

RETO PROGRAMACIÓN PYTHON

¿Te atreves??



? DE QUÉ VA ESTO...

Adjunto a este documento encontrarás un fichero CSV que contiene un listado con 5000 posiciones de varios vehículos. Cada una de las líneas detalla las coordenadas geográficas, fecha y distancia de esa posición/matricula.

Cada fila incluye los siguientes campos:

Matricula,Latitud,Longitud,Distance,Pos_date
E0047EE446A,41.12162,1.192766,0.0087,1701388800000

En el ejemplo:

- *Matricula* del vehículo: E0047EE446A
- *Latitud,Longitud* 41.12162,1.192766 (decimal)
- *Distance*: Distancia recorrida con respecto a la posición anterior: 0.0087metros.
- *Pos_date*: Fecha de la posición (timestamp en milisegundos) 1701388800000 -> 01/12/2023 01:00:47

QUÉ ESPERAMOS DE TÍ

Lee con atención los diferentes casos del reto, cada caso sube en dificultad! Puedes resolver todos ellos o solo aquellos en los que te sientas más cómodo.

El próximo 4 de marzo, repasaremos juntos el resultado en lo que llamaremos 'la defensa del reto'.

Acude a "la defensa" con las fuentes de código de la aplicación y todo lo que consideres necesario.

NO es estrictamente necesario que la aplicación generada sea 100% funcional, SI es importante que seas capaz de explicar la arquitectura propuesta, y los problemas que has afrontado a la hora de montarla.

Disfruta del reto y no te agobies!!

Ah! Si subes el código a un repositorio de github mucho mejor.

Y AHORA EL RETO

- **CASO 1. Empezamos por algo fácil:** Lee el fichero CSV e imprime cada línea.
- **CASO 2.** Lee el fichero CSV, convierte cada línea a un JSON e imprímelo.
- **CASO 3. Subimos de dificultad:** En cada fila se incluye la distancia que el vehículo ha recorrido con respecto a su posición anterior. Calcula el sumatorio de las distancias de cada vehículo, usando el campo "distancia" a partir del formato json. Imprime todos los resultados por consola.
- **CASO 4. Esto se complica:** Ahora calcula el sumatorio de distancias de cada matrícula, usando las coordenadas geográficas (el cálculo debe hacerse a partir del formato json) e imprime todos los resultados por consola.
- **CASO 5. ¿Listo?** Escribe un nuevo fichero de texto donde aparezca la fecha de la última posición de cada vehículo en formato (dd/mm/YYYY HH:MM:SS) a partir del campo "pos_date". Recuerda, pos_date es una fecha en Tiempo POSIX. Escribe el fichero ordenado por fecha, de la más reciente a la más antigua.
- **CASO 6. Ahora** Crea un API REST con un método GET del estilo (<http://localhost/XXX>) dónde XXX es la matrícula de un vehículo. Su resultado debe leerse del fichero del punto anterior y devolver la fecha de última posición de la matrícula indicada
- **CASO 7. ¿Te atreves** a levantar un **contenedor** Docker con una base de datos e incluir en ella los datos del fichero del punto 5?
- **CASO 8. ¡¡La última!!** Haz que el API REST del punto 6 lea de la base de datos del punto 7 en vez del fichero del punto 5.