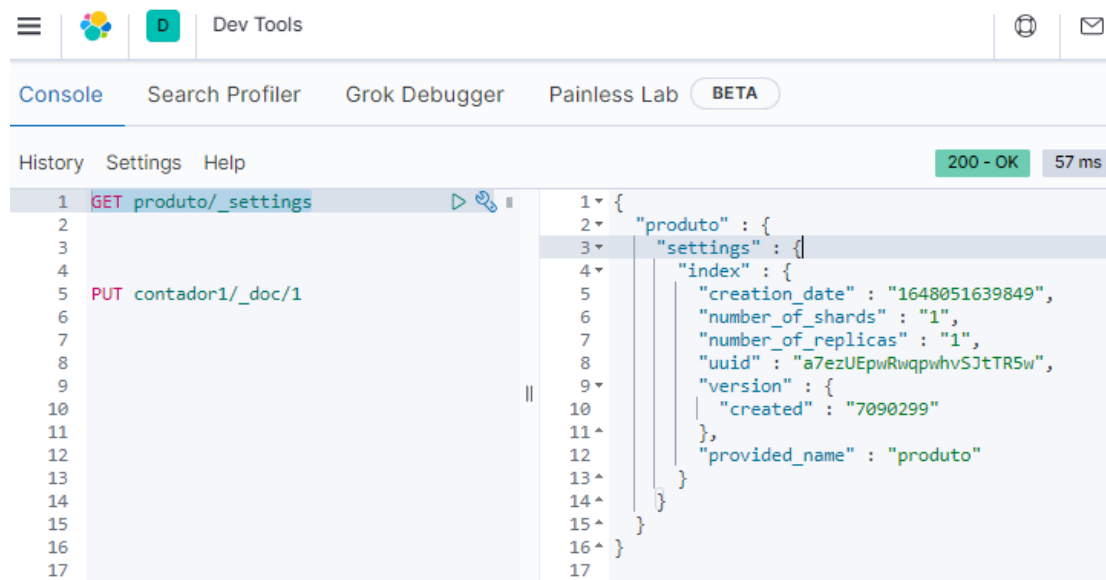


Exerc-Aula03-Elastic -Índices

1. Visualizar as configurações do índice produto

Resposta : GET produto/_settings

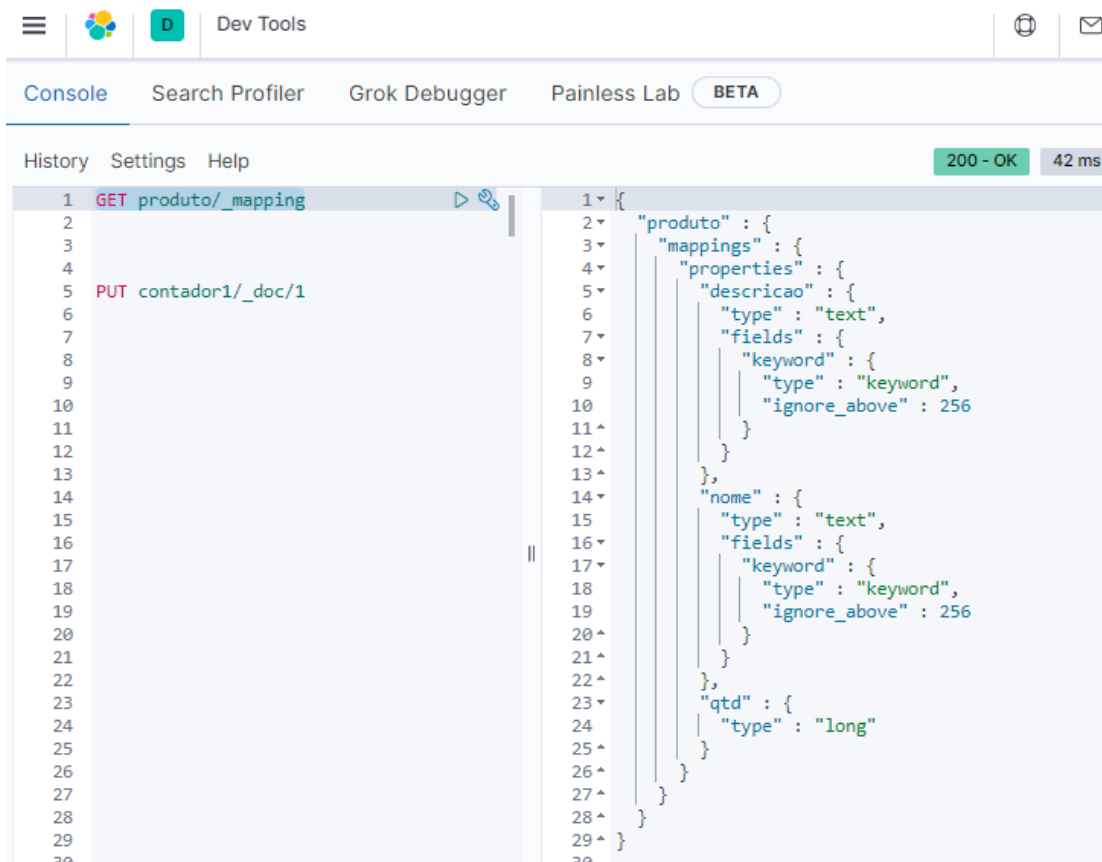


The screenshot shows the DevTools console with the 'Console' tab selected. The top bar indicates '200 - OK' and '57 ms'. The console history shows a GET request to 'produto/_settings' at line 1 and a PUT request to 'contador1/_doc/1' at line 5. The response for the GET request is a JSON object with the following structure:

```
1 {
2   "produto": {
3     "settings": {
4       "index": {
5         "creation_date": "1648051639849",
6         "number_of_shards": "1",
7         "number_of_replicas": "1",
8         "uuid": "a7ezUEpwRwqpwhvSJtTR5w",
9         "version": {
10          "created": "7090299"
11        },
12        "provided_name": "produto"
13      }
14    }
15  }
16 }
```

2. Visualizar o mapeamento do índice produto

** Resposta:GET produto/_mapping



3. Visualizar o mapeamento do atributo nome do índice produto

