

Elasticsearch: Plataforma Virtual

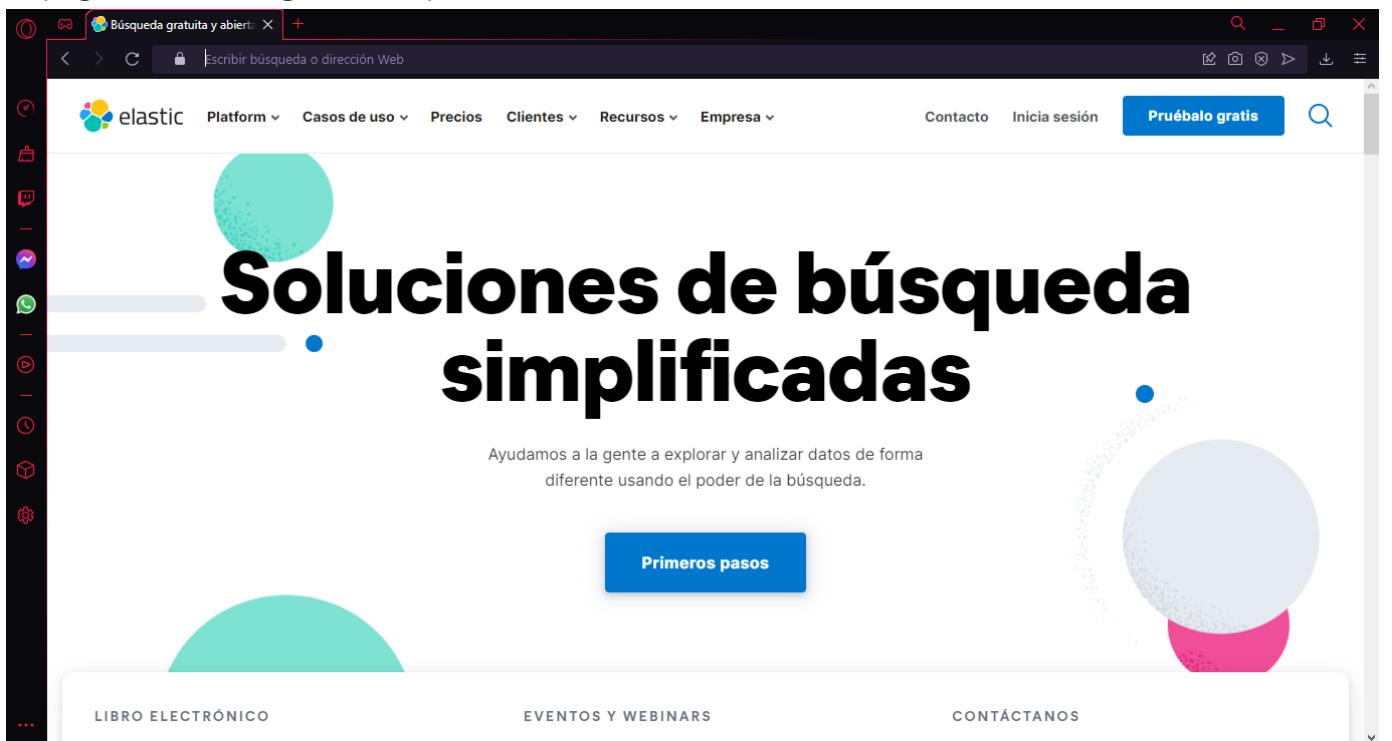
Ing.Luis Felipe Narvaez Gomez. E-mail: luis.narvaez@usantoto.edu.co. Cod: 2312660. Facultad de Ingeniería de Sistemas.

INGRESAR A ELASTIC

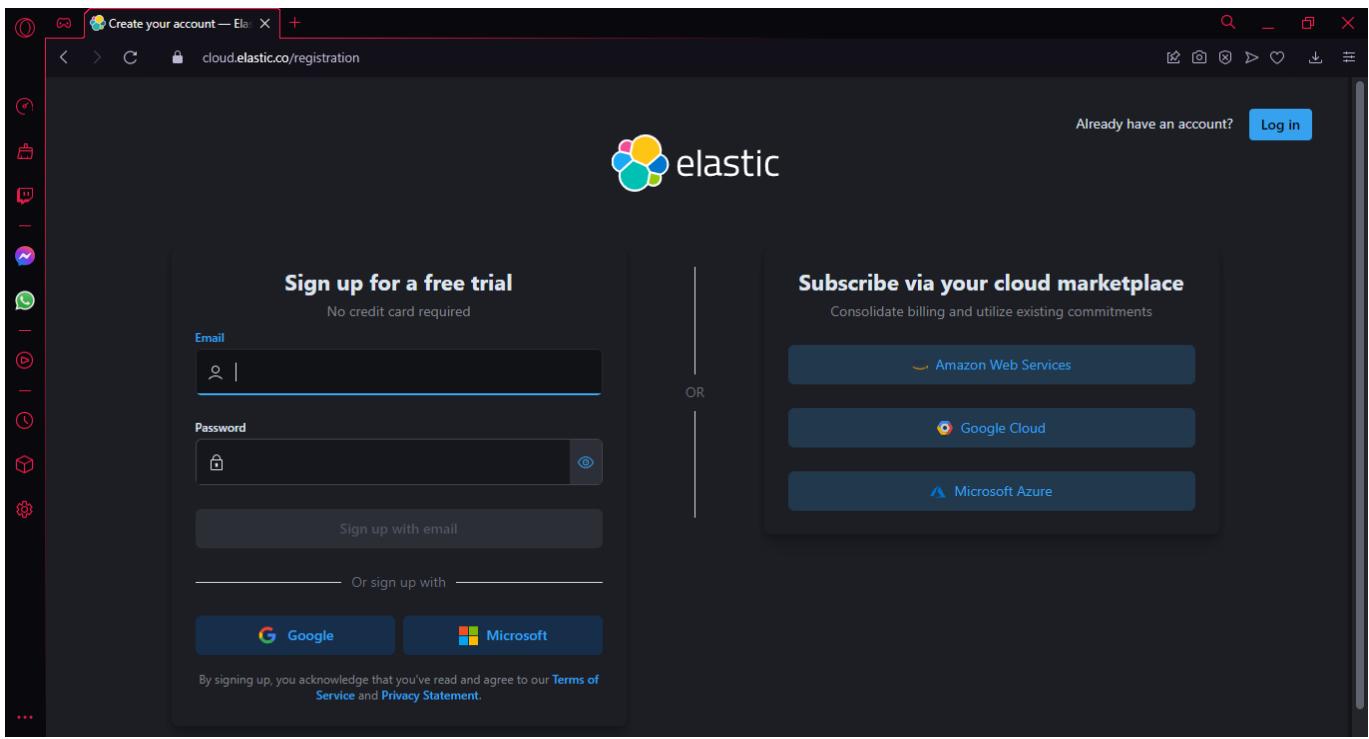
Primero hemos de ingresar al link de la pagina oficial de Elasticsearch, mediante el siguiente enlace:

