

## Iniciando o Elasticsearch em Docker

### Docker Desktop Windows

```
PS E:\projetos\docker-elasticsearch\elastic> wsl -l -v
NAME                STATE              VERSION
* docker-desktop-data Running            2
docker-desktop      Running            2
Ubuntu-20.04        Running            2
```

```
PS E:\projetos\docker-elasticsearch\elastic> wsl -d docker-desktop
LAPTOP-V176DRSL:/tmp/docker-desktop-root/mnt/host/e/projetos/docker-elasticsearch/elastic# sysctl -w vm.max_map_count=262144
vm.max_map_count = 262144
LAPTOP-V176DRSL:/tmp/docker-desktop-root/mnt/host/e/projetos/docker-elasticsearch/elastic#
```

### Docker Wsl2 Linux

```
feliciani@LAPTOP-V176DRSL:~$ sudo sysctl -w vm.max_map_count=262144
[sudo] password for feliciani:
vm.max_map_count = 262144
```

### Docker Desktop Windows


```
PS E:\projetos\docker-elasticsearch\elastic> docker-compose up -d
Docker Compose is now in the Docker CLI, try `docker compose up`
```




```
Starting elastic_elasticsearch_1 ... done
Starting elastic_kibana_1         ... done
Starting elastic_logstash_1       ... done
```

```
PS E:\projetos\docker-elasticsearch\elastic> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
d3d012693acc	docker.elastic.co/logstash/logstash:7.9.2	"/usr/local/bin/dock..."	17 hours ago	Up 33 minutes	0.0.0.0:5044->5044/tcp,
:::5044->5044/tcp,		elastic_logstash_1			0.0.0.0:9600->9600/tcp, :::9600->9600/tcp
ca700688aa0d	docker.elastic.co/kibana/kibana:7.9.2	"/usr/local/bin/dumb..."	17 hours ago	Up 33 minutes	0.0.0.0:5601->5601/tcp,
:::5601->5601/tcp		elastic_kibana_1			
37a2fb5958f4	docker.elastic.co/elasticsearch/elasticsearch:7.9.2	"/tini -- /usr/local..."	17 hours ago	Up 34 minutes	0.0.0.0:9200->9200/tcp,
:::9200->9200/tcp,		elastic_elasticsearch_1			9300/tcp

## Docker


 docker

Upgrade    Sign in

Containers / Apps




Images

Dev Environments


<  elastic

E:\projetos\docker-elasticsearch\elastic

Open in Visual Studio Code


  

CONTAINERS

 elastic\_logstash\_1


docker.elastic.co/logstash/logstash:7.9.2

RUNNING PORT: 5044

 elastic\_kibana\_1

docker.elastic.co/kibana/kibana:7.9.2



RUNNING PORT: 5601

 elastic\_elasticsearch\_1

docker.elastic.co/elasticsearch/elasticsearch:7.9.2

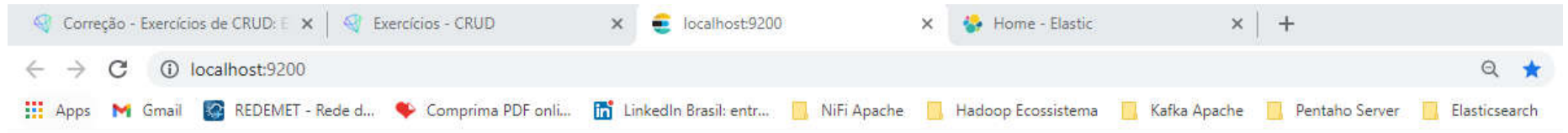
RUNNING PORT: 9200

```
dest":"empty","referrer":"http://localhost:5601/app/home","accept-encoding":"gzip, deflate, br","accept-language":"pt-BR,pt;q=0.9,en-US;q=0.8,en;q=0.7"},"remoteAddress":"172.18.0.1","userAgent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.77 Safari/537.36","referrer":"http://localhost:5601/app/home"},"res":{"statusCode":200,"responseTime":751,"contentLength":9},"message":"POST /api/ui_metric/report 200 751ms - 9.0B"}
elasticsearch_1 | {"type": "server", "timestamp": "2021-06-10T13:42:53,810Z", "level": "INFO", "component": "o.e.m.j.JvmGcMonitorService", "cluster.name": "my_cluster", "node.name": "node1", "message": "[gc][1024] overhead, spent [304ms] collecting in the last [1s]", "cluster.uuid": "O_RNH51SQVYOsVZ61x-zMg", "node.id": "IOMovYK2T0eNK1jQUwWviA" }
elasticsearch_1 | {"type": "server", "timestamp": "2021-06-10T13:50:02,818Z", "level": "WARN", "component": "o.e.m.f.FsHealthService", "cluster.name": "my_cluster", "node.name": "node1", "message": "health check of [/usr/share/elasticsearch/data/nodes/0] took [5569ms] which is above the warn threshold of [5s]", "cluster.uuid": "O_RNH51SQVYOsVZ61x-zMg", "node.id": "IOMovYK2T0eNK1jQUwWviA" }
elasticsearch_1 | {"type": "server", "timestamp": "2021-06-10T13:50:12,644Z", "level": "WARN", "component": "o.e.m.j.JvmGcMonitorService", "cluster.name": "my_cluster", "node.name": "node1", "message": "[gc][young][1454][20] duration [2.5s], collections [1]/[3.1s], total [2.5s]/[4.5s], memory [367.7mb]->[86.2mb]/[512mb], all_pools {[young] 282mb->[0b]/[0b]}{[old] 76.2mb->[76.2mb]/[512mb]}{[survivor] 9.4mb->[10mb]/[0b]}", "cluster.uuid": "O_RNH51SQVYOsVZ61x-zMg", "node.id": "IOMovYK2T0eNK1jQUwWviA" }
elasticsearch_1 | {"type": "server", "timestamp": "2021-06-10T13:50:12,645Z", "level": "WARN", "component": "o.e.m.j.JvmGcMonitorService", "cluster.name": "my_cluster", "node.name": "node1", "message": "[gc][1454] overhead, spent [2.5s] collecting in the last [3.1s]", "cluster.uuid": "O_RNH51SQVYOsVZ61x-zMg", "node.id": "IOMovYK2T0eNK1jQUwWviA" }
```

Search...  Stick to bottom 

## Acessado o Elasticsearch

<http://localhost:9200>