** Resposta: GET produto/_mapping/field/nome

The screenshot shows the DevTools Console with the 'Console' tab selected. The top bar indicates '200 - OK' and '33 ms'. The request is a GET to `produto/_mapping/field/nome`. The response is a JSON object:

```
1 {
2   "produto": {
3     "mappings": {
4       "nome": {
5         "full_name": "nome",
6         "mapping": {
7           "nome": {
8             "type": "text",
9             "fields": {
10              "keyword": {
11                "type": "keyword",
12                "ignore_above": 256
13              }
14            }
15          }
16        }
17      }
18    }
19  }
20 }
```

4. Inserir o campo data do tipo date no índice produto

** Resposta:

PUT produto/_mapping

```
{  "properties": {
    "data": {
      "type": "date"
    }
  }
}
```

The screenshot shows the DevTools Console with the 'Console' tab selected. The top bar indicates '200 - OK' and '93 ms'. The request is a PUT to `produto/_mapping` with the following JSON body:

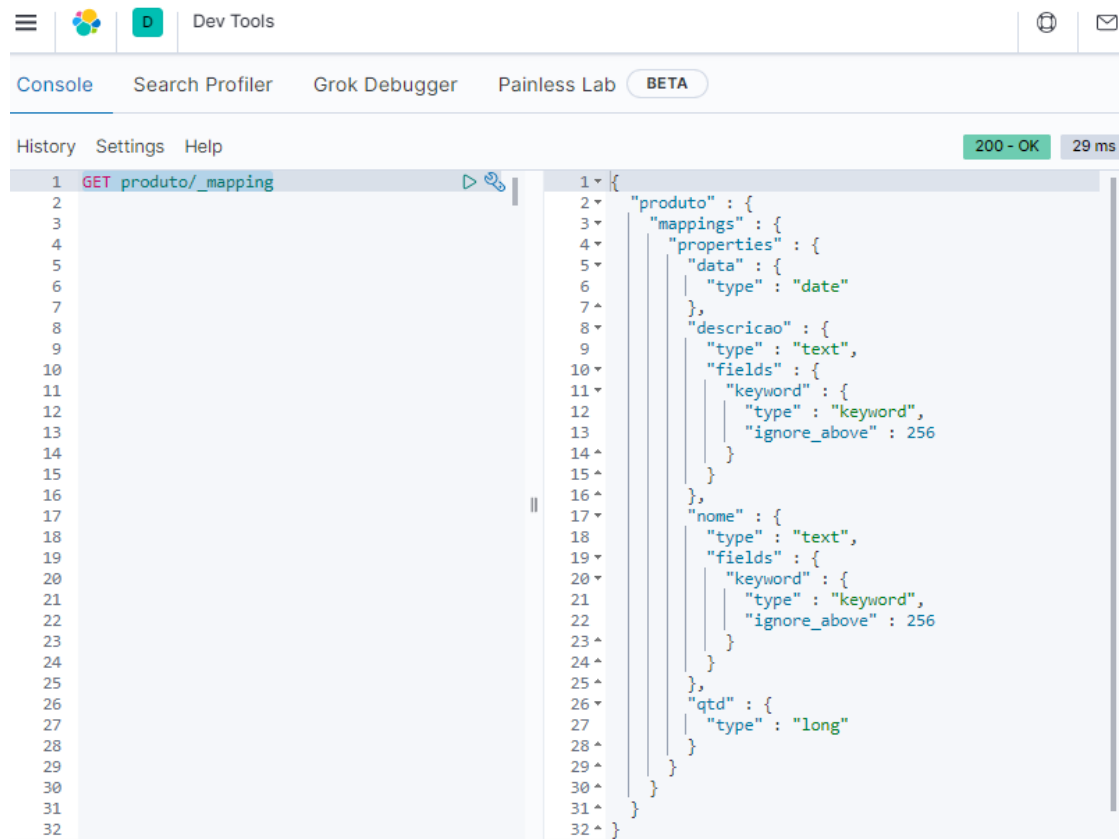
```
1 PUT produto/_mapping
2 {
3   "properties": {
4     "data": {
5       "type": "date"
6     }
7   }
8 }
```