<https://www.elastic.co/es/>

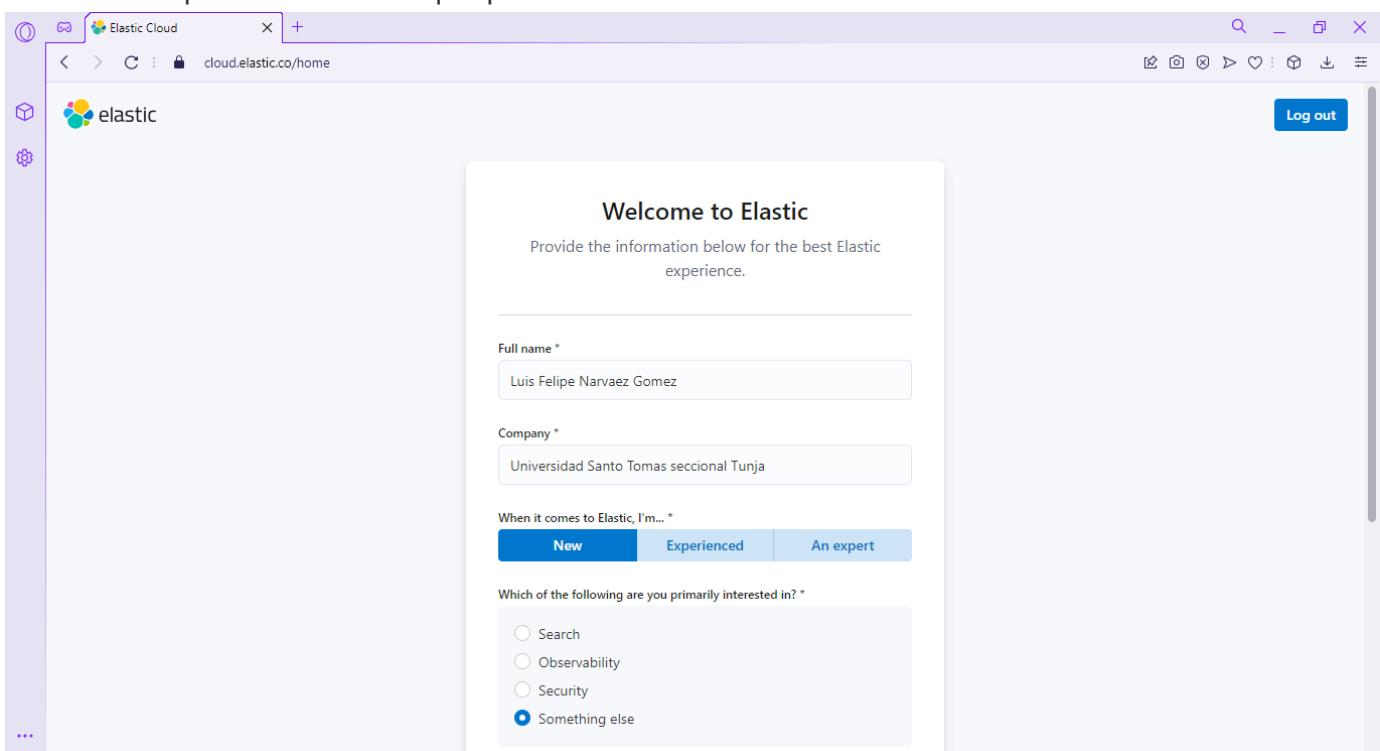
La pagina tiene el siguiente aspecto:



En esta pagina nos dirigiremos al botón de la esquina superior derecha "Pruébalo Gratis" dando clic aquí. Ahí nos encontraremos en una pagina donde se nos pedirá registrarnos para acceder al servicio. Podemos utilizar cuentas de google, microsoft o cualquier otro servicio de correo electrónico como yahoo.



Si somos nuevos en el sistema de elastic se nos pedirá anexar el porque de nuestro registro, así como la compañía o sector al que pertenecemos.



CREAR UN TRABAJO DE DESARROLLO

Automáticamente se nos dirigirá a que creemos nuestro primer trabajo desarrollo.

Create Deployment — Elastic X +

cloud.elastic.co/deployments/create

Create your first deployment

A deployment includes Elasticsearch, Kibana, and other Elastic Stack features, allowing you to store, search, and analyze your data.

Name

Settings Hide

Cloud provider: Google Cloud

Region: us-Iowa (us-central1)

Hardware profile: Storage optimized

Version: 8.4.2 (latest)

Create deployment

Luego se generaran nuestras credenciales para este Proyecto, las mismas podemos descargarlas en nuestro sistema como un Archivo CSV o bien copiarlas en algún sitio como un archivo de texto plano. Deberemos esperar 5 minutos en lo que se crea el proyecto y nos da tiempo de hacernos con las importantes credenciales de tipo par.

[f10503] Overview — Elastic X +

cloud.elastic.co/deployments/f10503963ca94cf2a829272ff88a71a5/getting-started

elastic Trial - 14 days left

Creating your deployment (takes about five minutes) Continue

Save the deployment credentials

These root credentials are shown only once.
They provide super user access to your deployment. Keep them safe.

