

Sistema de monitorización y detección de anomalías en Wallapop

Arturo Francés y Marcos Fraile

1. RESUMEN:

En este informe se describe el desarrollo e implementación de scripts, archivos de configuración y logs generados para nuestro pipeline.

<https://github.com/marcosf22/wallapop>

2. FUNCIONAMIENTO DEL SISTEMA:

En este caso, para el funcionamiento de nuestro pipeline, debemos realizar una serie de programas en python y archivos de configuración para conseguir que todo el conjunto de estos, nos den los resultados esperados. Para describir el funcionamiento individual, vamos a describir cada uno de los módulos que tenemos en github.

Para empezar, tenemos todos los archivos que involucran al poller. Primero, tenemos el poller.py. Este programa busca en wallapop, en la categoría deseada y producto elegido, en este caso relojes, y filtra los resultados de la búsqueda mediante palabras clave que usa este programa, para encontrar posibles estafas. En nuestro caso, buscamos relojes de marcas como Rolex, Ten Hauer, Omega etc. Este archivo genera un JSON donde se almacenan todos los datos reunidos y filtrados para detectar posibles estafas.

Luego tenemos el contenido de la carpeta ingestion. En este caso, hemos hecho el volcado de los logs de los JSONs generados mediante la opción A, que era de forma manual. Para ello, hemos usado los archivos creados en la práctica 2 y hemos hecho el script bulk_ ingest.py. Este script, permite volcar nuestros datos generados con el poller en elasticsearch.

Una vez hechos estos scripts y generados los logs correspondientes. Hemos configurado nuestro elastalert y hemos hecho las visualizaciones correspondientes en elastic.

3. VISUALIZACIONES EN ELASTIC

Como hemos dicho en el anterior apartado, el poller se encarga de generar unos JSONs con toda la información recopilada sobre los distintos productos encontrados que se vuelcan en elastic, donde podemos realizar distintos tipos de visualizaciones para ver la información recogida de forma ordenada y así podamos sacar conclusiones que apoyaremos con la interfaz gráfica que hemos creado.

4. INTERFAZ GRÁFICA:

Como añadido, hemos hecho una interfaz web de las publicaciones que hemos recopilado en nuestros JSONs. En esta interfaz podemos filtrar por el score de cosas que hemos considerado importantes a la hora de encontrar una estafa. Y además tienes un buscador por ciudad donde se vende, marca deseada, rango de precios, y filtrar por posible fraude, es decir robo, réplica etc.

The screenshot shows a web-based dashboard titled "Radar de Ofertas - Visualizador". On the left, there's a sidebar titled "Panel de Control" with a file list showing "wallapop_watches_202512 11.json" and a timestamp "Última actualización archivo: 16:27:22". It also includes a checkbox for "Auto-refrescar (5s)". Below this are sections for "Filtros" (Filters) with sliders for "Precio Mínimo (€)" (0), "Precio Máximo (€)" (20000), and "Nivel de Riesgo Mínimo" (0). On the right, a product card for an "Audemars Piguet Royal Oak Chronograph Oro" is displayed. The card shows the product name, price "1650.0 EUR", a yellow warning box "SOSPECHOSO: 25/100", and the brand "Marca: Audemars Piguet". It also includes a "Ver JSON" button and a note "Alerta: Low price (Index: 0.21)". A small image of the watch in its box is shown above the card.

5. DESARROLLO Y TRABAJO CON LA IA:

Para este trabajo, hemos seguido con nuestra forma de trabajar de siempre. La cual consiste en trabajar de forma paralela en los archivos correspondientes que había que desarrollar el trabajo, para así, tener varios puntos de vista y soluciones distintas de cada script hecho. Una vez que ya tuvimos lo necesario para mandar los logs a Elastic para verlos en Kibana, lo hicimos desde el PC de Marcos. Pero, aunque solo una persona tuviera las visualizaciones, seguimos trabajando en paralelo con la IA, para seguir completando el trabajo con 2 puntos de vista distintos de la IA.

Para este trabajo hemos usado solamente Gemini