



INFORME PROYECTO WIKI CHILD

1. Descripción

Este proyecto presenta una aplicación web orientada a enseñar a los niños los diferentes tipos de animales mamíferos existentes, de una forma entretenida y amigable.

La información de los animales es extraída dinámicamente de la DBpedia. El proyecto DBpedia aloja datos de la Wikipedia para proponer una versión Web Semántica que puede ser consultada utilizando el lenguaje de consultas SPARQL.

2. Desarrollo

2.1. Conexión DBPedia

Primeramente, se debe establecer la conexión con DBPedia, esta conexión se la realiza con la librería Jersey Client que proporciona un método GET que facilita el intercambio de datos entre clientes, en este caso DBpedia. En la Figura 1 se muestra con creación de un nuevo cliente http que realiza la conexión de DBpedia que responde como salida una consulta sparql en formato json.

```
Client client = new Client();  
url = WS_URL_DBPEDIA + "?default-graph-uri=" + defaultGraph  
    + "&query=" + qry + salida;  
WebResource resource = client.resource(url);  
String result = resource.accept( ...types: "application/sparql-results+json").  
    get(String.class);
```

Figura 1. Conexión DBpedia

2.2. Consultas SPARQL

Las consultas se las realizo utilizando el formato SPARQL, en total fueron tres consultas que se ejecutaron para el proyecto.

La primera consulta que se presenta en la Figura 2 obtiene los primeros 500 nombres y uris de los primeros 500 animales mamíferos, además se establece un filtro que devuelve solo los nombres que tengan idioma español.

```
public ArrayList<Maniferos> consultarAnimalesM() {  
    ArrayList<Maniferos> listaAnimales = new ArrayList<>();  
    String defaultGraph = "http://dbpedia.org";  
    String qry = "SELECT ?animal ?nombre " +  
        "WHERE { " +  
        "?animal rdf:type dbo:Mammal. " +  
        "?animal rdfs:label ?nombre ." +  
        "FILTER (lang(?nombre) = 'es') " +  
        "} LIMIT 500";
```

Figura 2. Consulta SPARQL - animales mamíferos

La segunda consulta se ilustra en la Figura 3 que obtiene la descripción de un animal específico, esta descripción contiene el abstract, la imagen y el



peso correspondiente a cada animal, cabe recalcar que fue necesario utilizar comandos como “**OPCIONAL**” que especifique que cierta información es opcional y **COALESCE** para reemplazar datos que son nulos.

```
public ManiferoCompleto consultarDescripcionAnimal(String tipoAnimal) {
    System.out.println("ID CONSULTA " + tipoAnimal);
    ManiferoCompleto maniferos = new ManiferoCompleto();
    String defaultGraph = "http://dbpedia.org";
    String qry = "SELECT COALESCE(str(?peso),\"S/N\") AS ?peso COALESCE(str(?imagen),\"S/N\") AS " +
        "?imagen str(?descripcion) AS ?descripcion " +
        "WHERE { " +
        "OPTIONAL { <http://dbpedia.org/resource/\"+tipoAnimal+> \"dbo:abstract ?descripcion } ." +
        "OPTIONAL { <http://dbpedia.org/resource/\"+tipoAnimal+> \"dbo:thumbnail ?imagen } ." +
        "OPTIONAL { <http://dbpedia.org/resource/\"+tipoAnimal+> \"dbp:fat ?peso } ." +
        "FILTER (lang(?descripcion) = 'es') " +
        "}";
}
```

Figura 3. Consulta SPARQL - animales mamíferos

La última consulta se observa en la Figura 4, esta consulta devuelve los primeros 50 nombres y uris de animales mamíferos que serán utilizados para crear un método aleatorio que obtenga solo tres animales mamíferos que servirán para el juego.

```
public ArrayList<Maniferos> AnimalAleatorio() {
    ArrayList<Maniferos> listaAnimales = new ArrayList<>();
    ArrayList<Maniferos> listaAnimalesAleatoria = new ArrayList<>();
    String defaultGraph = "http://dbpedia.org";
    String qry = "SELECT str(?nombre) as ?nombre ?imagen " +
        "WHERE { " +
        "?animal rdf:type dbo:Mammal. " +
        "?animal rdfs:label ?nombre ." +
        "?animal dbo:thumbnail ?imagen ." +
        "FILTER (lang(?nombre) = 'es')"+
        "} LIMIT 50";
}
```

Figura 4. Consulta SPARQL - animales mamíferos

2.3. Intercambio de datos Controlador Vista

Para el intercambio de datos se trabajó con la librería **stereotype.Controller** que permite utilizar métodos HTTP como GET y POST para el intercambio de datos entre clientes.