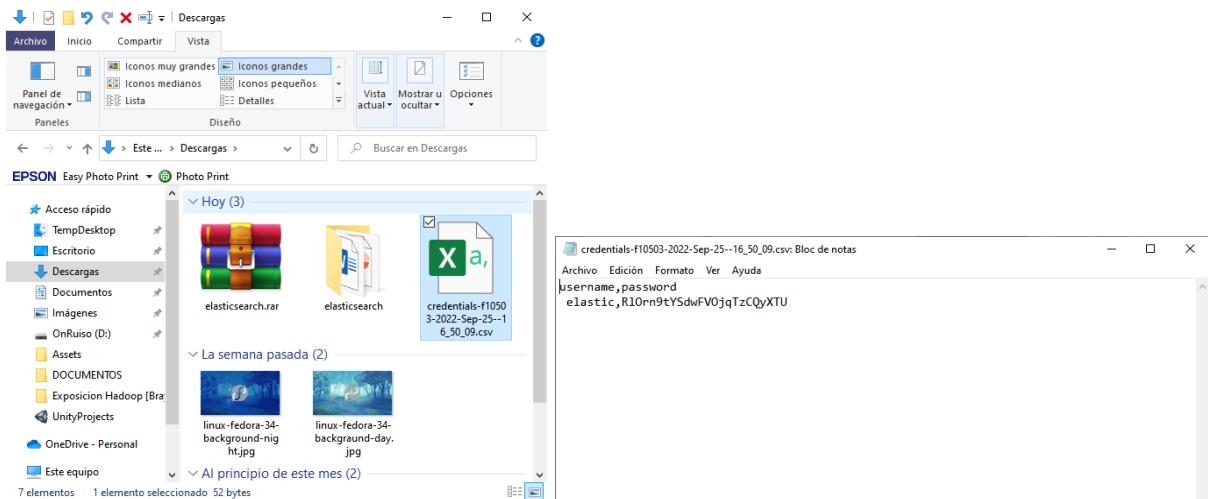
Username: elastic
Password: R10rn9tYSdwFVOjqTzCQyXTU

Copy to clipboard

Download

Skip

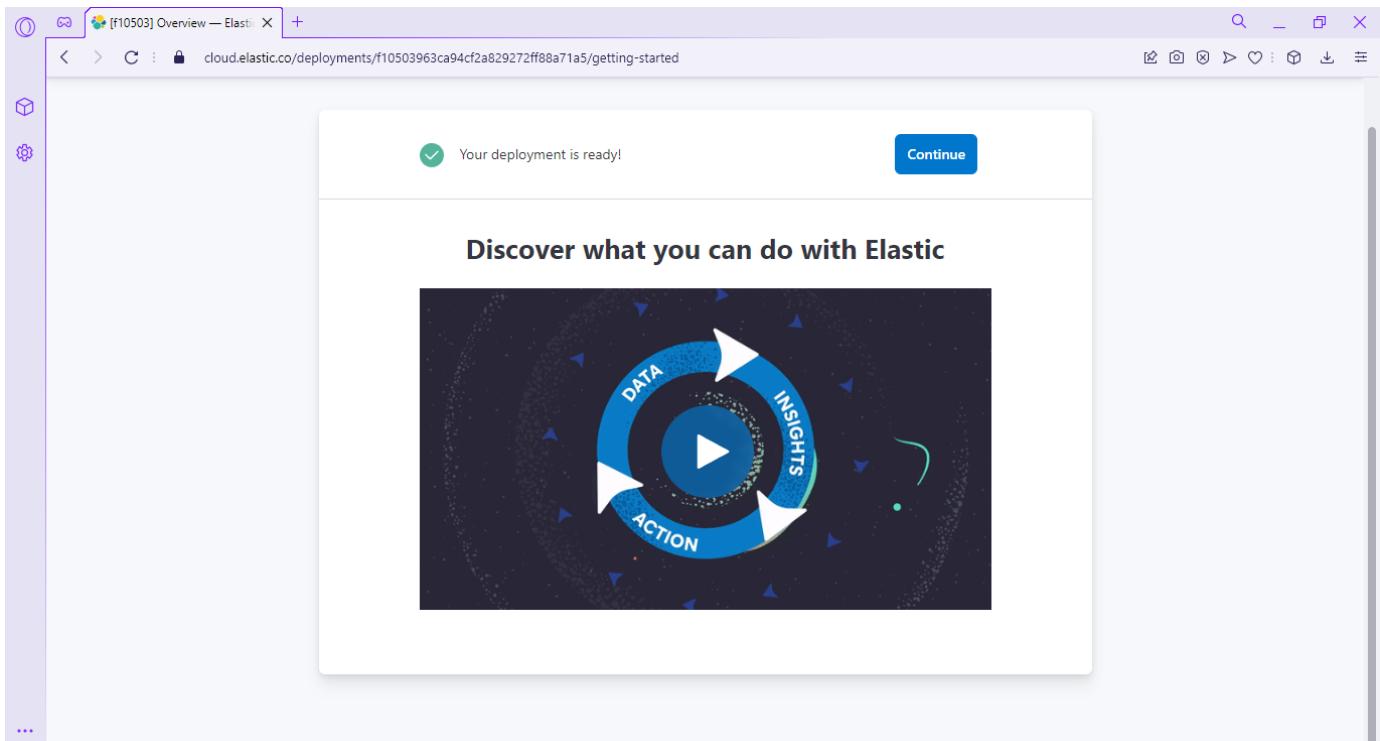
Con el archivo descargado tendríamos lo siguiente:



Mientras que como archivo de texto plano en nuestro portapapeles tendríamos algo similar a esto:

```
Username: elastic
Password: R10rn9tYSdwFV0jqTzCQyXTU
```

Luego de que termine con la creación del proyecto tendríamos lo siguiente:



Solo debemos dar clic en el botón de "continue" y tendremos una dashboard de trabajo similar a la siguiente:

The screenshot shows the Elastic Home dashboard. At the top, it asks "What would you like to do first?". Below are three cards:

- Search my data**: Create a finely-tuned search experience for your websites, applications, workplace content, and more.
- Monitor my environments**: Get end-to-end observability into your environments by consolidating your logs, metrics, and traces.
- Protect my environment**: Protect your environment against threats by unifying SIEM, endpoint security, and cloud security in one place.

