



Disponemos de datos sobre la inspección técnica de vehículos (ITV) que deben pasar de forma periódica todos los vehículos en España. Para cada inspección se tiene la siguiente información:

- **fecha de la inspección:** fecha en la que se ha realizado la inspección, de tipo date.
- **estación:** nombre de la estación donde se ha realizado la inspección, de tipo str.
- **número de inspección:** número de orden de la inspección, de tipo int. El valor 1 indica que se trata de la primera inspección. Si esta es desfavorable, el vehículo deberá hacer una segunda inspección; si esta también es desfavorable, deberá realizar una tercera, y así sucesivamente.
- **fecha límite:** fecha que tenía el vehículo como límite para realizar la inspección, de tipo date.
- **matrícula:** matrícula del vehículo, de tipo str.
- **tipo:** tipo del vehículo, de tipo str. En este ejercicio trabajaremos con tres tipos de vehículos: 'Turismo gasolina', 'Turismo diésel' y 'Turismo eléctrico'.
- **fecha de matriculación:** fecha de la primera matriculación del vehículo, de tipo date.
- **favorable:** indica si la inspección ha sido favorable (True) o desfavorable (False), de tipo bool.

Por ejemplo, la siguiente línea del fichero:

```
7/6/2022;Sevilla-El Pino;1;14/6/2022;3595KLW;Turismo gasolina;14/6/2018;S
```

indica que el turismo de gasolina con matrícula 3595KLW pasó favorablemente la ITV el 7 de junio de 2022 en primera inspección en la estación 'Sevilla-El Pino'. Lo hizo una semana antes de la fecha límite, que era el 14 de junio de 2022, justo a los 4 años de su matriculación. Observe que el valor 'S' indica que la inspección ha sido favorable, mientras que 'N' indicaría que ha sido desfavorable.

Para almacenar los datos de una inspección se usará obligatoriamente la siguiente namedtuple:

```
Inspeccion = namedtuple('Inspeccion', 'fecha_inspeccion, estacion, numero, fecha_limite, matricula, tipo, fecha_matriculacion, favorable')
```

Cree un módulo **inspecciones.py** e implemente en él las funciones que se piden. Cree un módulo **inspecciones\_test.py** y defina una función de test para cada función solicitada. Puede definir funciones auxiliares cuando lo considere necesario:

1. **lee\_inspecciones:** recibe una cadena de caracteres con el nombre de un fichero en formato CSV codificado en UTF-8, y devuelve una lista de tuplas de tipo `Inspeccion` conteniendo todos los datos almacenados en el fichero. Utilice la función `datetime.strptime` con el formato `"%d/%m/%Y"` para convertir las fechas. (1 punto)
2. **vehiculos\_mas\_antiguos:** recibe una lista de tuplas de tipo `Inspeccion`, un año y un número entero `n`, y devuelve una lista con las matrículas de los `n` vehículos con mayor antigüedad que han tenido una inspección favorable en el año dado, del más antiguo al más reciente. (1 punto)
3. **proximas\_inspecciones:** recibe una lista de tuplas de tipo `Inspeccion`, y devuelve una lista de tuplas (matrícula, fecha de la próxima inspección) con la matrícula y la fecha de la próxima inspección de todos los vehículos con inspección favorable. La fecha de la próxima inspección se calcula de la siguiente forma: si la diferencia entre el año actual y el de matriculación del vehículo es de 10 años o menos, la siguiente inspección será a los dos años de la fecha de inspección; si la diferencia es mayor de 10 años, la siguiente inspección será al año de la fecha de inspección. Suponga años de 365 días de duración. Notas: 1) para obtener la fecha del día de hoy, utilice la función `date.today()`; 2) para sumar un número de días `n` a una fecha `f`, utilice la expresión `f + timedelta(n)`; 3) La lista contiene una inspección por vehículo. (1,5 puntos)

4. **estacion\_mayor\_porcentaje\_inspecciones\_favorables:** recibe una lista de tuplas de tipo Inspeccion y un conjunto de estaciones, y devuelve el nombre de la estación del conjunto de entrada que tiene el mayor porcentaje de inspecciones favorables respecto al total de inspecciones realizadas en esa estación, junto con el valor de este porcentaje. (2 puntos)
5. **tipos\_de\_vehiculos\_mas\_inspeccionados:** recibe una lista de tuplas de tipo Inspeccion y dos valores de tipo date f1 y f2, que pueden tomar el valor None, y devuelve un diccionario que relaciona cada estacion con el tipo de vehículo que más veces se ha inspeccionado en esa estación, para las inspecciones realizadas entre las fechas f1 y f2 (ambas inclusive). Si f1 es None, se tendrán en cuenta todas las inspecciones anteriores a f2 (inclusive). Si f2 es None, se tendrán en cuenta todas las inspecciones posteriores a f1 (inclusive). Si ambas son None, se tendrán en cuenta todas las inspecciones. (2 puntos)
6. **incrementos\_recaudacion\_estacion:** recibe una lista de tuplas de tipo Inspeccion y una estación, y devuelve una lista con los incrementos de las recaudaciones de la estación en cada año respecto al anterior. El importe de cada inspección se calcula de la siguiente forma: se parte de un importe inicial que depende del tipo del vehículo (Turismo gasolina: 24.05 €; Turismo diésel: 28.27 €; Turismo eléctrico: 20.60 €). A este importe se le aplica un descuento del 10 % si la inspección se ha realizado antes del plazo límite, y un incremento del 12 % si se ha realizado con más de 30 días de retraso sobre el plazo límite. Luego se añaden 9.11 € si es la tercera inspección o superior. Al importe resultante se le suma el 21 % de IVA. Finalmente, se suman las tasas de tráfico, que ascienden a 4.18 €. Notas: 1) es posible que en algún año no haya habido ninguna inspección; 2) para sumar un número de días n a una fecha f, utilice la expresión f + timedelta(n). (2,5 puntos)