```
{
  "name" : "node1",
  "cluster_name" : "my_cluster",
  "cluster_uuid" : "O_RNH51SQVyOsVZ61x-zMg",
  "version" : {
    "number" : "7.9.2",
    "build_flavor" : "default",
    "build_type" : "docker",
    "build_hash" : "d34da0ea4a966c4e49417f2da2f244e3e97b4e6e",
    "build_date" : "2020-09-23T00:45:33.626720Z",
    "build_snapshot" : false,
    "lucene_version" : "8.6.2",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

## Acessando o KIBANA

<http://localhost:5601>

The screenshot shows a web browser with multiple tabs. The active tab is 'localhost:5601/app/home#/'. The browser's address bar shows the URL. Below the browser window, the Kibana home page is displayed. The page has a sidebar on the left with a menu. The main content area is divided into several sections: 'Recently viewed' (empty), 'Logs' (with an 'Add log data' button), 'Metrics' (with an 'Add metric data' button), 'Security' (with an 'Add events' button), 'Add sample data' (with a button to 'Set and a Kibana dashboard'), 'Upload data from log file' (with a button to 'Import a CSV, NDJSON, or log file'), 'Use Elasticsearch data' (with a button to 'Connect to your Elasticsearch index'), 'Explore Data' (with an 'App Search' button), and 'Manage and Administer the Elastic Stack' (with buttons for 'Console' and 'Rollups').

Correção - Exercícios de CRUD: E x Exercícios - CRUD x localhost:9200 x Home - Elastic x +

localhost:5601/app/home#/

Apps Gmail REDEMET - Rede d... Comprima PDF onli... LinkedIn Brasil: entr... NiFi Apache Hadoop Ecosystema Kafka Apache Pentaho Server Elasticsearch

Home

Home

Recently viewed

No recently viewed items

Kibana

Discover

Dashboard

Canvas

Maps

Machine Learning

Visualize

Enterprise Search

App Search

Workplace Search

Observability

Overview

Logs

Availability

Logs

Ingest logs from popular data sources and easily visualize in preconfigured dashboards.

Add log data

Metrics

Collect metrics from the operating system and services running on your servers.

Add metric data

Security

SIEM + Endpoint Security

Protect hosts, analyze security information and events, hunt threats, automate detections, and create cases.

Add events

Add sample data

Set and a Kibana dashboard

Upload data from log file

Import a CSV, NDJSON, or log file

Use Elasticsearch data

Connect to your Elasticsearch index

Explore Data

App Search

Leverage dashboards, analytics, and APIs for advanced application

Manage and Administer the Elastic Stack

Console

Skip cURL and use this JSON interface to work with your data directly.

Rollups

Summarize and store historical data in a smaller index for future analysis.

## Exercitando Analyzer

### Acessar o MENU -> Dev Tools

The screenshot displays the Kibana web interface in a browser window. The browser's address bar shows the URL `localhost:5601/app/home#`. The Kibana sidebar on the left contains the following menu structure:

- Home
- Recently viewed (No recently viewed items)
- Security
  - Overview
  - Detections
  - Hosts
  - Network
  - Timelines
  - Cases
  - Administration
- Management
  - Dev Tools (highlighted)
  - Ingest Manager
  - Stack Monitoring
  - Stack Management

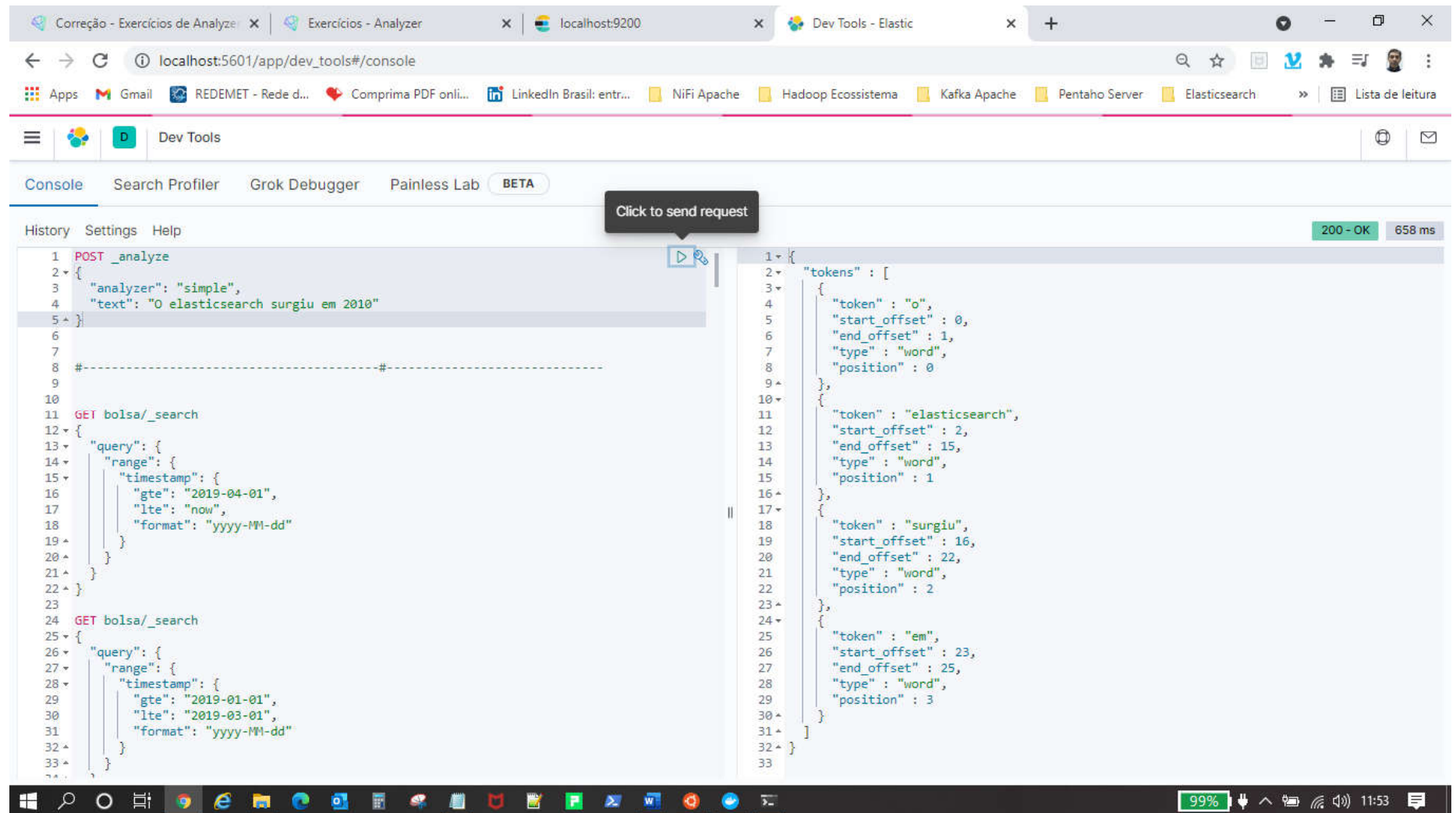
The main content area of the Kibana interface shows various tiles for data management and security, including sections for Logs, Metrics, Security, and Explore Data. The Windows taskbar at the bottom indicates the system time as 11:18 and shows a battery level of 99%.