AGREGAR UN DATAFRAME

Ahora para poder agregar un data frame, navegaremos por la dashboard a la sección del Menú de Hamburguesa en la parte superior izquierda seleccionándolo, allí encontraremos la función de Analíticas y dentro de ella la opción de Machine Learning.

The screenshot shows the Elastic Analytics dashboard. On the left, there is a sidebar with the following menu items:

- Manage this deployment
- Home
- Analytics
 - Discover
 - Dashboard
 - Canvas
 - Maps
 - Machine Learning
 - Graph
 - Visualize Library
- Enterprise Search
 - Overview
 - Content
- Add integrations

The "Machine Learning" option is highlighted. The main content area displays a "Welcome to Analytics!" message and a "Welcome to Machine Learning!" message with a link to learn more about ML.

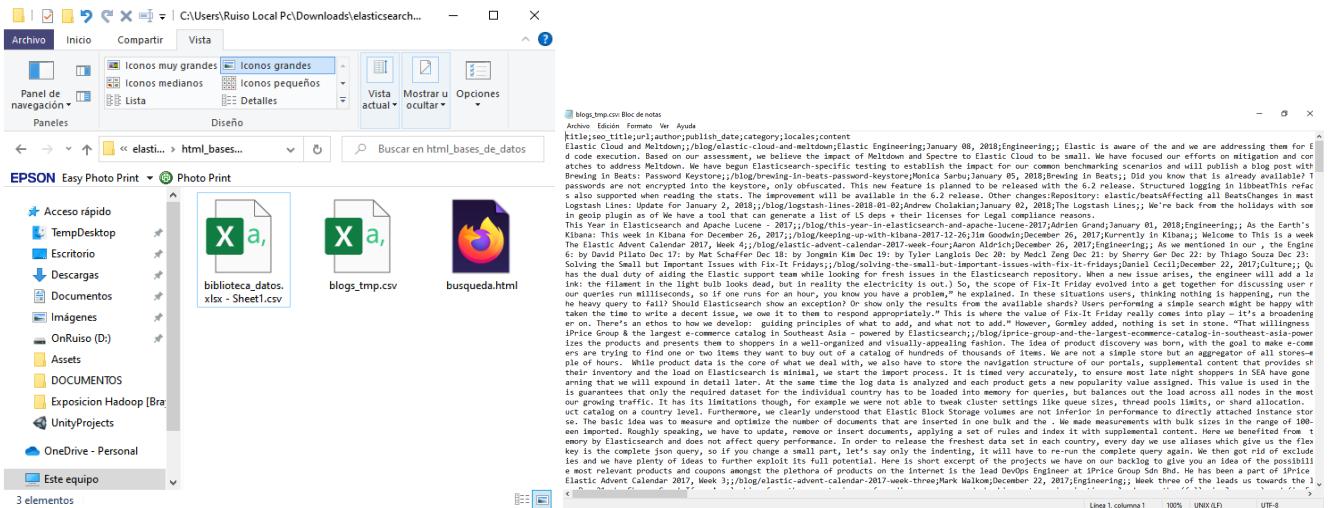
Al seleccionarla encontraremos lo siguiente:

The screenshot shows the 'Overview' page of the Elastic Machine Learning interface. On the left sidebar, under 'Machine Learning', there are sections for 'Overview', 'Anomaly Detection', 'Data Frame Analytics', and 'Model Management'. A 'File' option is also listed at the bottom of the sidebar. The main content area has a heading 'Overview' and a date range selector 'Last 15 minutes'. A yellow warning box states: '⚠ No ML nodes available. There are no ML nodes available. You will not be able to create or run jobs. Please edit your [Elastic Cloud deployment](#). You may enable a free 2GB machine learning node or expand your existing ML configuration.' Below this, a blue box contains 'Getting started' information: 'Welcome to Machine Learning. Get started by reviewing our [documentation](#) or creating a new job. If you have input or suggestions regarding your experience, please submit [feedback online](#)'. A 'Dismiss' button is located in this box. At the bottom, a table shows 'Total machine learning nodes: 0' with columns for 'Name', 'Total memory', and 'Memory usage'.

En la anterior pantalla buscaremos en el menú de la izquierda la opción de "Data Visualizer" y en ella seleccionamos "File".

The screenshot shows the 'File Data Visualizer' page. The left sidebar includes 'Jobs', 'Anomaly Explorer', 'Single Metric Viewer', 'Settings', 'Data Frame Analytics', 'Model Management', 'Data Visualizer' (which is selected and highlighted in blue), and 'AIOps'. Under 'Data Visualizer', there are 'File View' and 'Explain log rate spikes'. The main content area has a heading 'Visualize data from a log file' and instructions: 'Upload your file, analyze its data, and optionally import the data into an Elasticsearch index. The following file formats are supported: Delimited text files, such as CSV and TSV; Newline-delimited JSON; Log files with a common format for the timestamp. You can upload files up to 100 MB.' It features a large 'Select or drag and drop a file' button with a downward arrow icon.

Como podemos ver en la anterior pantalla podremos arrastrar o buscar un archivo que cumpla con los requerimientos que aquí se especifican. Subiremos un archivo CSV llamado "blogs_tmo.csv".



