

Docente:

Nelson Piedra

Departamento de Ciencias de la Computación

Universidad Técnica Particular de Loja

nopiedra@utpl.edu.ec

Estudiante:

Jairo Valle R.

Carrera de Sistemas

Informáticos y

Computación

Universidad Técnica

Particular de Loja

jvalle1@utpl.edu.ec

Estudiante:

José Luis Díaz.

Carrera de Sistemas

Informáticos y

Computación

Universidad Técnica

Particular de Loja

jldiaz4@utpl.edu.ec

Estudiante:

Margory Ayala V.

Carrera de Sistemas

Informáticos y

Computación

Universidad Técnica

Particular de Loja

meayala@utpl.edu.ec

TRABAJO GRUPAL “SISTEMA RECOMENDADOR DE PIZZAS EN BASE A ONTOLOGÍA OWL Y APACHE JENA”

Disponible en: <https://github.com/Jairstal/SISTEMA-RECOMENDADOR-APACHE-JENA-PROTEGE-OWL.git>

1. INTRODUCCIÓN

En este documento se encuentra todo el proceso que se a seguido para la implementación de un sistema de recomendación basado en una ontología. En el se describe todos los pasos realizados de acuerdo a la investigación acerca de web semántica e implementación de un razonador. El desarrollo de este proyecto se lo realizó de la siguiente forma:

- Creación de una Ontología sobre Pizza (Margory Ayala)
- Implementación del recomendador (Jairo Valle)
- interfaces visualizar y navegar por dicha ontología (Luis Díaz)

Además de lo antes descrito de debe mencionar que la parte de investigación se la realizó conjuntamente todos los miembros del proyecto.

En el documento se describen los avances realizados hasta la fecha los cuales se detallan a continuación:

- Creación de la ontología del dominio pizza.
- Construcción de clases, subclases.
- Propiedad de owl, subPropiedades y dominios y rangos.
- Restricciones
- Creación de pizzas
- Programación e implementación de recomendador
 - Implementación de librerías Jena
 - Definición del recomendador
 - Interacción del recomendador con la ontología
- Interfaz y navegación
- Desarrollo de interfaz de inicio
- Interfaz de búsqueda y resultados
- Obtención de datos para la presentación.

Para la culminación de este proyecto está aún en proceso la consulta a la ontología ya que solo logramos consultar la mitad de la ontología.

2.- JUSTIFICACIÓN

La web semántica nos invita a contribuir en la construcción de una integración entre tecnología, contenidos y pedagogía, es necesario el uso de tecnologías de información que vayan en pro del desarrollo semántico de la información disponible digitalmente.

Este proyecto por ende intenta hacer uso de la ontología web (OWL) sobre el tema “Pizzas” trabajado en la materia de Sistemas basados en el conocimiento, para proveer de un aplicativo de reconocimiento e interpretación de la ontología y que nos ayudará sin duda en la práctica de los aprendizajes recibidos en la materia.

3.- ESTADO DEL ARTE

Web Semántica¹

La web semántica [Berners-Lee 2001] propone superar las limitaciones de la web actual mediante la introducción de descripciones explícitas del significado, la estructura interna y la estructura global de los contenidos y servicios disponibles en la WWW. La Web Semántica como infraestructura basada en metadatos aporta un camino para razonar en la Web, extendiendo así sus capacidades. Se trata de aprovechar las capacidades que pueda tener un computador para resolver problemas específicos, a través de operaciones previamente establecidas que se llevarán a cabo sobre datos existentes pero muy bien definidos.

Características web Semántica:

- Estructura mejor la información con el fin de poder localizar y encontrarla de forma más sencilla.
- Se basa en metadatos que permiten razonar en la web.
- Utiliza un lenguaje universal que permite el intercambio con otros.
- Posee herramientas capaces de procesar la información de manera sencilla.
- Es estándar y permite que los contenidos puedan ser utilizados y entendidos por cualquier software.