1. Criar os Analyzer simple, standard, brazilian e portuguese para a seguinte frase:

- O elasticsearch surgiu em 2010

O simple deixa tudo minúsculo, remove os números. Por exemplo, 2010 não existe no print abaixo



The screenshot shows a web browser window with the DevTools console open. The console displays a POST request to the Elasticsearch API and its corresponding JSON response.

**Request:**

```
1 POST _analyze
2 {
3   "analyzer": "simple",
4   "text": "O elasticsearch surgiu em 2010"
5 }
6
7 -----
8
9
10
11 GET bolsa/_search
12 {
13   "query": {
14     "range": {
15       "timestamp": {
16         "gte": "2019-04-01",
17         "lte": "now",
18         "format": "yyyy-MM-dd"
19       }
20     }
21   }
22 }
23
24 GET bolsa/_search
25 {
26   "query": {
27     "range": {
28       "timestamp": {
29         "gte": "2019-01-01",
30         "lte": "2019-03-01",
31         "format": "yyyy-MM-dd"
32       }
33     }
34   }
35 }
```

**Response:**

```
1 {
2   "tokens": [
3     {
4       "token": "o",
5       "start_offset": 0,
6       "end_offset": 1,
7       "type": "word",
8       "position": 0
9     },
10    {
11      "token": "elasticsearch",
12      "start_offset": 2,
13      "end_offset": 15,
14      "type": "word",
15      "position": 1
16    },
17    {
18      "token": "surgiu",
19      "start_offset": 16,
20      "end_offset": 22,
21      "type": "word",
22      "position": 2
23    },
24    {
25      "token": "em",
26      "start_offset": 23,
27      "end_offset": 25,
28      "type": "word",
29      "position": 3
30    }
31  ]
32 }
33
```

The response shows the tokens extracted from the text "O elasticsearch surgiu em 2010". The tokens are "o", "elasticsearch", "surgiu", and "em". The "position" field indicates the starting index of each token in the original text.

Com standard, todas as letras ficam minúsculas e mantém os números

The screenshot shows a web browser window with the address bar at `localhost:5601/app/dev_tools#/console`. The browser tabs include "Correção - Exercícios de Analyzer", "Exercícios - Analyzer", "localhost:9200", and "Dev Tools - Elastic". The browser's top bar shows various extensions and a search bar. The DevTools interface is open, showing the "Console" tab. The console has a "History" section on the left and a "Settings" section on the right. The main console area displays a series of API calls and their responses.

