

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/saurabhbagchi/books-dataset/>

1. Importación.

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

Kibana Home Page → Upload a file

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:

Menu → Analytics → Discover → Data → Index Management → (índice) → Mappings

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?

Las cabeceras no corresponden con las del CSV. Esto es debido a que en algunos títulos, o otros textos, tenemos comillas que entran en conflicto con las comillas que delimitan nuestros strings.

2. ¿Te resulta cómodo trabajar con estos nombres de campos? Justifica tu respuesta.

No. Por que tendré que estar constantemente consultando qué dato corresponde a que nombre de columna.

3. ¿Qué campo se ha cogido como Timestamp? ¿Es correcto el campo que tiene asignado el tipo *timestamp*?

@timestamp:Nov 20, 2286 @ 18:46:39.000

Lo ha cogido de manera incorrecta. Está fecha no se encuentra en mi CSV. Debería ser Year-Of-Publication

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.

```
"%{NUMBER:iban}\";%{QUOTEDSTRING:book_title};%{QUOTEDSTRING:author};"%{NUMBER:year_publish}\";%{QUOTEDSTRING:publisher};\"%{URI:img_s}\";\"%{URI:img_m}\";\"%{URI:img_l}\";\" OJO!!! HE TENIDO QUE INDICAR TAMBIÉN LAS COMILLAS EN FECHA
```

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.

Mapping:

```
{
  "properties": {
    "@timestamp": {
      "type": "date"
    },
    "author": {
      "type": "text"
    },
    "book_title": {
      "type": "text"
    },
    "iban": {
      "type": "text"
    },
    "img_l": {
      "type": "keyword"
    },
    "img_m": {
      "type": "keyword"
    },
    "img_s": {
      "type": "keyword"
    },
    "message": {
      "type": "text"
    },
    "publisher": {
      "type": "text"
    }
  }
}
```

Ingest Pipeline:

```
{
  "description": "Ingest pipeline created by text structure finder",
  "processors": [
    {
      "grok": {
        "field": "message",
        "patterns": [
          "\\"%{NUMBER:iban}\\\\";\"%{QUOTEDSTRING:book_title};\"%{QUOTEDSTRING:author};\\"%{NUMBER:year_publish}\\\\";\"%{QUOTEDSTRING:publi
sher};\\"\"%{URI:img_s}\\\\";\\\\"\"%{URI:img_m}\\\\";\\\\"\"%{URI:img_l}\\\\"\"
        ]
      }
    },
    {
      "date": {
        "field": "year_publish",
        "formats": [
          "UNIX"
        ]
      }
    },
    {
      "convert": {
        "field": "iban",
        "type": "long",
        "ignore_missing": true
      }
    },
    {
      "remove": {
        "field": "message"
      }
    }
  ]
}
```

4. Búsquedas.

Mis columnas: `ban` - `book_title` - `author` - `year_publish` - `publisher` - `img_s` - `img_m` - `mg_l`

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, implementarlo).
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

```
177
178 PUT _scripts/section_libros
179 {
180   "script": {
181     "lang": "mustache",
182     "source": {
183       "query": {
184         "match": {
185           "book_title": "{{title}}"
186         }
187       }
188     }
189   }
190 }
191 }
192 }
193 }
```

```
23 # Ejercicio 4
24 # Búsqueda por título
25 GET library/_search
26 {
27   "query": {
28     "match": {
29       "book_title": "Explore America"
30     }
31   }
32 }
33
34
35 GET library/_search
36 {
37   "query": {
38     "match_phrase": {
39       "book_title": "Women Who Made a Difference"
40     }
41   }
42 }
43 }
```

Click to send request

```
6   "total" : 1,
7   "successful" : 1,
8   "skipped" : 0,
9   "failed" : 0
10 },
11 "hits" : {
12   "total" : {
13     "value" : 2,
14     "relation" : "eq"
15   },
16   "max_score" : 28.308393,
17   "hits" : [
18     {
19       "_index" : "library",
20       "_type" : "doc",
21       "_id" : "EyHTJY4BLEeRLcTNKNH9",
22       "_score" : 28.308393,
23       "_source" : {
24         "author" : "\"Malcolm Forbes\"",
25         "img_1" : "http://images.amazon.",
26         "img_m" : "http://images.amazon."
```

