

Valderi Leithardt, Dr.

IPW (Class eleven notes)

Summary

- Review of previous class *
- DB (Elasticsearch)
- Exercises
- Work 2 and 3

ViewEngine: Handlebars *(Review of previous class)*

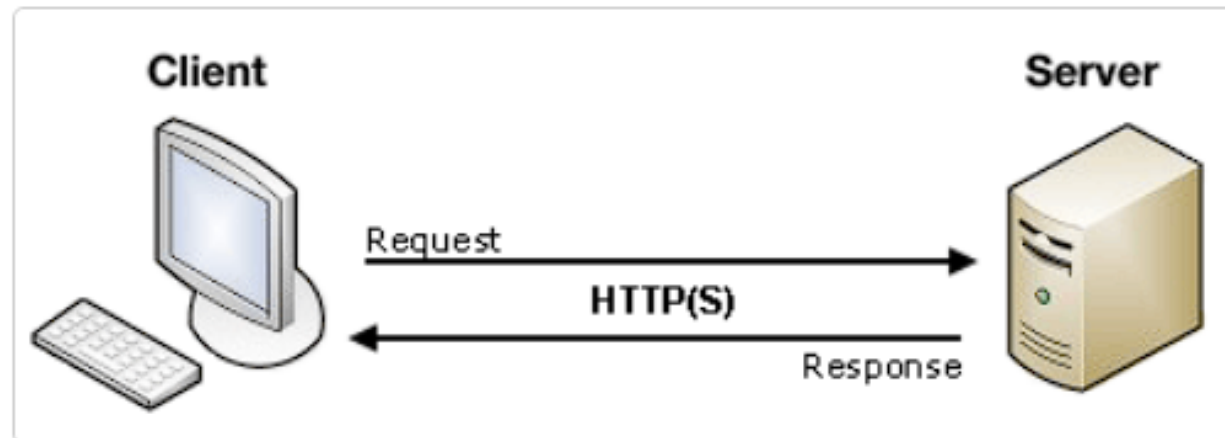
- [Handlebars](#).js é um compilador escrito com JavaScript que pode ser utilizado com HTML e expressão Handlebars, compilando para uma função JavaScript.
- Esta função JavaScript derivada recebe um parâmetro, um objeto - seus dados - e retorna uma string com o HTML e os valores das propriedades do objeto inseridos internamente no HTML.
- Com isso, é gerado um string (HTML) que tem valores de propriedades dos objetos inseridos em lugares relevantes.

- Objeto com Strings

```
var theData = {  
    headerTitle: "Shop Page",  
    weekDay: "Wednesday"  
};  
  
<script id="header" type="text/x-handlebars-template">  
    <div> {{ headerTitle }} </div>  
    Today is {{ weekDay }}  
</script>
```