- Los navegadores distinguen más que páginas HTML, con lenguajes como XML Y RDF.
- Posee orden y flexibilidad

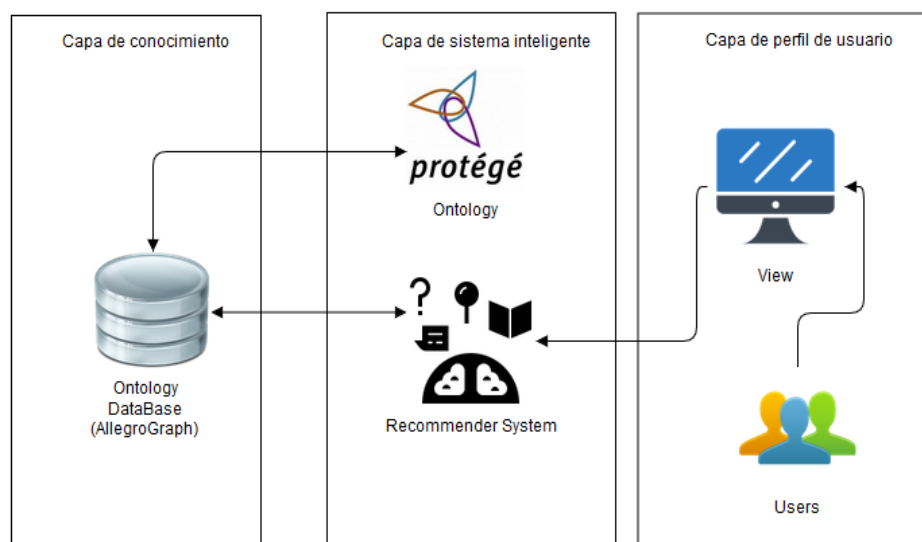
Tecnología a utilizar

Protégé: Plataforma gratuita y de código abierto que permite construir sistemas inteligentes basados en conocimiento con ontologías. Sus características son: Cuentan con el apoyo de una comunidad de usuarios y desarrolladores que permiten dar soporte a inquietudes de otros usuarios, es compatible con estándares W3C, código abierto.²

Java: Es un lenguaje de Programación de propósito general basado en clases, que presenta varias ventajas al estar definido como un lenguaje pensado para que el código escrito y compilado sea independiente del sistema operativo y del dispositivo en el que se ejecute. Se utilizó este lenguaje ya que contiene las librerías necesarias para la manipulación de los estándares utilizados en la Web Semántica y además proporciona amplias facilidades a la hora de la creación y publicación de servicios en la Web³.

Apache Jena: Es un marco de desarrollo open source escrito en Java utilizado para la construcción de aplicaciones de Web Semántica y Linked Data. Está compuesta por diferentes APIs que interactúan entre sí para procesar los datos RDF. Brinda soporte para los estándares publicados por las recomendaciones de la W3C. Jena integra un motor de inferencia basado en reglas y ontologías para realizar razonamiento semántico usando OWL o RDFS, además de varios mecanismos para el almacenamiento de tripletas RDF ya sea en memoria o en disco⁴.

4.- ARQUITECTURA Y COMPONENTES



Capa de perfil de usuario: En esta capa es la que permite la interacción del usuario con el sistema, mediante una aplicación web. Esta aplicación permite al usuario obtener recomendaciones de pizza.

Capa de sistema inteligente. - En esta capa es donde se crea el módulo de ontologías, así como también se crea el módulo del sistema recomendador tomando en cuenta los mismo factores que el módulo de ontologías.

Módulo de ontologías:

La principal razón para utilizar ontologías es el enriquecimiento de información que puede lograrse, en ella Se encuentra toda la información necesaria para el conocimiento que proporciona la aplicación web.

Para la creación de la ontología se hizo uso de la herramienta Protégé anteriormente explicada. Dicha ontología nos permitirá generar el conocimiento que se obtendrá posteriormente a través de consultas e inferencias.