**Console History:**

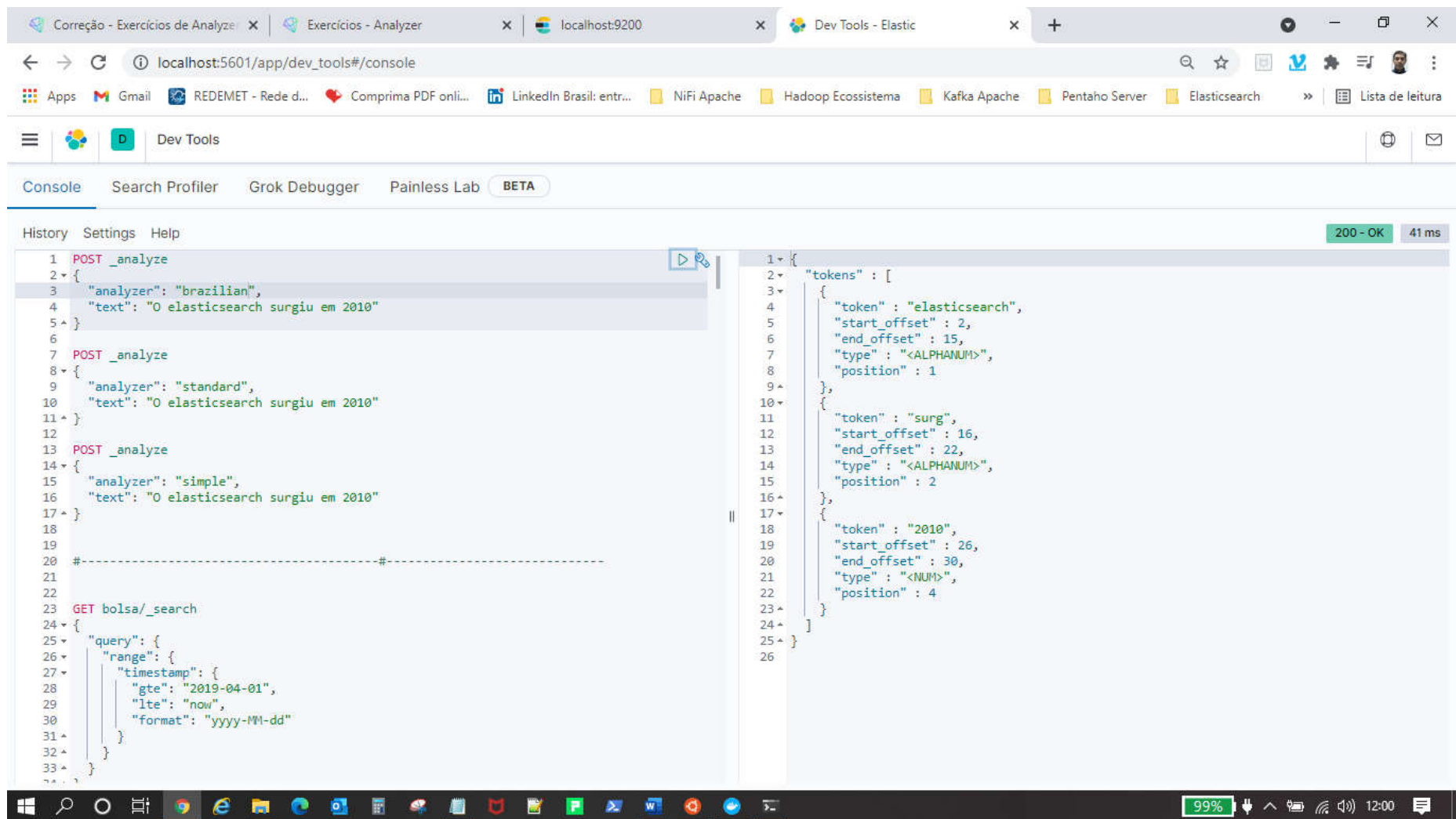
- 1 `POST _analyze`
- 2 `{`
- 3  `"analyzer": "standard",`
- 4  `"text": "O elasticsearch surgiu em 2010"`
- 5 `}`
- 6
- 7 `POST _analyze`
- 8 `{`
- 9  `"analyzer": "simple",`
- 10  `"text": "O elasticsearch surgiu em 2010"`
- 11 `}`
- 12
- 13 `#-----#`
- 14
- 15
- 16
- 17 `GET bolsa/_search`
- 18 `{`
- 19  `"query": {`
- 20  `"range": {`
- 21  `"timestamp": {`
- 22  `"gte": "2019-04-01",`
- 23  `"lte": "now",`
- 24  `"format": "yyyy-MM-dd"`
- 25  `}`
- 26  `}`
- 27  `}`
- 28 `}`
- 29
- 30 `GET bolsa/_search`
- 31 `{`
- 32  `"query": {`
- 33  `"range": {`
- 34  `"timestamp": {`
- 35  `"gte": "2019-04-01",`
- 36  `"lte": "now",`
- 37  `"format": "yyyy-MM-dd"`
- 38  `}`
- 39  `}`
- 40  `}`
- 41 `}`

**Console Output:**

```
{
  "tokens": [
    {
      "token": "O",
      "start_offset": 0,
      "end_offset": 1,
      "type": "<ALPHANUM>",
      "position": 0
    },
    {
      "token": "elasticsearch",
      "start_offset": 2,
      "end_offset": 15,
      "type": "<ALPHANUM>",
      "position": 1
    },
    {
      "token": "surgiu",
      "start_offset": 16,
      "end_offset": 22,
      "type": "<ALPHANUM>",
      "position": 2
    },
    {
      "token": "em",
      "start_offset": 23,
      "end_offset": 25,
      "type": "<ALPHANUM>",
      "position": 3
    },
    {
      "token": "2010",
      "start_offset": 26,
      "end_offset": 30,
      "type": "<NUM>",
      "position": 4
    }
  ]
}
```

The bottom of the screen shows the Windows taskbar with various application icons and a system tray indicating 99% battery and the time 11:57.

Com o analyzer brazilian, retirou o “o”, “em” e reduziu a palavra surgiu, deixando na forma raiz



The screenshot shows the Elastic Dev Tools console with a POST \_analyze request and its response. The request uses the 'brazilian' analyzer on the text 'O elasticsearch surgiu em 2010'. The response shows the tokens: 'elasticsearch', 'surg', '2010', with their respective offsets and positions. The 'o' and 'em' are removed, and 'surgiu' is reduced to 'surg'.

```
1 POST _analyze
2 {
3   "analyzer": "brazilian",
4   "text": "O elasticsearch surgiu em 2010"
5 }
6
7 POST _analyze
8 {
9   "analyzer": "standard",
10  "text": "O elasticsearch surgiu em 2010"
11 }
12
13 POST _analyze
14 {
15   "analyzer": "simple",
16   "text": "O elasticsearch surgiu em 2010"
17 }
18
19 #-----#
20
21
22 GET bolsa/_search
23 {
24   "query": {
25     "range": {
26       "timestamp": {
27         "gte": "2019-04-01",
28         "lte": "now",
29         "format": "yyyy-MM-dd"
30       }
31     }
32   }
33 }
```

```
1 {
2   "tokens": [
3     {
4       "token": "elasticsearch",
5       "start_offset": 2,
6       "end_offset": 15,
7       "type": "<ALPHANUM>",
8       "position": 1
9     },
10    {
11      "token": "surg",
12      "start_offset": 16,
13      "end_offset": 22,
14      "type": "<ALPHANUM>",
15      "position": 2
16    },
17    {
18      "token": "2010",
19      "start_offset": 26,
20      "end_offset": 30,
21      "type": "<NUM>",
22      "position": 4
23    }
24  ]
25 }
```

200 - OK 41 ms



O analyzer portuguese, removeu o “o”, “em” e as outras palavras ficaram inteiras como no português

The screenshot shows a web browser window with the address bar at `localhost:5601/app/dev_tools#/console`. The browser tabs include "Correção - Exercícios de Analyzer", "Exercícios - Analyzer", "localhost:9200", and "Dev Tools - Elastic". The browser's bookmark bar shows various links like "Apps", "Gmail", "REDEMET - Rede d...", "Comprima PDF onli...", "LinkedIn Brasil: entr...", "NiFi Apache", "Hadoop Ecosystema", "Kafka Apache", "Pentaho Server", "Elasticsearch", and "Lista de leitura".