2. Book-Author

```
225 PUT _scripts/section_libros
226 {
227   "script": {
228     "lang": "mustache",
229     "source": {
230       "query": {
231         "wildcard": {
232           "author": "*{{author}}*"
233         }
234       }
235     }
236   }
237 }
```

```
41 GET library2/_search
42 {
43   "query": {
44     "match": {
45       "author": "Edvard"
46     }
47   }
48 }
49
50
51 GET library2/_search
52 {
53   "query": {
54     "wildcard": {
55       "author": "*dvar*"
56     }
57   }
58 }
59
13   "value" : 1,
14   "relation" : "eq"
15 },
16   "max_score" : 12.019916,
17   "hits" : [
18     {
19       "_index" : "library2",
20       "_type" : "_doc",
21       "_id" : "2",
22       "_score" : 12.019916,
23       "_source" : {
24         "book_title" : "Nuevo Libro",
25         "author" : "Edvard",
26         "year_publish" : 2023,
27         "publisher" : "Editorial Nueva",
28         "color" : "Azul",
29         "nombre" : "Nombre Nuevo"
30       }
31     }
32 ]
```

```
13
14 # Búsqueda por autor ### NO ENTIENDO PORQUÉ CON MATCH_PHRASE NO ENCUENTRA
15 GET library/_search
16 {
17   "query": {
18     "match_phrase": {
19       "author": "Catherine M. Edmonson"
20     }
21   }
22 }
23
24 # Búsqueda que contenga tanto del autor
25 GET library/_search
26 {
27   "query": {
28     "wildcard": {
29       "author": "*Marshall*"
30     }
31   }
32 }
33
34
7   "successful" : 1,
8   "skipped" : 0,
9   "failed" : 0
10 },
11   "hits" : {
12     "total" : {
13       "value" : 261,
14       "relation" : "eq"
15     },
16     "max_score" : 1.0,
17     "hits" : [
18       {
19         "_index" : "library",
20         "_type" : "doc",
21         "_id" : "MydTJY4BLeeRLcTNC21A",
22         "_score" : 1.0,
23         "_source" : {
24           "author" : "\\Leslie Marshall\\",
25           "img_l" : "http://images.amazon.com/images/P/08
26           "img_m" : "http://images.amazon.com/images/P/08
27           "message" : "\\0802141390\\";\\A Girl Could Star
```

3. Year-Of-Publication

```
193
194 PUT _scripts/section_libros
195 {
196   "script": {
197     "lang": "mustache",
198     "source": {
199       "query": {
200         "range": {
201           "year_publish": {
202             "gte": "{{start_year}}",
203             "lte": "{{end_year}}"
204           }
205         }
206       }
207     }
208   }
209 }
210
```

```
8 GET library2/_search
9 {
10   "query": {
11     "range": {
12       "year_publish": {
13         "gte": "1996",
14         "lte": "2000"
15       }
16     }
17   }
18 }
19
20 GET library2/_search
21 {
22   "query": {
23     "range": {
24       "year_publish": {
25         "gte": "2020",
26         "lte": "2024"
27       }
28     }
29   }
30 }
31
```

```
6   "total" : 1,
7   "successful" : 1,
8   "skipped" : 0,
9   "failed" : 0
10 },
11 "hits" : {
12   "total" : {
13     "value" : 2,
14     "relation" : "eq"
15   },
16   "max_score" : 1.0,
17   "hits" : [
18     {
19       "_index" : "library2",
20       "_type" : "_doc",
21       "_id" : "SnIpLo4BBNu89zxQ8_k7",
22       "_score" : 1.0,
23       "_source" : {
24         "author" : "\"D. GIBBONS\"",
25         "year_publish" : "2024",
26         "img_l" : "http://images.amazon.co
27         "img_m" : "http://images.amazon.co
28         "message" : "\"0380000059\";\"CLOU
                ://images.amazon.com/images/P/03
```