The response is a JSON object:

```
1 {
2   "acknowledged": true
3 }
```

*** Conferindo

GET produto/_mapping



```
1 GET produto/_mapping
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
```

```
1 {
2   "produto": {
3     "mappings": {
4       "properties": {
5         "data": {
6           "type": "date"
7         },
8         "descricao": {
9           "type": "text",
10          "fields": {
11            "keyword": {
12              "type": "keyword",
13              "ignore_above": 256
14            }
15          }
16        },
17        "nome": {
18          "type": "text",
19          "fields": {
20            "keyword": {
21              "type": "keyword",
22              "ignore_above": 256
23            }
24          }
25        },
26        "qtd": {
27          "type": "long"
28        }
29      }
30    }
31  }
32 }
```

200 - OK 29 ms

5. Adicionar o documento:

_id: 6, "nome": "teclado", "qtd": 100, "descricao": "USB", "data": "2020-09-18"

** Resposta

PUT produto/_doc/6

```
{
  "nome": "teclado",
  "qtd": 100,
  "descricao": "USB",
```

```
"data":"2020-09-18"
}
```

```
1 PUT produto/_doc/6
2 {
3   "nome": "teclado",
4   "qtd": 100,
5   "descricao": "USB",
6   "data": "2020-09-18"
7 }
8
9
10
11
12
13
14
15
```

```
1 {
2   "_index" : "produto",
3   "_type" : "_doc",
4   "_id" : "6",
5   "_version" : 1,
6   "result" : "created",
7   "_shards" : {
8     "total" : 2,
9     "successful" : 1,
10    "failed" : 0
11  },
12   "_seq_no" : 7,
13   "_primary_term" : 4
14 }
15
```

6. Reindexar o índice produto para produto2, com o campo quantidade para o tipo short

**** Resposta**

Dica: Vamos olhar o mapeamento e perceber que a qtd está como long e o elastic NÃO permite mudar diretamente esse campo. Existe varias formas de resolver. Porém vamos criar uma "cópia com as estrutura desejada e reindexar.

1º Visualizar o mapeamento

GET produto/_mapping

```

1 {
2   "produto" : {
3     "mappings" : {
4       "properties" : {
5         "data" : {
6           "type" : "date"
7         },
8         "descricao" : {
9           "type" : "text",
10          "fields" : {
11            "keyword" : {
12              "type" : "keyword",
13              "ignore_above" : 256
14            }
15          }
16        },
17        "nome" : {
18          "type" : "text",
19          "fields" : {
20            "keyword" : {
21              "type" : "keyword",
22              "ignore_above" : 256
23            }
24          }
25        },
26        "qtd" : {
27          "type" : "long"
28        }
29      }
30    }
31  }
32 }

```

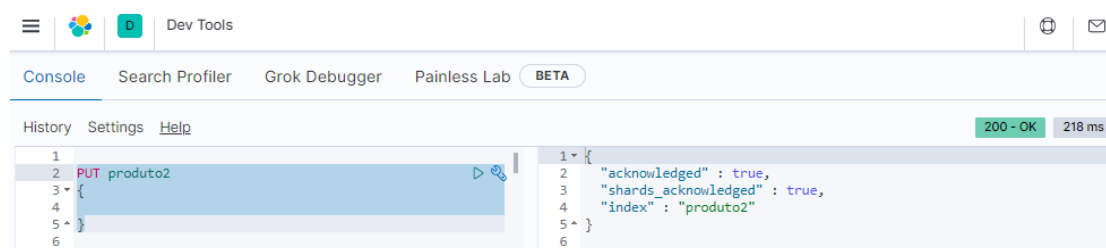
2º Criar o produto2

PUT produto2

```

{
}

```



3º Criar o mapeamento para o produto2 baseado no produto, alterando o que se deseja.