The DevTools console is open, showing a "History" tab with a list of requests. The first request is a `POST _analyze` with the following body:

```
1 POST _analyze
2 {
3   "analyzer": "portuguese",
4   "text": "O elasticsearch surgiu em 2010"
5 }
```

The response for this request is a JSON object with a `tokens` array:

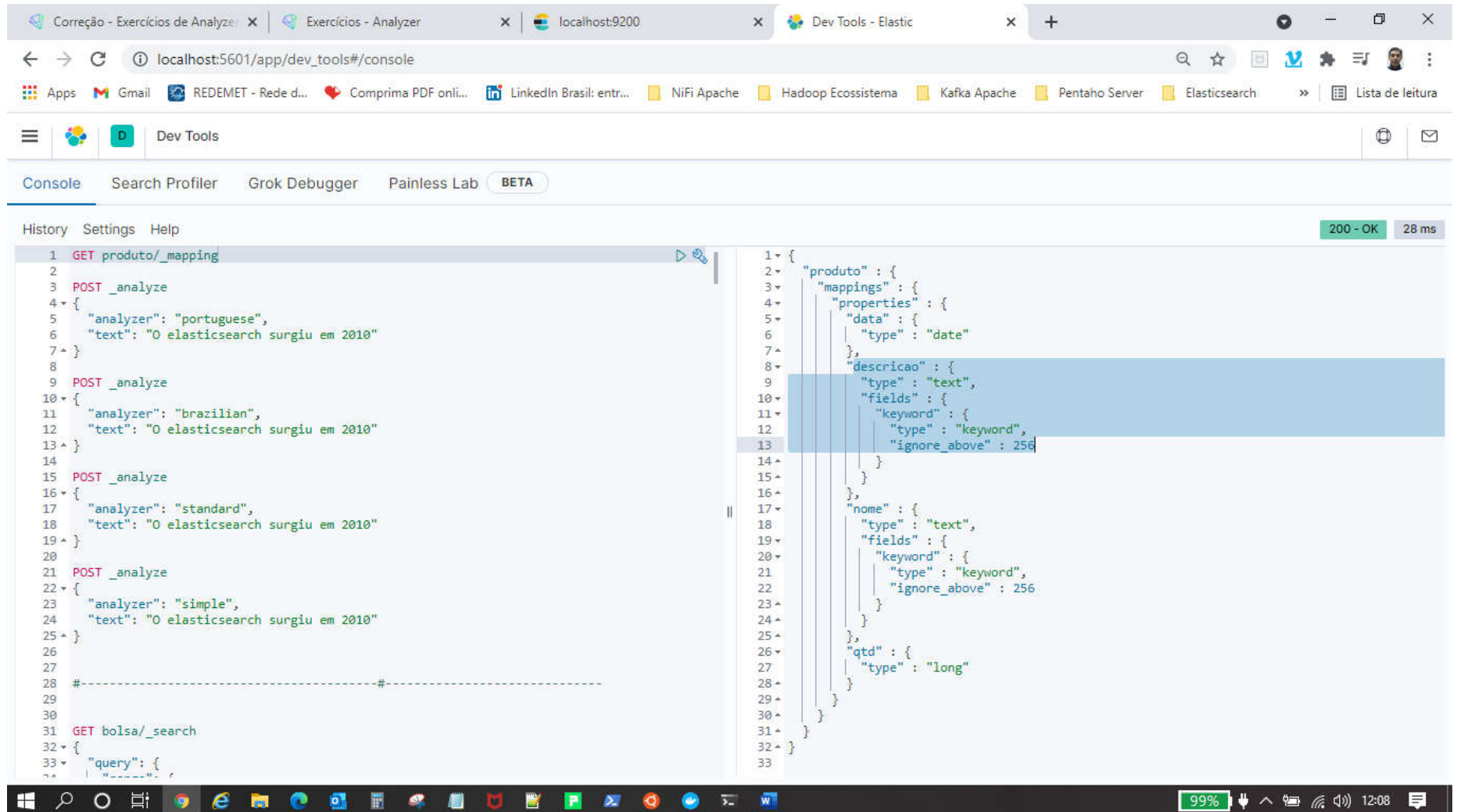
```
1 {
2   "tokens": [
3     {
4       "token": "elasticsearch",
5       "start_offset": 2,
6       "end_offset": 15,
7       "type": "<ALPHANUM>",
8       "position": 1
9     },
10    {
11      "token": "surgiu",
12      "start_offset": 16,
13      "end_offset": 22,
14      "type": "<ALPHANUM>",
15      "position": 2
16    },
17    {
18      "token": "2010",
19      "start_offset": 26,
20      "end_offset": 30,
21      "type": "<NUM>",
22      "position": 4
23    }
24  ]
25 }
```

The console also shows a `GET bolsa/_search` request and its response. The Windows taskbar at the bottom shows the system clock at 12:03 and 99% battery.

## 2. Realizar os passos no índice produto

### a) Criar um analyzer brazilian para o atributo descricao

Verificando que não existe analyzer para o atributo descricao



The screenshot shows a web browser window with the DevTools console open. The console displays a series of REST client requests and a JSON response from Elasticsearch.