4. Publisher

```
211 PUT _scripts/section_libros|
212 {
213   "script": {
214     "lang": "mustache",
215     "source": {
216       "query": {
217         "match": {
218           "publisher": "{{publisher}}"
219         }
220       }
221     }
222   }
223 }
```

```
90 # Por editorial "publisher"
91 GET library/_search
92 {
93   "query": {
94     "match": {
95       "publisher": "Creative Pub Intl"
96     }
97   }
98 }
99
100 GET library/_search
101 {
102   "query": {
103     "match_phrase": {
104       "publisher": "Pub Group West"
105     }
106   }
107 }
108
```

Click to send request

```
4   "timed_out" : false,
5   "_shards" : {
6     "total" : 1,
7     "successful" : 1,
8     "skipped" : 0,
9     "failed" : 0
10  },
11  "hits" : {
12    "total" : {
13      "value" : 737,
14      "relation" : "eq"
15    },
16    "max_score" : 11.023001,
17    "hits" : [
18      {
19        "_index" : "library",
20        "_type" : "doc",
21        "_id" : "KcdTJY4BLEeRLcTNCmV8",
22        "_score" : 11.023001,
23        "_source" : {
```

5. Se recomienda que se haga una 5ª consulta para que se busque un término en todos los campos anteriores.

```
267 PUT _scripts/section_libros
268 {
269   "script": {
270     "lang": "mustache",
271     "source": {
272       "query": {
273         "multi_match": {
274           "query": "{{query}}",
275           "fields": ["book_title", "author", "year_publish", "publisher"]
276         }
277       }
278     }
279   }
280 }
```

```
109 # Búsqueda de una query en todos los campos anteriores
110 GET library/_search
111 {
112   "query": {
113     "multi_match": {
114       "query": "Curiosities",
115       "fields": ["book_title", "author", "year_publish", "publisher"]
116     }
117   }
118 }
119
120 GET library/_search
121 {
122   "query": {
123     "multi_match": {
124       "query": "1997",
125       "fields": ["book_title", "author", "year_publish", "publisher"]
126     }
127   }
128 }
```