Como se observa en la Figura 5 se realizaron tres métodos GET para enviar los datos de los mamíferos a la vista y proceder a la presentación de los mismos utilizando html, javascript y jquery.



```
@GetMapping("/")
public String homePage(Model model) {
    listaManiferos = consultas.consultarAnimalesM();
    model.addAttribute( attributeName: "listaManiferos", listaManiferos);
    return "home";
}

@GetMapping("/manifero/{id}")
public ResponseEntity<ManiferoCompleto> maniferoDescripcion
    (@PathVariable(value="id") String id)
    throws JsonProcessingException {
    manifero = consultas.consultarDescripcionAnimal(id);
    return new ResponseEntity<ManiferoCompleto>(manifero, HttpStatus.OK);
}

@GetMapping("/juguemos")
public String juguemos(Model model) {
    listaManiferosAleatoria = consultas.AnimalAleatorio();
    model.addAttribute( attributeName: "listaManiferos", listaManiferosAleatoria);
    return "juguemos";
}
```

Figura 5. Intercambio de datos Controlador - Vista

2.4. Presentación de resultados

La presentación de resultados se la realizó con html utilizando **tymeleaf** para capturar los datos y presentarlos en los elementos de la plantilla. Como se observa en la Figura 6 la carga de los animales mamíferos se la realiza en una tabla html.

```
<div class="col-lg-3">
    
    
    <br><br>
    <div style="overflow-y: scroll; height: 600px">
        <table class="table table-bordered table-hover table-striped table-hidden">
            <thead>
                <tr>
                    <th style="text-align: center">Animales Mamiferos</th>
                </tr>
            </thead>
            <tbody>
                <tr th:each="manifero : ${listaManiferos}">
                    <td><a th:attr="onclick=cargarDescripcion('${manifero.uri}','${manifero.nombre})'" ><th:text="manifero.nombre" /></td>
                </tr>
            </tbody>
        </table>
    </div>
</div>
```

Figura 6. Presentación de animales mamíferos en html.

De igual forma la presentación del juego se la realiza mediante elementos html y jquery para la interacción.



```
<div class="container">
  <div style="text-align: center">
    <img th:attr="src=${listaMamiferos.get(0).uri}" height="250px" width="250px" style="border-radius: 25%;border: 2px solid cornflor" />
  </div>
  <br><br>
  <div class="row">
    <div class="col-lg-4">
      <p style="text-align: center"><button th:attr="onclick=|ValidarResultado('${listaMamiferos.get(0).nombre}',this.innerHTML)|">
    </div>
    <div class="col-lg-4">
      <p style="text-align: center"><button th:attr="onclick=|ValidarResultado('${listaMamiferos.get(0).nombre}',this.innerHTML)|">
    </div>
    <div class="col-lg-4">
      <p style="text-align: center"><button th:attr="onclick=|ValidarResultado('${listaMamiferos.get(0).nombre}',this.innerHTML)|">
    </div>
  </div>
</div>
```

Figura 7. Presentación de animales mamíferos en html.

3. Resultados

La Figura 8 representa la vista principal de usuario. En el bloque izquierdo se observa la lista de animales mamíferos en un scroll que el usuario puede manipular para seleccionar el mamífero. En el bloque céntrico se encuentran la descripción del animal seleccionado y en el bloque derecho se encuentra la información de los colaboradores del proyecto.



WIKI CHILD

URSUS AMERICANUS



Descripción

El oso negro (*Ursus americanus*), también llamado oso negro americano, es una especie de mamífero carnívoro de la familia de los úrsidos. Es el oso más común en Norteamérica. Se encuentra en una superficie geográfica que se extiende desde el norte de Canadá y Alaska hasta la Sierra Gorda de México, y de las costas atlánticas, a las costas pacíficas de Norteamérica. Está presente en un gran número de Estados estadounidenses y en todas las provincias canadienses. Prefiere los bosques y las montañas donde encuentra su comida y puede ocultarse.



JUGUEMOS!

Colaboradores



Ing. Nelson Piedra, PhD
Departamento de Ciencias de la Computación
Universidad Técnica Particular de Loja
nopiedra@utpl.edu.ec



Erika Vásquez
Sistemas Informáticos y Computación
Universidad Técnica Particular de Loja
etvasquez@utpl.edu.ec

Figura 8. Presentación de animales mamíferos en html.



Al dar click en el botón **JUGUEMOS** se abre una ventana que nos muestra un juego, ver Figura 10, en el cual se debe seleccionar el nombre correcto del mamífero presentado.

Si se selecciona una opción incorrecta se muestra un mensaje de error, tal como se observa en la Figura 10, si selecciona correctamente se muestra un mensaje de éxito, ver Figura 11.

¡HORA DE DIVERTIRNOS!

¿Cuál es el nombre nativo del siguiente animal?



Acinonyx jubatus

Vombatidae

Pholidota (animal)

JUGAR DE NUEVO

ATRÁS

Figura 9. Juego interactivo

¡HORA DE DIVERTIRNOS!

¿Cuál es el nombre nativo del siguiente animal?



Acinonyx jubatus

Pholidota (animal)

ATRÁS

Figura 10. Presentación selección errónea



UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA
SISTEMAS BASADOS EN EL CONOCIMIENTO

¡HORA DE DIVERTIRNOS!

¿Cuál es el nombre nativo del siguiente animal?



Figura 11. Presentación selección correcta