

## Mapeamento de tipos

Vimos nos exercícios do capítulo 1 que se utilizamos a API `_search` para buscar o termo *musica* ao invés de *música*, não obtemos resultado algum. Além disso, quando adicionamos documentos, não queremos estar somente limitados a atributos do tipo `string`. Gostaríamos, por exemplo, de poder indicar que determinado atributo deve ser tratado como data ou mesmo como número. Indo um pouco mais adiante, queremos também ter certeza de que os tipos que definimos sejam respeitados, pois não queremos ter valores incompatíveis com os tipos dos atributos.

### Tipos predefinidos

Por padrão, quando criamos documentos no Elasticsearch em índices e tipos que não foram previamente criados, o próprio ElasticSearch analisa o primeiro documento recebido e infere os tipos de dados. Para olharmos o que foi inferido, basta utilizamos a API `_mapping`:

```
GET catalogo/_mapping/pessoas
```

O resultado esperado é:

```
{
  "catalogo" : {
    "mappings" : {
      "pessoas" : {
        "properties" : {
          "cidade" : {
            "type" : "string"
          },
          "estado" : {
            "type" : "string"
          },
          "formação" : {
            "type" : "string"
          },
          "interesses" : {
            "type" : "string"
          },
          "nome" : {
            "type" : "string"
          },
          "país" : {
            "type" : "string"
          }
        }
      }
    }
  }
}
```

**Importante:** Ainda que novos documentos possam ter novos atributos que não foram previamente mapeados, mapeamentos para atributos já existentes não podem ser alterados.

## Inferência de tipos

Vamos adicionar o campo *nascimento* no registro de id 1.

```
PUT /catalogo/pessoas/1
{
  "nome": "Patrick von Steppat 2",
  "interesses": [
    "computação",
    "culinária",
    "cinema"
  ],
  "cidade": "Rio de Janeiro",
  "formação": "Gastronomia",
  "estado": "RJ",
  "país": "Brasil",
  "nascimento": "1984-10-03"
}
```

Se olharmos o tipo e formato inferido para o atributo nascimento, teremos:

```
"nascimento": {
  "type": "date",
  "format": "strict_date_optional_time||epoch_millis"
}
```

Repare o tipo `strict_date_optional_time||epoch_millis` que automaticamente foi inferido para o atributo nascimento.

## Tipos suportados no Elasticsearch

ElasticSearch suporta diferentes tipos de atributos, inclusive alguns que vão te deixar meio surpresos. Já imaginou um tipo IPv4? Por hora, vamos focar apenas nos tipos que possuem valor direto para nosso projeto. Para mais informações, acesse o link <https://www.elastic.co/guide/en/elasticsearch/reference/current/mapping-types.html> (<https://www.elastic.co/guide/en/elasticsearch/reference/current/mapping-types.html>).

Os tipos básico (também chamados de *core*) são:

- Para texto: *string*
- Para números: *long*, *integer*, *short*, *byte*, *double* e *float*.
- Para datas: *date*. O formato padrão é *strict\_date\_optional\_time||epoch\_millis*, que significa data com hora como opcional ou valor epoch em milissegundos.

Note que o formato da data pode ser customizado para garantir consistência no dado armazenado para buscas e outras operações.

Mais detalhes em <https://www.elastic.co/guide/en/elasticsearch/reference/current/date.html> (<https://www.elastic.co/guide/en/elasticsearch/reference/current/date.html>).

## O que aprendemos?

- Como verificar os mappings para nossos documentos.
- Tipos de dados existentes do Elasticsearch.
- Inferência de tipos pelo Elasticsearch.