Al subir nuestro dataframe, podemos corroborar su estado en la pantalla:

Data Visualizer

blogs_tmp.csv

File contents
First 101 lines

```

1 title;seo_title;url;author;publish_date;category;locales;content
2 Elastic Cloud and Meltdown;;;/blog/elastic-cloud-and-meltdown;Elastic Engineering;January 08, 2018;Engineering;; Elastic is aware of the
and we are addressing them for Elastic Cloud. We know that you entrust your data to our cloud service, and we take the
confidence of all data very seriously. At this time, we are not aware of any exploit on our cloud service that utilized the
Meltdown or Spectre vulnerabilities. Impact Assessment The Meltdown and Spectre to Elastic Cloud to be small. We have focused our efforts on mitigation and control while we carry out our regular process for operating system patches in an accelerated fashion.
Mitigation We disabled non-sandboxed scripting for all Elasticsearch 1.x clusters as a primary, customer-visible mitigation. We have
also disabled self-service uploads of custom bundles from you until we have fully completed our patching. Behind the scenes, we've
further increased our observability of system-level calls and isolated clusters running version 1.x of Elasticsearch on their own

```

Summary

Number of lines analyzed: 101

Format: delimited

Delimiter: ;

Has header row: true

File stats

All fields	8 of 8 total	Number fields	0 of 0 total	Field name	8	Field type	2	...
> Type	Name ↑	Documents (%)	Distinct values	Distributions	...			
> k	author	100 (100%)	51	top 10 of 51 categories	...			
> t	category	99 (99%)	9			
> t	content	99 (99%)	99			
> k	locales	7 (7%)	6	6 categories	...			
> k	publish_date	100 (100%)	45	top 10 of 45 categories	...			
> t	seo_title	19 (19%)	19			
> t	title	100 (100%)	100			
> k	url	100 (100%)	100	top 10 of 100 categories	...			

Rows per page: 25 < 1 >

Import **Cancel**

Continuamente podemos dar clic en Import habiendo verificado que todo este bien. Luego asignamos un nombre para el índice y damos nuevamente en Import.

The screenshot shows the Data Visualizer interface with the following details:

- Left sidebar:** Includes sections for Jobs, Anomaly Explorer, Single Metric Viewer, Settings, Data Frame Analytics (Jobs, Results Explorer, Analytics Map), Model Management (Trained Models, Nodes), Data Visualizer (File View, Data View), and AI Ops.
- Main area:** Title "Data Visualizer" and file "blogs_tmp.csv".
- Import data section:** Subtitle "Simple Advanced", "Index name" input field containing "blogs", checked checkbox "Create data view", and a blue "Import" button.
- Bottom buttons:** "Back" and "Cancel".

Luego esperamos a que se completen todos los pasos clave en el proceso de la subida de nuestro dataframe.

The screenshot shows the Data Visualizer interface after the import process has completed, displaying the following information:

- Summary of completed steps:**
 - File processed
 - Index created
 - Ingest pipeline created
 - Data uploaded
 - Data view created
- Import complete details:**
 - Index: blogs
 - Data view: blogs
 - Ingest pipeline: blogs-pipeline
 - Documents ingested: 100
- Action buttons:**
 - View index in Discover
 - Index Management
 - Data View Management
 - Create Filebeat configuration
 - Open in Data Visualizer
- Bottom buttons:** "Back" and "Cancel".

CONSULTAS SOBRE EL DATAFRAME

Y listo, tenemos nuestro DataFrame en nuestro Elastic, ahora podemos empezar a realizar consultas en él. Para esto volvemos al menú lateral izquierdo en la sección de "Management" allí seleccionamos "DevTools".

The screenshot shows the Elasticsearch Dev Tools interface. On the left, there's a sidebar with 'Management' selected under 'Dev Tools'. Below it are links for Integrations, Fleet, Osquery, Stack Monitoring, and Stack Management. A blue button at the bottom says '+ Add Integrations'. The main area has a title 'Import complete' with a checkmark. It lists 'Index' (blogs), 'Data view' (blogs), 'Ingest pipeline' (blogs-pipeline), and 'Documents ingested' (100). Below this is a row of five buttons: 'View index in Discover' (with a magnifying glass icon), 'Index Management' (with a gear icon), 'Data View Management' (with a gear icon), 'Create Filebeat configuration' (with a file icon), and 'Open in Data Visualizer' (with a magnifying glass icon).

Al dar clic se nos abrirá la siguiente ventana tipo consola:

The screenshot shows the Elasticsearch Dev Tools Console interface. At the top, there are tabs for 'Console', 'Search Profiler', 'Grok Debugger', and 'Painless Lab (BETA)'. The 'Console' tab is selected. In the editor pane on the left, there is some sample code:

```
1 # Click the Variables button, above, to create your own variables.
2 GET ${exampleVariable1} // _search
3 +
4 {
5   "query": {
6     | "${exampleVariable2)": {} // match_all
7   }
7 }
```

On the right, there's a response pane with a title 'Welcome to Console'. It includes a 'Quick intro to the UI' section with text about the two-pane layout and compact request format. It also shows a snippet of curl-like code:

```
1 # index a doc
2 PUT index/_doc/1
3 +
4 {
5   "body": "here"
6 +
7 # and get it ...
8 GET index/_doc/1
```

A 'Dismiss' button is at the bottom right of the response pane.

Primero verificamos que los datos estén correctamente almacenados en Elastic, esto por medio de la siguiente función:

```
GET _cat/indices
```

Para ejecutarla basta con dar clic en el símbolo de "Run" que se encuentra en la misma linea.

```

1 # Click the Variables button, above, to create your own variables.
2 GET ${exampleVariable1} // _search
3 {
4   "query": {
5     | "${exampleVariable2)": {} // match_all
6   }
7 }
8 GET _cat/indi

```

▶ 🔍

_cat/indices endpoint

La respuesta es:

200 - 168 ms

```

1 green open blogs
FQGVwMERQ004tEgfwQV4Pg 1 1 100 0 827.7kb 413.8kb
2 green open .ds-logs-enterprise_search.api-default-2022.09.25-000001
Xyqdj01JTQOEf4gxe-nd1Q 1 1 1 0 28.7kb 14.3kb
3 green open .ds-logs-enterprise_search.audit-default-2022.09.25-000001
K302U2WhSBYGkqJgRA28Fw 1 1 7 0 92.8kb 46.4kb
4

```

Ahora verificamos que los datos que subimos sean los correctos con la siguiente función:

GET blogs/_search

Como respuesta obtenemos lo siguiente:

Console Search Profiler Grok Debugger Painless Lab (BETA)

History Settings Variables Help

```

1 # Click the Variables button, above, to create your own variables.
2 GET ${exampleVariable1} // _search
3 {
4   "query": {
5     | "${exampleVariable2)": {} // match_all
6   }
7 }
8 GET _cat/indices
9 GET blogs/_search

```

▶ 🔍

```

1 {
2   "took": 1,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 100,
13      "relation": "eq"
14    }
15    "max_score": 1,
16    "hits": [
17

```

200 - 248 ms

Podemos consultar una palabra clave dentro de los blogs con la siguiente consulta:

```

GET blogs/_search
{
  "query": {
    "match": {
      "content": "elastic stack"
    }
  }
}

```

Dando como resultado que encontramos 67 registros con la directriz dada.