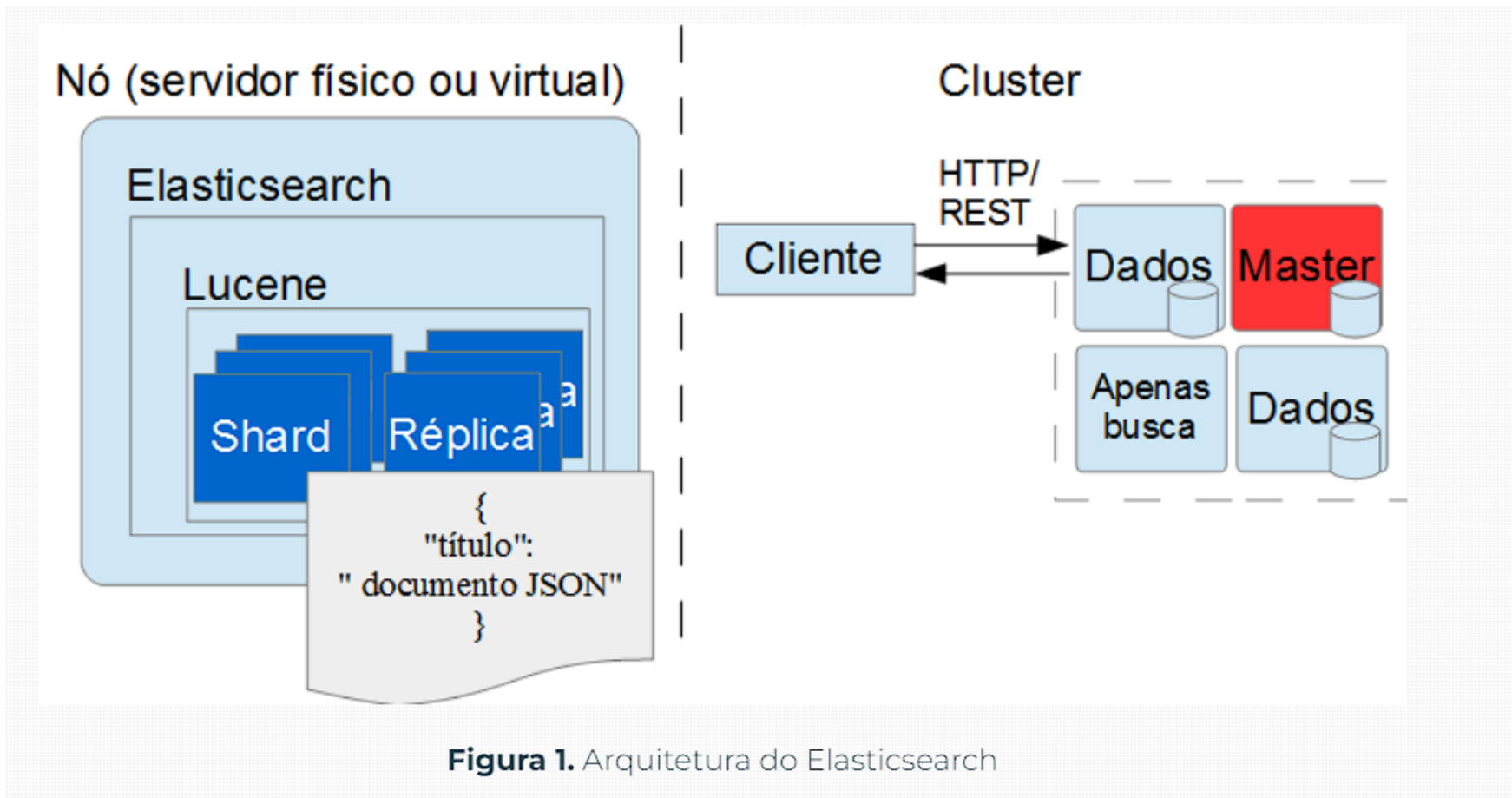
Forms *(Review of previous class)*

- De acordo com ([MDN 2023](#)), Formulários HTML são um dos principais pontos de interação entre um utilizador e um web site ou aplicativo. Eles permitem que utilizadores enviem dados para o web site. Na maior parte do tempo, os dados são enviados para o servidor da web, mas a página da web também pode intercetar para usá-los por conta própria.
- A principal diferença entre um formulário de HTML e um documento regular de HTML é que, maioria das vezes, o dado coletado é enviado ao servidor. Nesse caso, é preciso configurar um servidor web para receber e processar os dados. Como configurar um servidor web, conforme figura disponível em: ([Tutorial MDN](#), 2023).



Elasticsearch

- ✓ Elasticsearch é, de forma geral, uma versão distribuída do Lucene – framework para mineração e tratamento de texto desenvolvido pela [Fundação Apache](#). Isto porque cada nó de um cluster ES contém tal framework para o gerenciamento das informações armazenadas;