**Requests:**

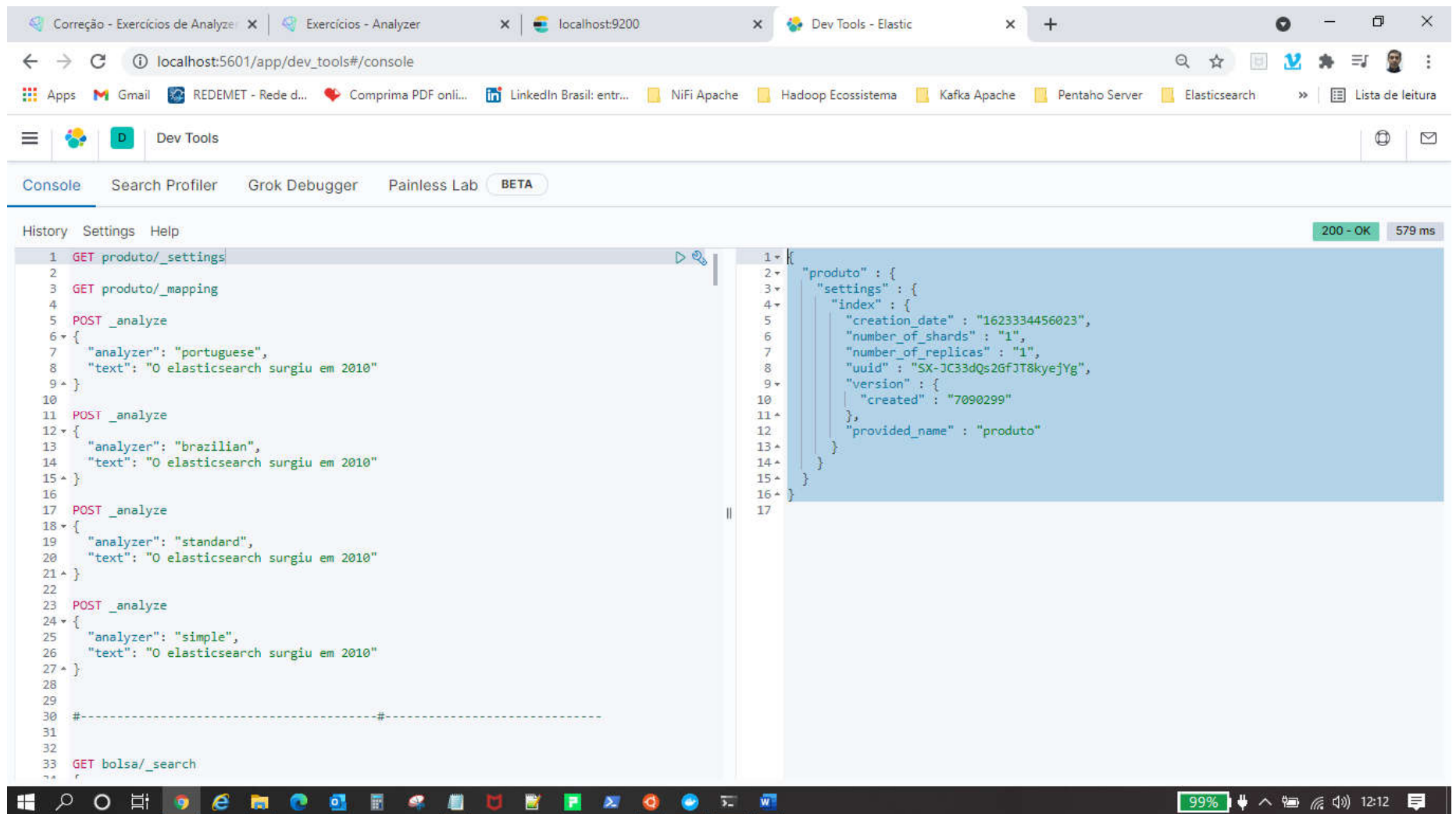
- 1 GET produto/\_mapping
- 3 POST \_analyze
- 4 {  
5 "analyzer": "portuguese",  
6 "text": "O elasticsearch surgiu em 2010"  
7 }  
8
- 9 POST \_analyze
- 10 {  
11 "analyzer": "brazilian",  
12 "text": "O elasticsearch surgiu em 2010"  
13 }  
14
- 15 POST \_analyze
- 16 {  
17 "analyzer": "standard",  
18 "text": "O elasticsearch surgiu em 2010"  
19 }  
20
- 21 POST \_analyze
- 22 {  
23 "analyzer": "simple",  
24 "text": "O elasticsearch surgiu em 2010"  
25 }  
26
- 27 -----
- 28
- 29
- 30
- 31 GET bolsa/\_search
- 32 {  
33 "query": {  
34 "text": "O elasticsearch surgiu em 2010"

**Response:**

```
{  
  "produto": {  
    "mappings": {  
      "properties": {  
        "data": {  
          "type": "date"  
        },  
        "descricao": {  
          "type": "text",  
          "fields": {  
            "keyword": {  
              "type": "keyword",  
              "ignore_above": 256  
            }  
          }  
        },  
        "nome": {  
          "type": "text",  
          "fields": {  
            "keyword": {  
              "type": "keyword",  
              "ignore_above": 256  
            }  
          }  
        },  
        "qtd": {  
          "type": "long"  
        }  
      }  
    }  
  }  
}
```

The response status is 200 - OK and the response time is 28 ms.

## Verificar as configurações para reindexar



## Criando o Index produto1 com o analyzer brazilian

The screenshot shows the Elastic DevTools console interface. The browser address bar displays `localhost:5601/app/dev_tools#/console`. The console has tabs for **Console**, **Search Profiler**, **Grok Debugger**, **Painless Lab**, and **BETA**. The **Console** tab is active, showing a list of commands and their results.

A tooltip **Click to send request** is visible over the **Send** button (a blue play icon) in the command editor.