Console Search Profiler Grok Debugger Painless Lab (BETA)

History Settings Variables Help

```

1 # Click the Variables button, above, to create your own variables.
2 GET ${exampleVariable1} // _search
3 {
4   "query": {
5     | "${exampleVariable2)": {} // match_all
6   }
7 }
8 GET _cat/indices
9 GET blogs/_search
10 {
11   "query": {
12     "match": {
13       | "content": "elastic stack"
14     }
15   }
16 }

```

▶ 🔍

```

1 {
2   "took": 7,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 67,
13      "relation": "eq"
14    }
15    "max_score": 2.9073086,
16    "hits": [
17

```

200 - 297 ms

CONSTRUIR UNA PLANTILLA PARA LAS CONSULTAS

Las plantillas predefinidas de consultas nos benefician al momento de agilizar nuestras consultas, esto es debido a que en vez de que cada tipo de consulta escribamos todo el query respectivo, con la plantilla creada, solo lo utilizamos una vez.

Creemos nuestra consulta de la siguiente manera:

```
PUT _scripts/busquedas_blogs
{
  "script": {
    "lang": "mustache",
    "source": {
      "query": {
        "match": {
          "content": "{{terminos}}"
        }
      }
    }
  }
}
```

Lo cual se ve así con su confirmación de creación.

The screenshot shows the Elasticsearch Dev Tools interface with the 'Console' tab selected. On the left, a code editor displays the PUT request for creating a script. On the right, the results of the operation are shown, including the acknowledged status and the ID of the created script.

```
200 - 211 ms
1 { "acknowledged": true
2 }
```

Una vez creada la plantilla podemos utilizarla en diversas consultas, en este caso de términos entre todos los registros.

```
GET blogs/_search/template
{
  "id": "busquedas_blogs",
  "params": {
    "terminos": "AQUI PONER EL TERMINO A BUSCAR"
  }
}
```

Obteniendo resultados de la siguiente forma:

Console Search Profiler Grok Debugger Painless Lab BETA

History Settings Variables Help

```

1 # Click the Variables button, above, to create your own variables.
2 GET ${exampleVariable1} // _search
3 { }
8 GET _cat/indices
9 GET blogs/_search
10 { }
17 PUT _scripts/busquedas_blogs
18 { }
30 GET blogs/_search/template | 
31 { }
32 "id": "busquedas_blogs",
33 "params": {
34 | "terminos": "Logstash"
35 }
36 }
```

1 {
2 "took": 4,
3 "timed_out": false,
4 "_shards": {
5 "total": 1,
6 "successful": 1,
7 "skipped": 0,
8 "failed": 0
9 },
10 "hits": {
11 "total": {
12 "value": 27,
13 "relation": "eq"
14 }},
15 "max_score": 2.6010914,
16 "hits": [

200 - 364 ms

Console Search Profiler Grok Debugger Painless Lab BETA

History Settings Variables Help

```

1 # Click the Variables button, above, to create your own variables.
2 GET ${exampleVariable1} // _search
3 { }
8 GET _cat/indices
9 GET blogs/_search
10 { }
17 PUT _scripts/busquedas_blogs
18 { }
30 GET blogs/_search/template | 
31 { }
32 "id": "busquedas_blogs",
33 "params": {
34 | "terminos": "girls"
35 }
36 }
```

1 {
2 "took": 1,
3 "timed_out": false,
4 "_shards": {
5 "total": 1,
6 "successful": 1,
7 "skipped": 0,
8 "failed": 0
9 },
10 "hits": {
11 "total": {
12 "value": 1,
13 "relation": "eq"
14 }},
15 "max_score": 7.776444,

200 - 157 ms

Console Search Profiler Grok Debugger Painless Lab BETA

History Settings Variables Help

```

1 # Click the Variables button, above, to create your own variables.
2 GET ${exampleVariable1} // _search
3 { }
8 GET _cat/indices
9 GET blogs/_search
10 { }
17 PUT _scripts/busquedas_blogs
18 { }
30 GET blogs/_search/template | 
31 { }
32 "id": "busquedas_blogs",
33 "params": {
34 | "terminos": "science"
35 }
36 }
```

1 {
2 "took": 1,
3 "timed_out": false,
4 "_shards": {
5 "total": 1,
6 "successful": 1,
7 "skipped": 0,
8 "failed": 0
9 },
10 "hits": {
11 "total": {
12 "value": 1,
13 "relation": "eq"
14 }},
15 "max_score": 2.886918,

200 - 163 ms

Console Search Profiler Grok Debugger Painless Lab BETA

History Settings Variables Help

```

1 # Click the Variables button, above, to create your own variables.
2 GET ${exampleVariable1} // _search
3 { }
8 GET _cat/indices
9 GET blogs/_search
10 { }
17 PUT _scripts/busquedas_blogs
18 { }
30 GET blogs/_search/template | 
31 { }
32 "id": "busquedas_blogs",
33 "params": {
34 | "terminos": "technology"
35 }
36 }
```

1 {
2 "took": 1,
3 "timed_out": false,
4 "_shards": {
5 "total": 1,
6 "successful": 1,
7 "skipped": 0,
8 "failed": 0
9 },
10 "hits": {
11 "total": {
12 "value": 8,
13 "relation": "eq"
14 }},
15 "max_score": 3.8679426,
16 "hits": [