El sistema de recomendador usa el razonamiento generado a través de las inferencias que se realizan en el conocimiento almacenado en nuestra base de datos para determinar las recomendaciones que se debe presentar al usuario.

Módulo del sistema recomendador:

En este módulo se define un algoritmo en el que se indican los pasos necesarios para analizar y presentar la recomendación, tomando en cuenta que el sistema recomendador se basa en los contenidos del objeto de aprendizaje y en el conocimiento generado por la red ontológica.

El recomendador nos va a crear dos modelos de comparación para poder definir si estos modelos se relacionan, al momento de ser así nos devuelve una variable booleana con valor true, que nos permitirá devolver la consulta.

El razonador que se va a utilizar es el Jena este Apache Jena este nos va a permitir la manipulación de las ontologías o de los recursos RDF, RDFS o OWL, realiza las consultas necesarias a las ontologías e interactúa directamente con los módulos de perfiles de usuario, con el módulo recomendador.

Incluye un motor de reglas de inferencia añadiendo reglas y creando modelos nuevos mediante intersecciones, uniones y diferencia entre distintas clases de una ontología.

El razonador que usamos es OWLReasoner, que es uno de los que están incluidos en Jena.

Los razonadores Jena OWL podrían describirse como razonadores basados en instancias. Es decir, funcionan mediante el uso de reglas para propagar las implicaciones if-and only-if- de las construcciones OWL en los datos de la instancia.

La aplicación acceden a la maquinaria de inferencia mediante el uso [ModelFactory](#) que asocia un conjunto de datos con el razonador para crear un nuevo modelo.

Las consultas al modelo creado devolverán no solo las declaraciones que estaban presentes en los datos originales, sino también declaraciones adicionales que pueden derivarse de los datos utilizando las reglas u otros mecanismos de inferencia implementados por el razonador.

Después de haber creado un modelo de inferencia, cualquier operación de API que acceda a declaraciones RDF podrá acceder a declaraciones adicionales que se derivan de los datos vinculados por medio del razonador.

```

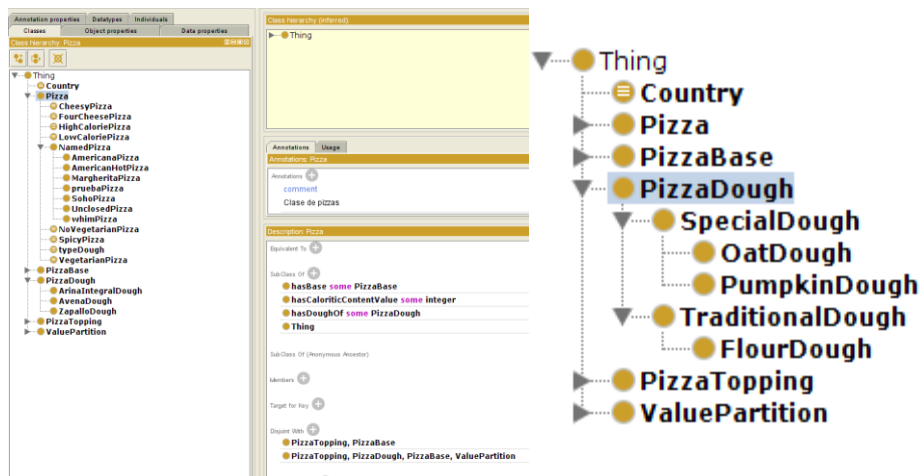
    OntModel mode = null;
mode = ModelFactory.createOntologyModel( OntModelSpec.OWL_MEM_RULE_INF );

```

Capa de conocimiento: Esta capa permite almacenar toda la información para generar conocimiento a través de tripletas RDF, las cuales vienen dadas por un conjunto de tres entidades que son: sujeto, predicado y objeto.