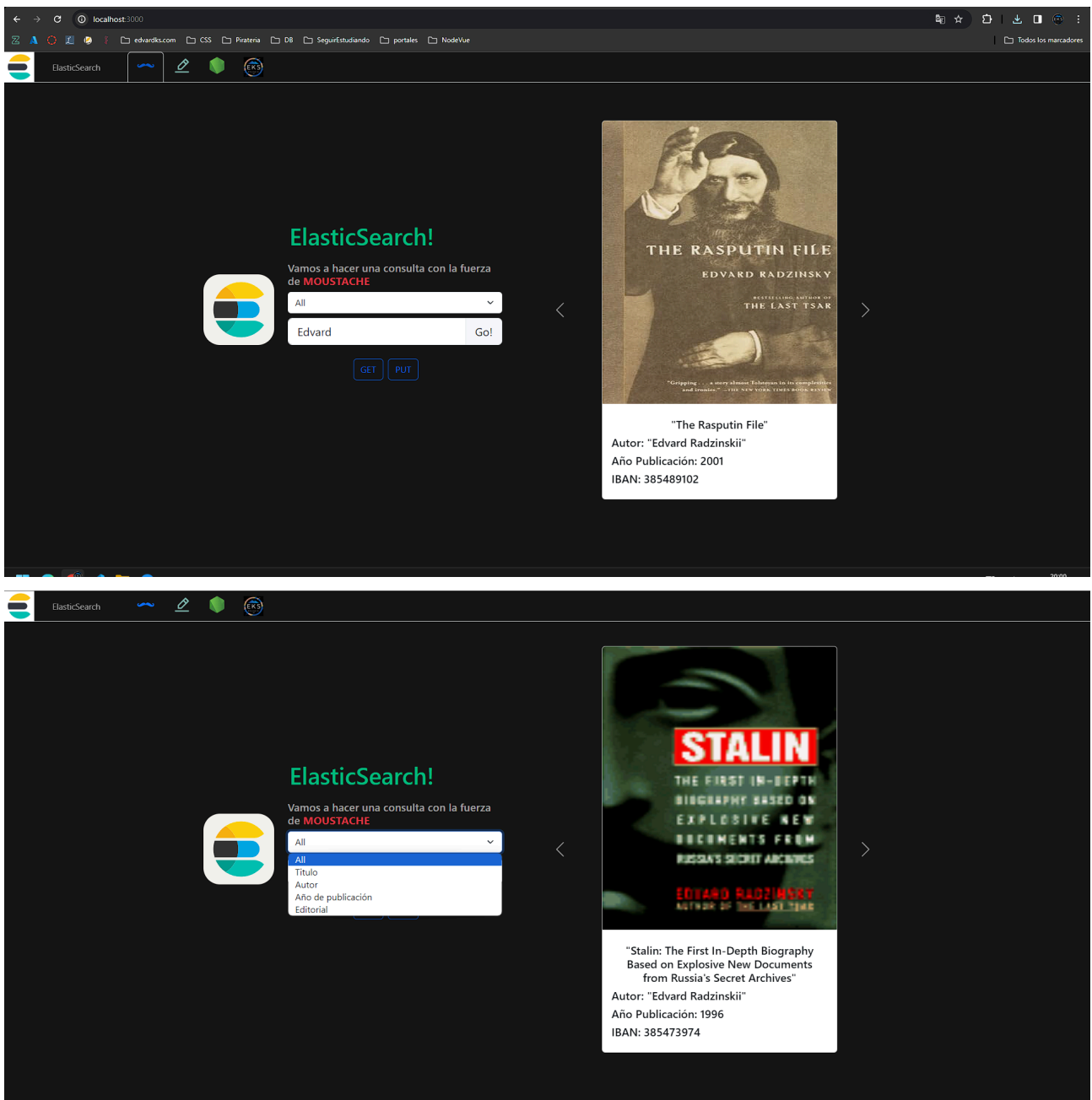
```
4   "timed_out" : false,
5   "_shards" : {
6     "total" : 1,
7     "successful" : 1,
8     "skipped" : 0,
9     "failed" : 0
10  },
11  "hits" : {
12    "total" : {
13      "value" : 78,
14      "relation" : "eq"
15    },
16    "max_score" : 11.1640835,
17    "hits" : [
18      {
19        "_index" : "library",
20        "_type" : "_doc",
21        "_id" : "KiHTJY4BLEeRLcTNKuCd",
22        "_score" : 11.1640835,
23        "_source" : {
24          "author" : "\"Christina Martin\""
```

(2 puntos)

6. Web de consulta

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 apretarán 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 consulta y modificación