Dica: Faz o mapping do produto1 e minimiza, depois em "proprietes " copia e depois cola dentro do maping do produto2 e altera o campos desejado

PUT produto2/_mapping

```

{
  "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": "short"
    }
}
}

```

The screenshot shows a REST client interface with a PUT request and its response. The request is to the endpoint `produto2/_mapping` with a mapping configuration for a product index. The response is a JSON object indicating that the mapping was successfully acknowledged.

```

1 PUT produto2/_mapping
2 {
3   "properties": {
4     "data": {
5       "type": "date"
6     },
7     "descricao": {
8       "type": "text",
9       "fields": {
10        "keyword": {
11          "type": "keyword",
12          "ignore_above": 256
13        }
14      }
15    },
16    "nome": {
17      "type": "text",
18      "fields": {
19        "keyword": {
20          "type": "keyword",
21          "ignore_above": 256
22        }
23      }
24    },
25    "qtd": {
26      "type": "short"
27    }
28  }
29 }
30
31

```

```

1 {
2   "acknowledged" : true
3 }
4

```

200 - OK 470 ms

- *** Resposta

POST_reindex

```
{
  "source": {
    "index": "produto"},
  "dest": {
    "index": "produto2"}
}
```

The screenshot shows the DevTools Console with the 'Console' tab selected. A POST request to `_reindex` is shown in the 'History' pane. The request body is a JSON object with 'source' and 'dest' indices. The response in the 'Response' pane is a JSON object with various status and timing information, including 'took' (28), 'total' (5), and 'failures' (empty array).

```
1 POST _reindex
2 {
3   "source": {
4     "index": "produto"},
5   "dest": {
6     "index": "produto2"}
7 }
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
```

```
1 {
2   "took" : 28,
3   "timed_out" : false,
4   "total" : 5,
5   "updated" : 0,
6   "created" : 5,
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
```

200 - OK 77 ms

7. Visualizar o mapeamento do índice produto2

** Resposta

GET produto2/_mapping

The screenshot shows the DevTools Console with the 'Console' tab selected. A GET request to `produto2/_mapping` is shown in the 'History' pane. The response in the 'Response' pane is a JSON object showing the mapping for the 'produto2' index, including 'data' (date), 'descricao' (text with keyword field), and 'nome' (text with keyword field).

```
1 GET produto2/_mapping
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
```

```
1 {
2   "produto2" : {
3     "mappings" : {
4       "properties" : {
5         "data" : {
6           "type" : "date"
7         },
8         "descricao" : {
9           "type" : "text",
10          "fields" : {
11            "keyword" : {
12              "type" : "keyword",
13              "ignore_above" : 256
14            }
15          }
16        },
17        "nome" : {
18          "type" : "text",
19          "fields" : {
20            "keyword" : {
21              "type" : "keyword",
22              "ignore_above" : 256
23            }
24          }
25        },
26        "qtd" : {
27          "type" : "short"
28        }
29      }
30    }
31  }
32 }
```

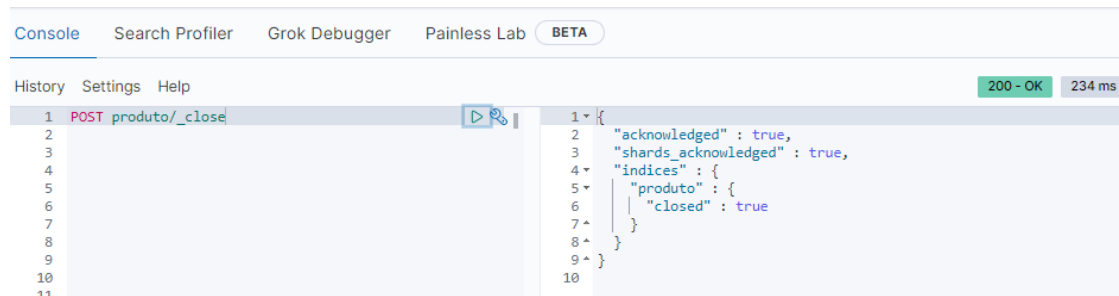
200 - OK 49 ms

8. Fechar o índice produto

**** Resposta**

POST produto/_close

Dica: produt*/_close , fecharia todos que tivesse mesmo radical



The screenshot shows a REST client interface with a 'Console' tab. A POST request to 'produto/_close' has been executed successfully, returning a 200 OK status in 234 ms. The response body is a JSON object indicating that the index 'produto' has been closed.

```
1 POST produto/_close
2
3
4
5
6
7
8
9
10
11
```

```
1 {
2   "acknowledged" : true,
3   "shards_acknowledged" : true,
4   "indices" : {
5     "produto" : {
6       "closed" : true
7     }
8   }
9 }
10
```