The command list on the left includes:

- 1 **PUT** produto1
- 2 {
- 3 "settings": {
- 4 "index": {
- 5 "number\_of\_shards": "1",
- 6 "number\_of\_replicas": "0"
- 7 }
- 8 },
- 9 "mappings": {
- 10 "properties": {
- 11 "descricao": {
- 12 "type": "text",
- 13 "analyzer": "brazilian"
- 14 }
- 15 }
- 16 }
- 17 }
- 18 }
- 19 **GET** produto/\_settings
- 20
- 21 **GET** produto/\_mapping
- 22
- 23 **POST** \_analyze
- 24 {
- 25 "analyzer": "portuguese",
- 26 "text": "O elasticsearch surgiu em 2010"
- 27 }
- 28
- 29 **POST** \_analyze
- 30 {
- 31 "analyzer": "brazilian",
- 32 "text": "O elasticsearch surgiu em 2010"
- 33 }
- 34 }

The result pane on the right shows the response for the first **PUT** command:

```
1 {
2   "acknowledged": true,
3   "shards_acknowledged": true,
4   "index": "produto1"
5 }
6
```

At the top right of the result pane, the status **200 - OK** and the execution time **14577 ms** are displayed.

## Faz reindex do produto com produto1

The screenshot shows a web browser window with the URL `localhost:5601/app/dev_tools#/console`. The DevTools console is open, displaying a REST client request and its response.

**Request:**

```
1 POST _reindex
2 {
3   "source": {
4     "index": "produto"
5   },
6   "dest": {
7     "index": "produto1"
8   }
9 }
10
11 PUT produto1
12 {
13   "settings": {
14     "index": {
15       "number_of_shards": "1",
16       "number_of_replicas": "0"
17     }
18   },
19   "mappings": {
20     "properties": {
21       "descricao": {
22         "type": "text",
23         "analyzer": "brazilian"
24       }
25     }
26   }
27 }
28
29 GET produto/_settings
30
31 GET produto/_mapping
32
33 POST _analyze
```

**Response:**

```
1 {
2   "took" : 12159,
3   "timed_out" : false,
4   "total" : 4,
5   "updated" : 0,
6   "created" : 4,
7   "deleted" : 0,
8   "batches" : 1,
9   "version_conflicts" : 0,
10  "noops" : 0,
11  "retries" : {
12    "bulk" : 0,
13    "search" : 0
14  },
15  "throttled_millis" : 0,
16  "requests_per_second" : -1.0,
17  "throttled_until_millis" : 0,
18  "failures" : [ ]
19 }
20
```

The response status is **200 - OK** and the execution time is **12338 ms**. A tooltip "Click to send request" is visible over the send button in the REST client interface.



## Produto1 está com analyzer brazilian e type como text

The screenshot shows a web browser window with the URL `localhost:5601/app/dev_tools#/console`. The browser tabs include "Correção - Exercícios de Analyzer", "Exercícios - Analyzer", "localhost:9200", and "Dev Tools - Elastic". The browser's address bar shows the URL. Below the browser window, the DevTools console is open, displaying the "Console" tab. The console shows a series of commands and their results:

```
1 GET produto/_mapping
2
3 POST _reindex
4 {
5   "source": {
6     "index": "produto"
7   },
8   "dest": {
9     "index": "produto1"
10  }
11 }
12
13 PUT produto1
14 {
15   "settings": {
16     "index": {
17       "number_of_shards": "1",
18       "number_of_replicas": "0"
19     }
20   },
21   "mappings": {
22     "properties": {
23       "descricao": {
24         "type": "text",
25         "analyzer": "brazilian"
26       }
27     }
28   }
29 }
30
31 GET produto/_settings
32
33 GET produto/_mapping
```

The right side of the console shows the resulting mapping for the `produto1` index:

```
1 {
2   "produto1" : {
3     "mappings" : {
4       "properties" : {
5         "data" : {
6           "type" : "date"
7         },
8         "descricao" : {
9           "type" : "text",
10          "analyzer" : "brazilian"
11        },
12        "nome" : {
13          "type" : "text",
14          "fields" : {
15            "keyword" : {
16              "type" : "keyword",
17              "ignore_above" : 256
18            }
19          }
20        },
21        "qtd" : {
22          "type" : "long"
23        }
24      }
25    }
26  }
27 }
```

The console status bar at the bottom right indicates "200 - OK" and "33 ms". The Windows taskbar is visible at the bottom of the screen, showing the time as 12:22 and a battery level of 99%.

Para deixar como produto. Devo deletar o produto, pegar produto1 que está correto e reindexar novamente para produto

b) Para o atributo descricao aplicar o analyzer brazilian para o tipo de campo text e criar o atributo descricao.original com o dado do tipo keyword

## Deletando o produto

The screenshot shows a web browser window with the DevTools console open. The browser's address bar shows `localhost:5601/app/dev_tools#/console`. The console has tabs for Console, Search Profiler, Grok Debugger, and Painless Lab (marked BETA). The Console tab is active, showing a list of REST API commands and their responses.

**Commands and Responses:**

- `DELETE produto` (Line 1): Response: `{ "acknowledged": true }` (Line 1-3). Status: 200 - OK, 1668 ms.
- `GET produto/_mapping` (Line 3): No response shown.
- `POST _reindex` (Line 5): No response shown.
- `PUT produto/_settings` (Line 15): No response shown.

**Index Mapping (GET produto/\_mapping):**

```
{
  "produto": {
    "properties": {
      "descricao": {
        "type": "text",
        "analyzer": "brazilian"
      }
    }
  }
}
```

**Index Settings (PUT produto/\_settings):**

```
{
  "settings": {
    "index": {
      "number_of_shards": "1",
      "number_of_replicas": "0"
    }
  }
}
```

## Produto criado novamente

The screenshot shows the Elastic Dev Tools console in a web browser. The browser tabs include 'Correção - Exercícios de Analyzer', 'Exercícios - Analyzer', 'localhost:9200', and 'Dev Tools - Elastic'. The address bar shows 'localhost:5601/app/dev\_tools#/console'. The console interface has tabs for 'Console', 'Search Profiler', 'Grok Debugger', 'Painless Lab', and 'BETA'. The 'Console' tab is active, showing a list of commands and their responses.