200 - 289 ms

Console Search Profiler Grok Debugger Painless Lab BETA

History Settings Variables Help

```

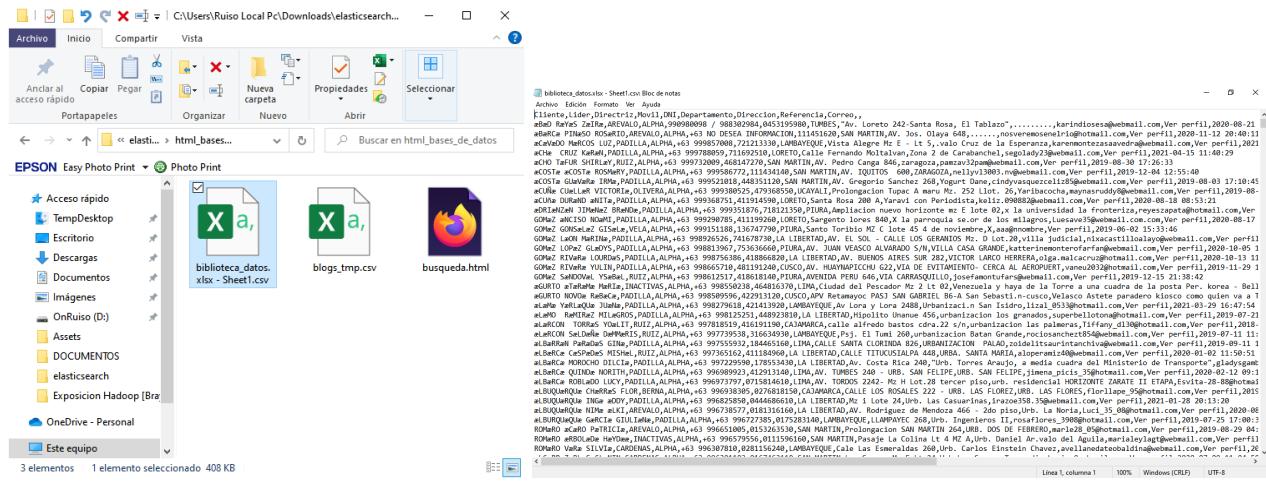
1 # Click the Variables button, above, to create your own variables.
2 GET ${exampleVariable1} // _search
3 { }
8 GET _cat/indices
9 GET blogs/_search
10 { }
17 PUT _scripts/busquedas_blogs
18 { }
30 GET blogs/_search/template | 
31 { }
32 "id": "busquedas_blogs",
33 "params": {
34 | "terminos": "business"
35 }
36 }
```

1 {
2 "took": 2,
3 "timed_out": false,
4 "_shards": {
5 "total": 1,
6 "successful": 1,
7 "skipped": 0,
8 "failed": 0
9 },
10 "hits": {
11 "total": {
12 "value": 7,
13 "relation": "eq"
14 }},
15 "max_score": 4.1983347,
16 "hits": [

200 - 264 ms

UN NUEVO TRABAJO

Tratemos de hacer lo mismo ahora con otro dataframe:



Subir el DataFrame en Data Visualizer.

Data Visualizer

Visualize data from a log file

Upload your file, analyze its data, and optionally import the data into an Elasticsearch index.



The following file formats are supported:

- Delimited text files, such as CSV and TSV
- Newline-delimited JSON
- Log files with a common format for the timestamp

You can upload files up to 100 MB.



Select or drag and drop a file

[Hide chat](#)

Importar el dataframe tras verificar el estado de la subida.

File Data Visualizer - Machi

bigdata-f10503.kb.us-central1.gcp.cloud.es._source

elastic

Machine Learning > Data Visualizer > File

Override settings Analysis explanation

File stats

All fields 11 of 11 total Number fields 1 of 1 total Field name 11

Type	Name ↑	Documents (%)	Distinct values	Distributions
> k	Cliente	999 (100%)	981	top 10 of 981 categories
> k	Correo	997 (99.8%)	996	top 10 of 996 categories
> #	DNI	999 (100%)	999	min 111116 median 416841 max 913726 21 750 480
> k	Departamento	999 (100%)	25	top 10 of 25 categories
> t	Direccion	999 (100%)	989	
> k	Directriz	999 (100%)	1	1 category
> k	Lider	999 (100%)	21	top 10 of 21 categories
> k	Movil	999 (100%)	992	top 10 of 992 categories
> t	Referencia	999 (100%)	964	
> k	column10	999 (100%)	1	1 category
> #	column11	999 (100%)	999	

Rows per page: 25 Hide chat < 1 >

Import Cancel

Indicar el indice de la data

Data Visualizer

biblioteca_datos.xlsx - Sheet1.csv

Import data

Simple Advanced

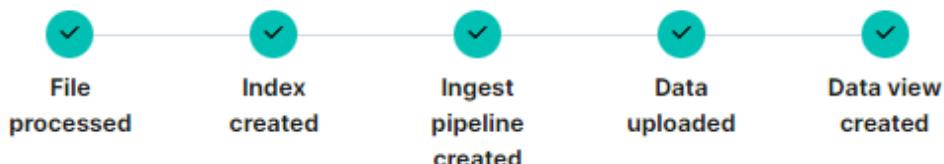
Index name

biblio

Create data view

Import

Confirmar la data subida.



✓ Import complete

Index biblio

Data view biblio

Ingest biblio-pipeline
pipeline

Documents 2196
ingested

Failed 1
documents

Acceder a la consola de consultas. Confirmación de los datos almacenados.