Elasticsearch

- De acordo com ([AWS](#), 2023), O Elasticsearch é um mecanismo distribuído de pesquisa e análise desenvolvido com base no Apache Lucene. Desde seu lançamento em 2010, o Elasticsearch se tornou rapidamente o mecanismo de pesquisa mais popular e é comumente usado para casos de uso de análises de log, pesquisa de texto completo, inteligência de segurança, análises de negócios e inteligência operacional.
- Em 21 de janeiro de 2021, a Elastic NV anunciou que mudaria sua estratégia de licenciamento de software e não lançaria novas versões do Elasticsearch e do Kibana sob a licença permissiva Apache License, Version 2.0 (ALv2). Em vez disso, novas versões do software serão oferecidas sob a licença Elastic, com o código-fonte disponível sob a licença Elastic ou a SSPL. Essas licenças não são de código aberto e não oferecem as mesmas liberdades aos utilizadores.
- Para que a comunidade de código aberto continuem a ter um conjunto de análise e investigação segura, há o projeto [OpenSearch](#), uma ramificação do Elasticsearch e Kibana de código aberto, licenciada sob a ALv2 e orientada pela comunidade.

Elasticsearch

- É uma ferramenta de busca e análise de dados. Pode ser um dado textual, numérico, geoespacial, estruturado e não estruturado. * [Tutorial](#) (2023).
- Possui recursos REST APIs simples, com velocidade e escalabilidade que permitem mais agilidade ao desenvolvedor. Dessa forma, é possível indexar vários conteúdos em muitos casos, como:
 - ✓ busca em website;
 - ✓ busca em aplicação;
 - ✓ métricas de infraestrutura;
 - ✓ monitoramento de container;
 - ✓ visualização e análise de dados geoespaciais;
 - ✓ análise de dados empresariais.
- Elasticsearch pode ser usado com várias linguagens, tais como:

.NET (C#); GoLang; Java; **JavaScript**; Perl; PHP; Python; Ruby.

Elasticsearch

➤ Algumas Vantagens

- ✓ É uma ferramenta de busca e análise de dados. Pode ser um dado textual, numérico, busca em website;
- ✓ Rapidez: ele realiza buscas praticamente em tempo real, reduzindo a latência e adaptando-se muito bem em análise de dados de segurança e monitoramento de infraestrutura;
- ✓ Distribuído de fábrica: documentos que estão no Elasticsearch podem ser distribuídos em vários containers. Eles são chamados de shards e podem ser duplicados para oferecer mais segurança nos casos de falhas de hardware.
- ✓ Conjunto de recursos: o Elasticsearch contém uma série de recursos que oferecem mais velocidade, escalabilidade, resiliência, ampla capacidade de armazenamento e uma busca de dados mais eficiente.
- ✓ Gratuidade: os recursos são gratuitos e abertos sob a [licença da Elastic](#).

Elasticsearch

➤ Schema-free – Mapping

- ✓ Mapping é o “schema” de uma tabela, onde definimos os campos, tipos e regras aplicadas; ([Tutorial básico](#) (2023)).
- ✓ Elasticsearch não tem um tipo “list” ou “array”, mas todos os campos podem conter mais de um valor por definição, desde que tenham o mesmo tipo; (2023, [Guide elasticsearch Arrays](#)).
- ✓ Mapeamento é o processo de definir como um documento e os campos que ele contém são armazenados e indexados, cada documento é uma coleção de campos, cada um com seu próprio [tipo de dados](#);
- ✓ Uma definição de mapeamento também inclui campos de metadados, como o *_source field*, que personalizam a forma como os [metadados](#) associados a um documento são tratados;
- ✓ Utilize [mapeamento dinâmico](#) e [mapeamento explícito](#) para definir seus dados. Cada método oferece benefícios diferentes com base em onde você está na jornada de dados.

Elasticsearch

- A estrutura de dados é semelhante a um “schema” dentro de uma base relacional, onde, por exemplo, para um campo que receberá um datatype com valores do tipo “texto”, em uma base dados relacional poderia ser o “char”, já para Elasticsearch, temos o “text” ou “keyword”.

Exemplo documento no Elasticsearch:

```
PUT exemplo1/_doc/0
{
  "nome" : "Aluno A",
  "endereco" : "Rua que sobe e desce e numero não aparece, 1000",
  "idade" : 90
}
```

Elasticsearch

schema-free

- Elasticsearch indexará sem problemas o documento sem previamente definirmos uma estrutura de dados (SCHEMA-FREE), mas será atribuído com base nos dados informados e valores para a estrutura por padrão, como “keyword”, “text” e “long”.