**History** **Settings** **Help**

**200 - OK** **4913 ms**

```
1 PUT produto
2 {
3   "settings": {
4     "index": {
5       "number_of_shards": "1",
6       "number_of_replicas": "0"
7     }
8   },
9   "mappings": {
10    "properties": {
11      "descricao": {
12        "type": "text",
13        "analyzer": "brazilian"
14      }
15    }
16  }
17 }
18
19 DELETE produto
20
21 GET produto/_mapping
22
23 POST _reindex
24 {
25   "source": {
26     "index": "produto"
27   },
28   "dest": {
29     "index": "produto1"
30   }
31 }
32
33 PUT produto1
```

The console shows the execution of several Elasticsearch commands. The first command is a PUT request to create an index named 'produto' with specific settings and mappings. The response is a 200 OK status with a 4913 ms execution time. The subsequent commands include deleting the index, retrieving its mapping, performing a reindex operation, and creating a new index named 'produto1'.

## Reindex de produto1 para produto

The screenshot shows the Elastic Dev Tools console in a web browser. The browser tabs include 'Correção - Exercícios de Analyzer', 'Exercícios - Analyzer', 'localhost:9200', and 'Dev Tools - Elastic'. The address bar shows 'localhost:5601/app/dev\_tools#/console'. The console has tabs for 'Console', 'Search Profiler', 'Grok Debugger', 'Painless Lab', and 'BETA'. The 'Console' tab is active, showing a list of requests and a detailed view of the selected request.

**History:**

- 1 POST \_reindex
- 2 {
- 3 "source": {
- 4 "index": "produto1"
- 5 },
- 6 "dest": {
- 7 "index": "produto"
- 8 }
- 9 }
- 10
- 11 PUT produto
- 12 {
- 13 "settings": {
- 14 "index": {
- 15 "number\_of\_shards": "1",
- 16 "number\_of\_replicas": "0"
- 17 }
- 18 },
- 19 "mappings": {
- 20 "properties": {
- 21 "descricao": {
- 22 "type": "text",
- 23 "analyzer": "brazilian"
- 24 }
- 25 }
- 26 }
- 27 }
- 28
- 29 DELETE produto
- 30
- 31 GET produto1/\_mapping
- 32
- 33 POST \_reindex
- 34

**Click to send request**

**200 - OK 2679 ms**

**Request:**

```
1 POST _reindex
2 {
3   "source": {
4     "index": "produto1"
5   },
6   "dest": {
7     "index": "produto"
8   }
9 }
```

**Response:**

```
1 {
2   "took" : 2518,
3   "timed_out" : false,
4   "total" : 4,
5   "updated" : 0,
6   "created" : 4,
7   "deleted" : 0,
8   "batches" : 1,
9   "version_conflicts" : 0,
10  "noops" : 0,
11  "retries" : {
12    "bulk" : 0,
13    "search" : 0
14  },
15  "throttled_millis" : 0,
16  "requests_per_second" : -1.0,
17  "throttled_until_millis" : 0,
18  "failures" : [ ]
19 }
20
```

The Windows taskbar at the bottom shows the system clock as 12:38 and 99% battery.

### c) Buscar a palavra “compatível” no campo descricao.original (hits = 0)

Não retornou nada porque está com o token compat

The screenshot shows the Elastic Dev Tools console with a search query and its response. The query is a GET request to `produto/_search` with a match query on `descricao.original` for the value `"compatível"`. The response shows that the search was successful but returned 0 hits because the word `"compatível"` is tokenized into `["compat", "ível"]`, and no document contains both tokens.

```
1 GET produto/_search
2 {
3   "query": {
4     "match": {
5       "descricao.original": "compatível"
6     }
7   }
8 }
9
10
11
12 PUT produto
13 {
14   "settings": {
15     "index": {
16       "number_of_shards": "1",
17       "number_of_replicas": "0"
18     }
19   },
20   "mappings": {
21     "properties": {
22       "descricao": {
23         "type": "text",
24         "analyzer": "brazilian",
25         "fields": {
26           "original": {"type": "keyword"}
27         }
28       }
29     }
30   }
31 }
32
33 POST _reindex
```

```
1 {
2   "took": 0,
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": 0,
13      "relation": "eq"
14    },
15    "max_score": null,
16    "hits": [ ]
17  }
18 }
19
```

200 - OK 50 ms



#### d) Buscar a palavra “compatível” no campo descrição

#### Encontrou por que o token está compatível

The screenshot shows the Elastic Dev Tools console interface. The left pane displays the search query and the index settings. The right pane shows the search results.

**Search Query:**

```
1 GET produto/_search
2 {
3   "query": {
4     "match": {
5       "descricao": "compatível"
6     }
7   }
8 }
9
10
11
12 PUT produto
13 {
14   "settings": {
15     "index": {
16       "number_of_shards": "1",
17       "number_of_replicas": "0"
18     }
19   },
20   "mappings": {
21     "properties": {
22       "descricao": {
23         "type": "text",
24         "analyzer": "brazilian",
25         "fields": {
26           "original": {"type": "keyword"}
27         }
28       }
29     }
30   }
31 }
32
33 POST _reindex
```

**Search Results:**

```
1 {
2   "took" : 157,
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" : 1.1129161,
16    "hits" : [
17      {
18        "_index" : "produto",
19        "_type" : "_doc",
20        "_id" : "1",
21        "_score" : 1.1129161,
22        "_source" : {
23          "nome" : "mouse",
24          "qtd" : 50,
25          "descricao" : "com fio USB, compatível com Windows, Mac e Linux"
26        }
27      }
28    ]
29  }
30 }
31
```

The status bar at the bottom right indicates 200 - OK and 181 ms.