache=900)], top=True)
    Festival(nombre='Coachella', fecha_comienzo=datetime.date(2024, 4, 12),
fecha_fin=datetime.date(2024, 4, 14), estado='CELEBRADO', precio=400.99,
entradas_vendidas=50000, artistas=[Artista(nombre='The Strokes',
hora_comienzo=datetime.time(20, 0), cache=500), Artista(nombre='Beyoncé',
hora_comienzo=datetime.time(21, 0), cache=1000), Artista(nombre='Childish
Gambino', hora_comienzo=datetime.time(22, 30), cache=800), Artista(nombre='Post
Malone', hora_comienzo=datetime.time(0, 0), cache=900), Artista(nombre='Billie
Eilish', hora_comienzo=datetime.time(1, 30), cache=850), Artista(nombre='Kanye
West', hora_comienzo=datetime.time(3, 0), cache=950)], top=True)
    Festival(nombre='Viña Rock', fecha_comienzo=datetime.date(2024, 4, 28),
fecha_fin=datetime.date(2024, 5, 1), estado='CELEBRADO', precio=90.99,
entradas_vendidas=17000, artistas=[Artista(nombre='Residente',
hora_comienzo=datetime.time(21, 0), cache=300), Artista(nombre='Natos y Waor',
hora_comienzo=datetime.time(22, 30), cache=350), Artista(nombre='Kase.O',
hora_comienzo=datetime.time(0, 0), cache=400), Artista(nombre='Zetazen',
hora_comienzo=datetime.time(1, 30), cache=200), Artista(nombre='Skrillex',
hora_comienzo=datetime.time(3, 30), cache=600)], top=False)
    Festival(nombre='Electric Daisy Carnival',
fecha_comienzo=datetime.date(2024, 5, 17), fecha_fin=datetime.date(2024, 5, 19),
estado='CELEBRADO', precio=350.0, entradas_vendidas=55000, artistas=
[Artista(nombre='Marshmello', hora_comienzo=datetime.time(20, 0), cache=750),
Artista(nombre='Zedd', hora_comienzo=datetime.time(21, 30), cache=700),
Artista(nombre='Diplo', hora_comienzo=datetime.time(23, 0), cache=650),
Artista(nombre='Kygo', hora_comienzo=datetime.time(0, 30), cache=800),
Artista(nombre='Deadmau5', hora_comienzo=datetime.time(2, 0), cache=750)],
top=True)
=====

Test Ej2: total_facturado
Entre None y None el total es: 75416650.0
Entre None y 2024-06-15 el total es: 47771650.0
Entre 2024-06-15 y None el total es: 27645000.0
Entre 2024-06-01 y 2024-06-15 el total es: 6925320.0
```

```
Entre 2024-06-01 y 2024-06-23 el total es: 13540320.0
```

```
=====
```

```
Test Ej3: artista_top
```

```
El artista que ha actuado en más festivales es The Strokes, con 4 festivales.
```

```
=====
```

```
Test Ej4: mes_beneficio_medio_mas_rentable
```

```
El mes con más beneficio es Mayo.
```

```
=====
```

```
Test 5: artistas_comunes
```

```
Los artistas comunes entre Creamfields y Tomorrowland son ['David Guetta']
```

```
Los artistas comunes entre Primavera Sound y Coachella son ['The Strokes', 'Billie Eilish']
```

```
Los artistas comunes entre Iconica Fest y Primavera Sound son []
```

```
=====
```

```
Test 6: festivales_top_mejor_ratio
```

```
Los mejores 1 festivales por número de días son:
```

```
2: ['Tomorrowland']
```

```
4: ['Glastonbury']
```

```
7: ['Roskilde Festival']
```

```
3: ['Creamfields']
```

```
6: ['Sziget Festival']
```

```
8: ['Burning Man']
```

```
Los mejores 3 festivales por número de días son:
```

```
2: ['Tomorrowland', 'Lollapalooza', 'Electric Daisy Carnival']
```

```
4: ['Glastonbury', 'Primavera Sound']
```

```
7: ['Roskilde Festival']
```

```
3: ['Creamfields', 'Festival Internacional de Benicàssim', 'Rock in Rio']
```

```
6: ['Sziget Festival']
```

```
8: ['Burning Man']
```

```
=====
```