Estrutura de dados

Exemplo:

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

Elasticsearch

schema-free

Documento

```
PUT exemplonewestrustructure/_doc/1
{
  "nome" : "Aluno ",
  "endereco" : "Rua que sobe e
desce e o numero nao aparece,
1000",
  "idade" : 90
}
```

*** Verificar se Estrutura esta de acordo**

```
GET meuteste/_mapping
```

Nova Estrutura de dados

```
PUT exemplonewestrustructure
{
  "mappings": {
    "properties": {
      "nome" : {
        "type" : "text"
      },
      "endereco" : {
        "type" : "keyword"
      },
      "idade" : {
        "type" : "integer"
      }
    }
  }
}
```

Elasticsearch

schema-free

Comando para verificação de índice

GET exemplonewestrustructure/_settings

Resultado

```
{
  "exemplonewestrustructure" : {
    "mappings" : {
      "properties" : {
        "endereco" : {
          "type" : "keyword"
        },
        "idade" : {
          "type" : "integer"
        },
        "nome" : {
          "type" : "text"
        }
      }
    }
  }
}
```

Elasticsearch - RESTful

- ✓ Elasticsearch [RESTful também é centrado em API](#), tornando-o mais utilizável. Suas operações podem ser facilmente acessadas por HTTP por meio da API RestFul para que possam ser integradas perfeitamente em qualquer aplicativo;
- ✓ Vários wrappers estão disponíveis em diversas linguagens de programação, eliminando a necessidade de usar a API manualmente, a maioria das operações pode ser acessado por meio de chamadas de função de biblioteca que trata a comunicação com o próprio mecanismo;
- ✓ Com uso de [operações CRUD](#) - C reate, R ead, U pdate, D elete - é possível operar efetivamente sobre os dados presentes no armazenamento persistente. São semelhantes ao CRUD obtido por base de dados relacionais sendo realizados por meio da interface HTTP presente nas APIs RESTful;
- ✓ Elasticsearch is a [distributed, RESTful](#) search engine optimized for speed and relevance on production-scale workloads;
- ✓ [Elasticsearch Rest API](#);
- ✓ [Elasticsearch REST API reference](#).

Exercises

- Exercício capítulo 15 do livro da disciplina disponível em:
https://eloquentjavascript.net/15_event.html
- 2º Trabalho disponibilizado em:
<https://github.com/isel-leic-ipw/2324i-IPW-LEIC31D/wiki>

**** Concluir exercícios da aulas anteriores.**

References

- Acessado em 2023 <https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html>
- Acesso em 2023 <https://www.elastic.co/guide/en/elasticsearch/reference/current/mapping.html>
- Também foram realizadas adaptações e modificações com base no material disponibilizado por Professor Luís Falcão, acesso online em: <https://github.com/isel-leic-ipw/>
- Aulas gravadas Professor Falcão:
- Aula 14 (30/10): https://videoconf-colibri.zoom.us/rec/share/uJqbBQ6gKIQOmm3eB_l0h64uX5wh97jwPVU3cGe5lhunB_uCaljuuijBasP0p9FW_1m.0uErW8kPXolpwm7q
- Aula 14 (03/11): https://videoconf-colibri.zoom.us/rec/share/ut946W9DizlhxybhBuFzKTNyH581Pugb_0_dJP-fK0c5dRamq_XB3ocQ_XlSkFeJW.RJ1E_XJfBMHk02Ziw
- Aula 15 (06/11): https://videoconf-colibri.zoom.us/rec/share/pD0bViUhr6cNutXipMID3DZjDPwdJevXq_IFMv1uN1KVo5jQtIPGYILIQY_EetmU5O.r1GBu4nB_jZuKiWk
- Aula 16 (08/11): https://videoconf-colibri.zoom.us/rec/share/M8sjBAme3bwnrXuEMFafdnDFS0kCpwVLZEU-cdGLX8SQYsl8-Au3OO_0wIXzCSmA.UpR716AVkQCKQAIL

* Todos os links foram acessado em 2023

Valderi Leithardt, Dr.

Professor IPW

valderi.leithardt@isel.pt