GET _cat/indices

Este es el resultado:

200 - 179 ms

```
1 green open blogs          FQGvMERQ004tEgfwQV4Pg 1 1 100 0 827.7kb 413.8kb
2 green open biblio         1nrKcniaSBUMioChr-toPg 1 1 2196 0 1.5mb 810kb
3 green open .ds-logs-enterprise_search.api-default-2022.09.25
   -000001 Xyqdj01JTQ0Ef4gxe-nd1Q 1 1 1 0 28.7kb 14.3kb
4 green open .ds-logs-enterprise_search.audit-default-2022.09.25
   -000001 K302U2WhsByGkqJgRA28Fw 1 1 7 0 92.8kb 46.4kb
5 |
```

Verificación de los datos sean los correctos:

```
GET biblio/_search
```

Tenemos alrededor de 2196 resultados en nuestro dataframe, que comparten una estructura similar a la siguiente

```
1+ {
2   "took": 1,
3   "timed_out": false,
4+   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9+   },
10+  "hits": {
11    "total": {
12      "value": 2196,
13      "relation": "eq"
14    },
15    "max_score": 1,
16    "hits": [
17      {
18        "_index": "biblio",
19        "_id": "D04-d4MB6FQ9BS4vssJy",
20        "_score": 1,
21        "_source": {
22          "column11": "2020-08-21 16:58:55",
23          "column10": "Ver perfil",
24          "Movil": "990980098 / 988302984",
25          "Directriz": "ALPHA",
26          "Departamento": "TUMBES",
27          "Referencia": ".....",
28          "Correo": "karindiosesa@webmail.com",
29          "@timestamp": "2020-08-21T16:58:55.000-05:00",
30          "Liden": "AREVALO",
31          "Direccion": "Av. Loreto 242-Santa Rosa, El
              Tablazo",
32          "Cliente": "ÆBæD RæYæS ZæIRæ",
33          "DNI": 453195980
34        }
35      },
36      {
37        "_index": "biblio",
```

Ahora hagamos una consulta mas especifica, por ejemplo sobre la columna de departamento.

```
GET biblio/_search
{
  "query": {
    "match": {
      "Departamento": "TUMBES"
    }
  }
}
```

Tenemos una respuesta de 22 consultas que tienen como departamento TUMBES.

```

1 # Click the Variables button
  , above, to create your
  own variables.
2 GET ${exampleVariable1} // _search
3 { }
8 GET _cat/indices
9 GET biblio/_search ➔ 🔍
10 {
11   "query": {
12     "match": {
13       | "Departamento": "TUMBES"
14     }
15   }
16 }
17

```

```

1 [
2   "took": 1,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 22,
13      "relation": "eq"
14    },
15    "max_score": 4.581332,
16    "hits": [
17

```

Ahora hagamos una consulta sobre cuantas personas hay en este dataframe de la ciudad de CUSCO.

```

GET biblio/_search
{
  "query": {
    "match": {
      "Departamento": "CUSCO"
    }
  }
}

```

Tenemos como respuesta a 36 personas con el valor de departamento = CUSCO.

```

1 # Click the Variables button
  , above, to create your
  own variables.
2 GET ${exampleVariable1} // _search
3 { }
8 GET _cat/indices
9 GET biblio/_search ➔ 🔍
10 {
11   "query": {
12     "match": {
13       | "Departamento": "CUSCO"
14     }
15   }
16 }
17

```

```

1 [
2   "took": 1,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 36,
13      "relation": "eq"
14    },
15    "max_score": 4.0975356,
16    "hits": [
17

```

Sabiendo de antemano que los nombres de los lideres son AREVALO, PADILLA, RUIZ, OLIVERA, VELA, INACTIVAS, BERNA, CARDENAS, GONZAGA JUAREZ, HUILCA, VERA, LOPEZ, REAT, GUI, ROJAS, FRETEL, JULCA, MESTANZA, REYES, JIMENEZ, DAVILA, CAYAMPE, VILLENA, PINEDA, ANGELES, SAAVEDRA. Podemos hacer una consulta burda por cada uno y contar los hits obtenidos.

```
GET biblio/_search
```

Sumándole uno por uno

```
{"query": {"match": {"Lider": "AREVALO"}}}
{"query": {"match": {"Lider": "PADILLA"}}}
{"query": {"match": {"Lider": "RUIZ"}}}
{"query": {"match": {"Lider": "OLIVERA"}}}
{"query": {"match": {"Lider": "VELA"}}}
{"query": {"match": {"Lider": "INACTIVAS"}}}
{"query": {"match": {"Lider": "BERNA"}}}
{"query": {"match": {"Lider": "CARDENAS"}}}
```

```
{
  "query": {"match": {"Lider": "GONZAGA"}}

  {"query": {"match": {"Lider": "JUAREZ"}}

  {"query": {"match": {"Lider": "HUILCA"}}

  {"query": {"match": {"Lider": "VERA"}}

  {"query": {"match": {"Lider": "LOPEZ"}}

  {"query": {"match": {"Lider": "REATEGUI"}}

  {"query": {"match": {"Lider": "ROJAS"}}

  {"query": {"match": {"Lider": "FRETEL"}}

  {"query": {"match": {"Lider": "JULCA"}}

  {"query": {"match": {"Lider": "MESTANZA"}}

  {"query": {"match": {"Lider": "REYES"}}

  {"query": {"match": {"Lider": "JIMENEZ"}}

  {"query": {"match": {"Lider": "DAVILA"}}

  {"query": {"match": {"Lider": "CAYAMPE"}}

  {"query": {"match": {"Lider": "VILLENA"}}

  {"query": {"match": {"Lider": "PINEDA"}}

  {"query": {"match": {"Lider": "ANGELES"}}

  {"query": {"match": {"Lider": "SAAVEDRA"}}
}
```

Esto nos da como resultado:

	A	B	C	D	E	F	G	H
1	AREVALO	183	REATEGUI	79	CARDENAS	91	CAYAMPE	1
2	PADILLA	1194	ROJAS	13	GONZAGA	24	VILLENA	1
3	RUIZ	144	FRETEL	18	JUAREZ	61	PINEDA	1
4	OLIVERA	11	JULCA	4	HUILCA	98	ANGELES	1
5	VELA	54	MESTANZA	7	VERA	16	SAAVEDRA	1
6	INACTIVAS	111	REYES	4	LOPEZ	12		
7	BERNA	50	JIMENEZ	2	DAVILA	15		