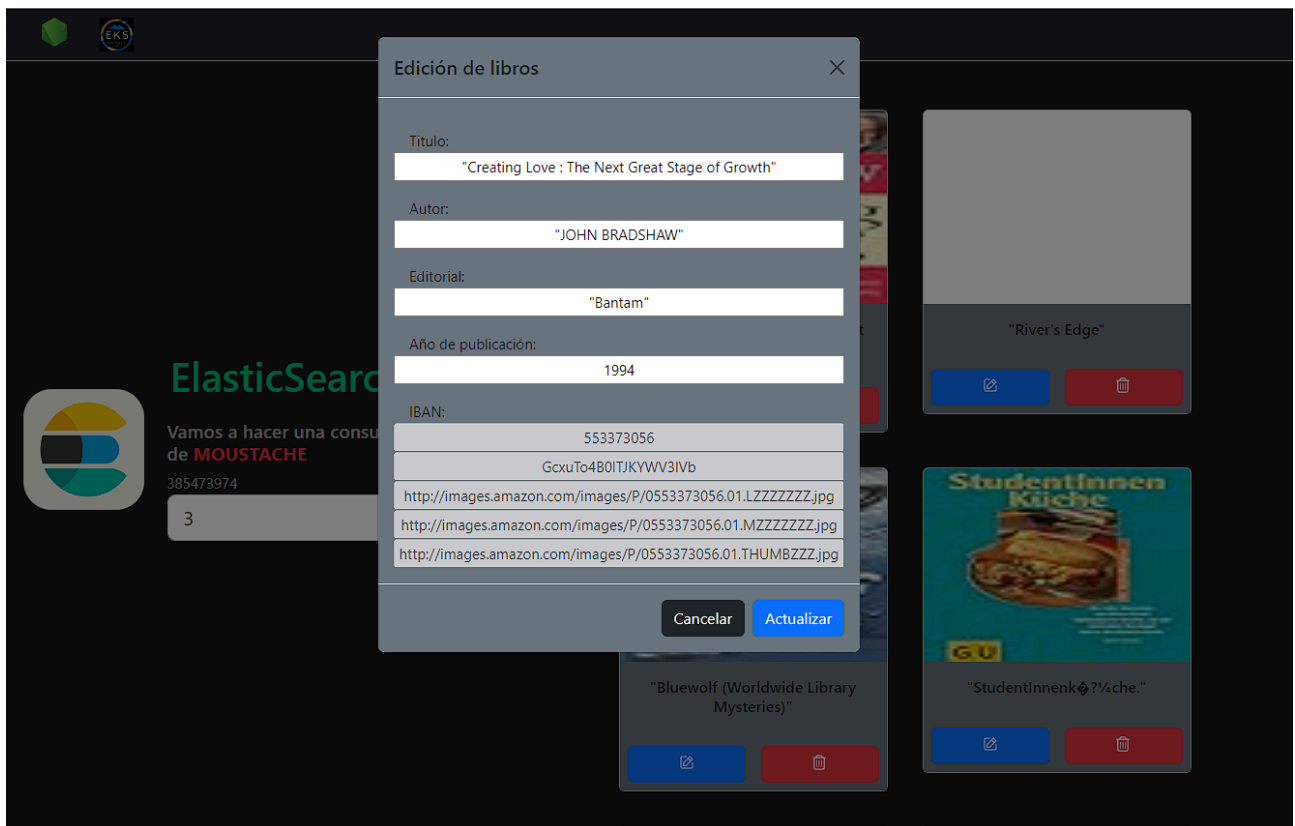
Haz una web donde habrá un formulario con todos los campos. En esta web se podrá buscar solo por ISBN.

El primer campo será el ISBN y tendrá un botón que será de búsqueda.

Al recuperar el libro rellenamos todos los controles con los datos correspondientes.


Podrás hacer la modificación que desees en cualquier campo.

Existirá un botón “Actualizar Datos” que hará que se envíen los datos al servidor y se modifiquen los datos del libro editado.



```
3 GET goodbot/_search/template
4
5 {
6   "id": "b_iban",
7   "params": {
8     "iban": "553373056"
9   }
10 }
11
12
```

```
20 "type": "_doc",
21 "_id": "GcxuTo4B0ITJKYWV3IVb",
22 "_score": 1.0,
23 "_source": {
24   "book_title": "\"Creating Love : The Next Great Stage of Growth\"",
25   "author": "\"JOHN BRADSHAW\"",
26   "publisher": "Edvard",
27   "year_publish": "1994",
28   "iban": "553373056",
29   "img_1": "http://images.amazon.com/images/P/0553373056.01.LZZZZZZZ.jpg"
30 }
```



ElasticSearch

Vamos a hacer una consulta de **MOUSTACHE**

385473974

3

Edición de libros

Título:
"Creating Love : The Next Great Stage of Growth"

Autor:
"JOHN BRADSHAW"

Editorial:
Edvard

Año de publicación:
1994

IBAN:
553373056
GcxuTo4B0ITJKYWV3IVb
<http://images.amazon.com/images/P/0553373056.01.LZZZZZZZ.jpg>
<http://images.amazon.com/images/P/0553373056.01.MZZZZZZZ.jpg>
<http://images.amazon.com/images/P/0553373056.01.THUMBZZZ.jpg>

Cancelar Actualizar