

4.4 Importación de CSV, testing en Dev tools y Rest API web.

Todas las respuestas se harán en color azul para facilitar la corrección de las actividades

En esta actividad crearás un índice nuevo desde un *dataset* de Kaggle, <https://www.kaggle.com/datasets/saurabhbhagchi/books-dataset/>

1. Importación.

Para importar el fichero (JSON, CSV, etc) puedes ir a la opción:

- ⑩ Kibana Home Page → Upload a file
- ⑩ Menú → Analytics → Machine Learning → Importa Data

2. Análisis de los datos importados. Importante, ver el Mapping.

Creado el índice deberás analizar sus campos desde la sección:

- ⑩ Menu → Analytics → Discover → Data → Index Management → (índice) → Mappings
- ⑩ Menu → Management → Dev Tools

Ya conoces todos los campos del índice. Verás que algunos campos son tipo keyword, otros son tipo text, otros tipo timestamp.

(1,5 puntos) 2.1. Contesta a las siguientes preguntas.

1. ¿Se corresponden con las cabeceras de tu CSV?
2. ¿Te resulta cómodo trabajar con estos nombres de campos? Justifica tu respuesta.
3. ¿Qué campo se ha cogido como Timestamp? ¿Es correcto el campo que tiene asignado el tipo *timestamp*?

3. Modificación del Mapping y del Ingest Pipeline para la importación de datos.

Modifica tu Mapping y el Ingest Pipeline para que se ajuste a las necesidades de tus datos. Se recomienda modificar el los tipos de datos y los nombres de los campos, tanto en el Mapping como en el Ingest Pipeline. Una vez tengas tu Mapping y el Ingest Pipeline, importa tu CSV.

Recuerda que los tipos:

Timestamp.- en contextos como bases de datos, Elasticsearch, y otros sistemas de almacenamiento de datos, se utiliza para almacenar marcas de tiempo. Estas marcas indican cuándo ocurrieron eventos específicos, permitiendo a los sistemas llevar un registro cronológico de acciones o cambios. Se

pueden usar para filtrar, ordenar datos basados en el tiempo, realizar análisis temporales, y gestionar eventos en secuencia cronológica. Esto es crucial en aplicaciones que necesitan realizar seguimientos temporales precisos, como logs de eventos, registros de transacciones, o datos históricos.

Keyword.- Facilita búsquedas exactas y operaciones de agregación eficientes. Se buscan términos completos facilitando su exactitud en filtros y búsquedas.

Text.- Para los campos que contienen texto libre como descripciones, títulos, autores, etc., el tipo `text` es más adecuado, ya que permite un análisis completo del texto para mejorar la relevancia de las búsquedas de texto completo. Esto te permitirá realizar búsquedas más flexibles y potentes en esos campos. El tipo `text` permite a Elasticsearch hacer una “Análisis de textos”. Como se ha comentado en clase el análisis de texto en Elasticsearch se refiere al proceso de convertir texto en términos o tokens que puedan ser indexados. Incluye pasos como la tokenización (dividir el texto en palabras o términos), la eliminación de palabras vacías (como "y", "o", "el"), y la aplicación de filtros de stemming (reducir las palabras a su raíz, por ejemplo, "corriendo" a "correr"). Este proceso permite realizar búsquedas más efectivas en campos de texto completo, mejorando la relevancia y precisión de los resultados de búsqueda.

Otros tipos: También existen en Elasticsearch otros tipos de datos como *long*, *double*, *boolean*, *date*, *geo_point* para ubicaciones geográficas, y *nested* para documentos anidados. También ofrece tipos especializados como *ip* para direcciones IP y *completion* para sugerencias de autocompletado.

(2 puntos) Cuando hayas obtenido el Mapping óptimo y el Ingest Pipeline pégalos bajo esta líneas.

⑩ **Mapping:**

⑩ **Ingest Pipeline:**

4. Búsquedas.

Una vez ya conoces los campos del índice vas a realizar varias pruebas de búsqueda en la **Dev Tools** para buscar por:

1. Book-Title
2. Book-Author
3. Year-Of-Publication (¿Se puede hacer que busque desde o hasta un año determinado? Si lo haces, impleméntalo).
4. Publisher
5. Se recomienda que se haga una 5ª consulta para que se busque un término en todos los campos anteriores.

Recuerda que debes buscar usando la API `_search` y en tu JSON debes indicar que haces una consulta “*query*” y luego puedes afinar tu búsqueda usando *match*, *match_phrase*, *must*, *should*, *must_not*, *operator*, *minimum_should_match*, etc.

5. Search template

Cuando ya hayas afinado las 5 búsquedas anteriores, crearás un Search Template por cada búsqueda.

(2,5 puntos) Pega aquí debajo esas 5 búsquedas almacenadas:

1. Book-Title
2. Book-Author
3. Year-Of-Publication
4. Publisher
5. Se recomienda que se haga una 5ª consulta para que se busque un término en todos los campos anteriores.

(2 puntos) 6. Web de consuta

Haz una web donde puedas buscar por diferentes términos. En la web habrá un control desplegable (DropDown) donde se elegirá buscar por una de las 5 opciones anteriores, la 1ª opción por defecto, será la 5ª búsqueda, buscar en todo, el resto de opciones se tendrán que seleccionar desplegando el control.

Habrà un botón donde se apretará para enviar la consulta a elasticsearch y con el JSON de respuesta de elasticsearch pintaremos los resultados obtenidos en nuestra página.

(2 puntos) 6. Web de consuta y modificación

- ⑩ Haz una web donde habrá un formulario con todos los campos. En esta web se podrá buscar solo por ISBN.
- ⑩ El primer campo será el ISBN y tendrá un botón que será de búsqueda.
- ⑩ Al recuperar el libro rellenarás todos los controles con los datos correspondientes.
- ⑩ Podrás hacer la modificación que desees en cualquier campo.
- ⑩ Existirá un botón “Actualizar Datos” que hará que se envíen los datos al servidor y se modifiquen los datos del libro editado.