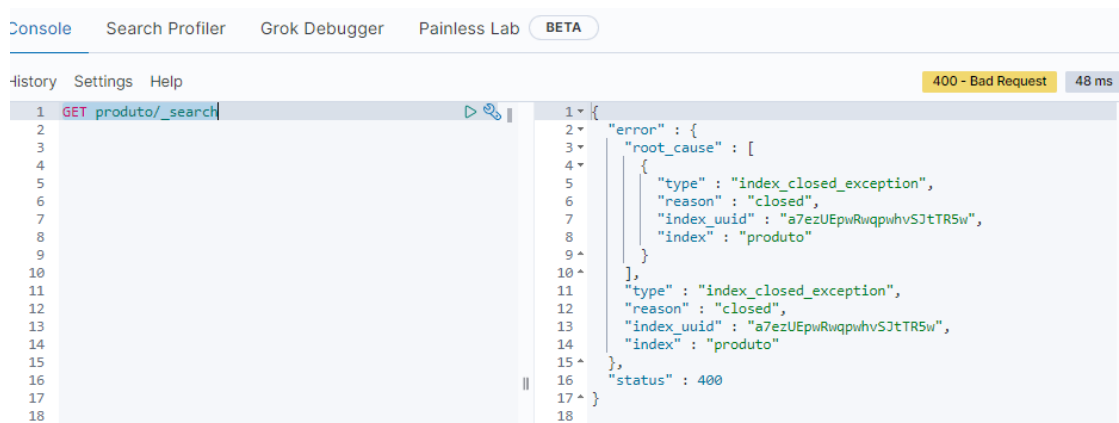
9. Pesquisar todos os documentos no índice produto

**** Resposta**

GET produto/_search

Dica: Não foi possível pois esta fechado, não temos acesso aos dados.

Temos acesso aos metadados



The screenshot shows a REST client interface with a 'Console' tab. A GET request to 'produto/_search' has been executed, resulting in a 400 Bad Request status in 48 ms. The response body is a JSON object indicating that the index 'produto' is closed and cannot be searched.

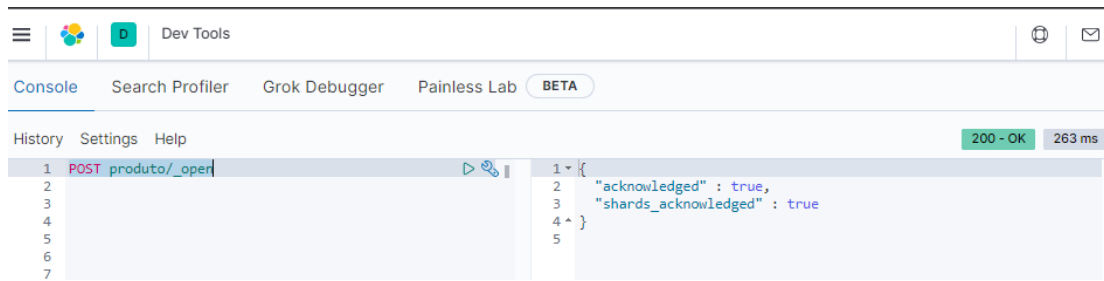
```
1 GET produto/_search
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
```

```
1 {
2   "error" : {
3     "root_cause" : [
4       {
5         "type" : "index_closed_exception",
6         "reason" : "closed",
7         "index_uuid" : "a7ezUEpwRwqpwhvS3tTR5w",
8         "index" : "produto"
9       }
10      ],
11     "type" : "index_closed_exception",
12     "reason" : "closed",
13     "index_uuid" : "a7ezUEpwRwqpwhvS3tTR5w",
14     "index" : "produto"
15   },
16   "status" : 400
17 }
18
```

10. Abrir o índice produto

**** Resposta**

POST produto/_open



11. Clicar no botão de Enviar Tarefa, e enviar o texto: ok