

Roteiro da Parte Prática

Minicurso SBSI 2015

Ernesto Fonseca Veiga
Márcio Vinícius Oliveira Sena
Renato de Freitas Bulcão Neto

Apache Jena

- Apache Jena
 - Tutoriais - <https://jena.apache.org/tutorials/index.html>
 - [RDF](#)
 - [SPARQL](#)
 - API's
 - [RDF](#)
 - [SPARQL](#)
 - [TDB](#)
 - [entre outras.](#)



O Projeto

- **MinicursoSBSI**

- GitHub

- <https://github.com/marciovsena/MinicursoSBSI>

Colabore!



Aplicação 1

Recomendação de Revisores de Artigos Científicos

- **Problema:** encontrar revisores para artigos científicos pode ser um problema se não se conhecer um pesquisador especialista na área de conhecimento de cada artigo
- **Solução:** sabendo-se a(s) área(s) de conhecimento de um artigo científico, deve-se localizar especialista(s) dessa mesma área de conhecimento, ou de subáreas da mesma. Esses revisores são candidatos em potencial para revisar os devidos artigos.
- **Ontologias:** FOAF e ACM CS

Ontologias

- Aplicação 1
 - **Protégé**
 - Navegação na ferramenta
 - Ontologias
 - ACM
 - FOAF (Friend of a friend)



Schemagen

- Schemagen

- O que é?
- Exemplo de uso
 - Baixar o pacote java
 - `java jena.schemagen -i <input> [-o <output file>] --ontology`
 - Obs: Setar \$PATH

Estrutura das aplicações

```
src
├── app1
│   ├── controller
│   │   ├── ModelController
│   │   ├── QueryController
│   │   └── ResourceController
│   ├── database
│   │   └── PersistenceTDB
│   ├── main
│   │   ├── App
│   │   └── README-ME.txt
│   └── ...
├── app2
│   ├── controller
│   │   ├── ModelController
│   │   ├── QueryController
│   │   └── ResourceController
│   ├── database
│   │   └── PersistenceTDB
│   ├── main
│   │   └── App
│   └── ...
├── ontologies
│   ├── ACM
│   │   └── acm.owl
│   ├── FOAF
│   │   └── foaf.owl
│   └── ...
└── tutorials
    ├── RDF
    ├── SPARQL
    └── TDB
```

```
public class ModelController {

    static Model model;
    static String FOAF = "src/ontologies/foaf.owl";
    static String ACM = "src/ontologies/acm.owl";

    public static void inicializaModelo() {...}

    public static void imprimirModeloTURTLE() {...}

    public static InfModel runRDFSReasoner(Model model) {...}

    public static InfModel runOWLMicroReasoner(Model model) {...}

    public static InfModel runPelletReasoner(Model model) throws MalformedURLException {...}

    public static void listarPessoas() {...}

    public static void listarArtigos() {...}

    public static void listarAutores() {...}

    public static void listarRevisores() {...}

    public static void listarArtigosPorArea() {...}

    public static void listarArtigosPorSubarea() {...}

    public static void sugerirRevisoresParaTodosArtigos() {...}

    public static void sugerirRevisoresParaArtigoEspecifico() {...}

    public static void persistirDadosRepositorioTDB() {...}

    public static void deletarDadosRepositorioTDB() {...}

}
```



```
public class QueryController {  
  
    /**...*/  
    public static void getPerson(Model model) {...}  
  
    /**...*/  
    public static void getDocument(Model model) {...}  
  
    /**...*/  
    public static void getActor(Model model) {...}  
  
    /**...*/  
    public static void getReviewersArea(Model model) {...}  
  
    /**...*/  
    public static void getArticleByArea(Model model) {...}  
  
    /**...*/  
    public static void getSubarea(Model model) {...}  
  
    /**...*/  
    public static void suggestReviewersArea(Model model) {...}  
  
    /**...*/  
    public static void suggestReviewersSubarea(Model model) {...}  
  
    /**...*/  
    public static void suggestReviewersForArticleByArea(Model model) {...}  
  
    /**...*/  
    public static void suggestReviewersForArticleBySubarea(Model model) {...}  
  
    /**...*/  
    public static void queryModel(String queryString, Model m) {...}  
  
}
```

```
public class PersistenceTDB {  
    Dataset dataset = null;  
    public PersistenceTDB() {...}  
    /**...*/  
    public void updateModel(Model model) {...}  
    /**...*/  
    public void queryModel(String query) {...}  
    /**...*/  
    public void removeModel(Model model) {...}  
    /**...*/  
    public void closeConection() { dataset.close(); }  
}
```

Aplicação 2

Socialização de pessoas com interesses comuns

- **Problema:** uma pessoa deseja se socializar com outras pessoas que possuam interesse em comum, por exemplo, que têm interesse nas mesmas áreas de conhecimento
- **Solução:** sabendo-se a área de conhecimento na qual a pessoa tem interesse, deve-se localizar outras pessoas que possuam as mesmas áreas de interesse, ou interesse em subáreas dessas. Essas pessoas podem ser socializadas, por exemplo, por meio da criação de grupos, cujo tema/título representa a área de conhecimento em questão