



Ejercicio Skyscanner

En este ejercicio vas a trabajar con la base de datos de [Skyscanner](#), uno de los mayores metabuscadores de vuelos en todo el mundo.

El objetivo de este proyecto es dar con el mejor precio para un vuelo que tenemos pensado hacer en algún momento. Tendrás que monitorizar el vuelo dentro de un intervalo de fechas concretas, por ejemplo, Madrid – Montevideo, con salida en junio y vuelta en julio. Además de atacar a la API y formatear los datos, habrá que desarrollar un sistema de alertas para que te lleguen al correo las novedades de esos vuelos en tales fechas, con el precio incluido.

Con este programa corriendo todos los días podrás encontrar el momento perfecto para comprar tu vuelo deseado, y lo mejor de todo, sin necesidad de tener que meternos en la página de Skyscanner y consultar los vuelos para las fechas escogidas. Tendremos la información bien procesada y notificada mediante correo electrónico.

Se pide:

1. **OBLIGATORIO** trabajar con un IDE como Pycharm, VSCode o Spyder.
2. API de Skyscanner. Realmente atacarás a la API a través de RapidAPI, que es una empresa a través de la cual puedes consultar diferentes APIs de varios proveedores.

Registro en RapidAPI:

<https://rapidapi.com/skyscanner/api/skyscanner-flight-search?endpoint=5aa1edd5e4b06ec3937b23f0>

Documentación de la API de Skyscanner:

<https://skyscanner.github.io/slate/#api-documentation>

Te tendrás que registrar en RapidAPI, pero en ningún momento introducir la tarjeta de crédito. Tienes gratuitas 50 peticiones al minuto.

El enunciado está cerrado a un único destino en unas fechas concretas, pero podrás probar con varios destinos en diferentes llamadas a la API o incluso usar las palabras clave “anywhere” o “anytime” para realizar una búsqueda de “Madrid a cualquier lugar” o de “Madrid a Montevideo en la fecha que sea”. Consulta la documentación. Te recomiendo que vayas a la [página de Skyscanner](#) para que veas cómo funciona el buscador.

3. Trabajar con JSON. Recibirás un JSON como respuesta a la API. Monta una tabla con los vuelos, ida, vuelta, precios y fechas, y guárdalo en un fichero .csv con la fecha de extracción de los datos en el nombre del fichero
4. Crea un fichero main.py y funciones.py para encapsular tu código.
5. Si se te ocurre alguna mejora o idea para el proyecto, siéntete libre para implementarla



Bonus track

Posibles mejoras (Opcional):

1. Tener monitorizados varios destinos al mismo tiempo.
2. Dejar corriendo el script de Python en un planificador de tareas y que se actualice diariamente (planificador de Windows, crontab de Unix o servicio de hosting).
3. Crear una SQLite con los datos recibidos e ir actualizando los vuelos.
4. Recibir únicamente mails cuando de verdad sea una alerta, es decir, si atacamos a la API y el precio es el mismo que la última llamada (consulta a la BD), no mandará mail.
5. Puedes recibir esa tabla por correo. Para ello te tendrás que apoyar en los paquetes “email” y “smtplib”.