

Programación – Ejercicio de Diccionarios

Trenes

En la ciudad de Pythonia todos los trenes realizan el mismo recorrido, pasando por cada una de las estaciones de la *Línea P*, cuyo orden se encuentra en la lista `linea_P`.

```
lineaP = [
    'Estacion Central' ,
    'Puente Sal y Santo' ,
    'Pytronato' ,
    'La Pysterna' ,
    'Pynstein' ,
    'Estadio Nacional' ,
    'Pio-Pio' ,
    'Plaza Python Alto' # ...
]
```

El tiempo entre una estación y otra depende del tipo de tren. Los trenes *Expreso* tardan 2 minutos en ir desde una estación a la siguiente, los trenes *Normal* demoran 5 minutos, y los trenes de tipo *Corto* tardan 7 minutos. Suponga que el tiempo que el tren se detiene en las estaciones es despreciable, por lo que no será considerado para la solución de este problema. Los tipos de trenes se encuentran en el diccionario `tipos`, en el que la llave es un *string* con el tipo de tren y el valor es una lista de *strings* con los identificadores únicos de los trenes que pertenecen a ese tipo.

```
tipos = {
    'Expreso': ['E07', 'F07', 'F08', ...],
    'Normal' : ['G00', 'H00', ...],
    'Corto'  : ['B00', 'C01', ...]
}
```

Finalmente, el diccionario `trenes` contiene como llave el identificador único del tren y como valor una lista `['HH', 'MM']` que indica el horario de salida desde la primera estación de la línea, la que es siempre la Estación Central de Pythonia.

```
trenes = {
    'E07': ['09', '03'], 'F07': ['10', '14'], 'H00': ['09', '23'], 'B00': ['08', '00'],
    'G00': ['08', '46'], 'C01': ['13', '59'], 'F08': ['11', '02'] # ...
}
```

Desarrolle la función `proximos_trenes(estaciones, tipos, trenes, estacion, hora)` que recibe como parámetro las colecciones descritas anteriormente, además del *string* `estacion` que corresponde al nombre de la estación en la que se encuentra el pasajero, y la lista `hora` que contiene la hora actual en formato `['HH', 'MM']`. La función debe retornar un diccionario que cuyas llaves son los identificadores de los trenes que aún no han pasado por `estacion` en `hora`. El valor asociado a cada llave corresponde a los minutos que faltan para el arribo del tren a `estacion`. En caso de que no existan trenes pendientes de pasar, se debe retornar un diccionario vacío.

Ejemplos:

```
>>> print(proximos_trenes(estaciones, tipos, trenes, 'Pytronato', [9,6]))
{'E07': 1, 'F07': 72, 'H00': 27, 'C01': 307, 'F08': 120}
>>> print(proximos_trenes(estaciones, tipos, trenes, 'Pio-Pio', [11,0]))
{'C01': 221, 'F08': 14}
>>> print(proximos_trenes(estaciones, tipos, trenes, 'Puente Sal y Santo', [15,00]))
{}
```