5.- DESARROLLO

Comenzamos definiendo bien nuestra ontología de pizza **en protégé**, en ella especificamos que todas las pizza deben constar de un tipo de masa, el cual nos permitirá realizar la recomendación en base al tipo de masa de su preferencia.



En la siguiente figura se puede observar que toda piza debe llevar por lo menos un tipo de masa

Description: WhimPizza	
Equivalent To	+
SubClass Of	+
● hasDoughOf only OatDough	
● hasTopping some OnionTopping	
● hasTopping some ParmezanTopping	
● hasTopping some PepperoniTopping	
● hasTopping some TomatoTopping	
● NamedPizza	
SubClass Of (Anonymous Ancestor)	
● hasBase some PizzaBase	
● hasCaloriticContentValue some integer	
● hasDoughOf some PizzaDough	

Con ello se puede clasificar en las pizzas que llevan un tipo específico de masa como masa tradicional como especial.

Ya definida nuestra ontología podemos comenzar a programar, para ello utilizaremos Java como lenguaje de programación y Jena para implementar la tecnología de la web semántica.

Debemos tener en cuenta agregar las librerías de Jena a nuestro proyecto.

```
import com.hp.hpl.jena.ontology.OntModel;
import com.hp.hpl.jena.ontology.OntModelSpec;
import com.hp.hpl.jena.query.QueryExecution;
import com.hp.hpl.jena.query.QueryExecutionFactory;
import com.hp.hpl.jena.query.QueryFactory;
import com.hp.hpl.jena.query.ResultSetFormatter;
import com.hp.hpl.jena.rdf.model.ModelFactory;
import com.hp.hpl.jena.util.FileManager;
import com.hp.hpl.jena.util.iterator.ExtendedIterator;
```

Ahora bien, comenzaremos a cargar la ontología con Jena. Cargamos el modelo y empezamos a trabajar con el mediante el api de Jena.

OWL_MEM_RULE_INF es uno de los modelos definidos por Jena para el acceso a elementos de la ontología. Razonador basado en reglas

```

class OpenOWL {
    static String s;

    //Conexion
    static OntModel OpenConnectOWL(){

        OntModel mode = null;
        // crea modelo utilizando como razonador OWL_MEM_RULE_INF
        mode = ModelFactory.createOntologyModel( OntModelSpec.OWL_MEM_RULE_INF );
        //Abrimos el archivo con la ontologia
        java.io.InputStream in = FileManager.get().open( "C:/CreacionPizza.owl" );
        if (in == null) {
            throw new IllegalArgumentException("no encontrado");
        }
        //leemos archivo
        return (OntModel) mode.read(in, "");
    }

    // conectar nuestro archivo owl con jena y retornar resultado
    static com.hp.hpl.jena.query.ResultSet ExecSparql(String Query){

        com.hp.hpl.jena.query.Query query = QueryFactory.create(Query);
        QueryExecution qe = QueryExecutionFactory.create(query, OpenConnectOWL());
        com.hp.hpl.jena.query.ResultSet results = qe.execSelect();

        return results;
    }
}

```

Establecemos el uso de un razonador incorporado a Jena por defecto mediante un método de obtención de resultados de la ontología.

```

// Connecté au OWL File et retourner le String
static String GetResultAsString(String Query){
    try {
        com.hp.hpl.jena.query.Query query = QueryFactory.create(Query);
        QueryExecution qe = QueryExecutionFactory.create(query, OpenConnectOWL());
        com.hp.hpl.jena.query.ResultSet results = qe.execSelect();
        //Razonador
        Reasoner reasoner = ReasonerRegistry.getRDFSReasoner();
        reasoner.bindSchema(mode);
        if(results.hasNext()){
            ByteArrayOutputStream go = new ByteArrayOutputStream ();
            ResultSetFormatter.out( (OutputStream)go ,results, query);
            // String s = go.toString();
        }
    }
}

```

Ahora incorporamos con nuestra interfaz



Al realizar la consulta de Toppings que conforman una pizza americana se realiza la consulta

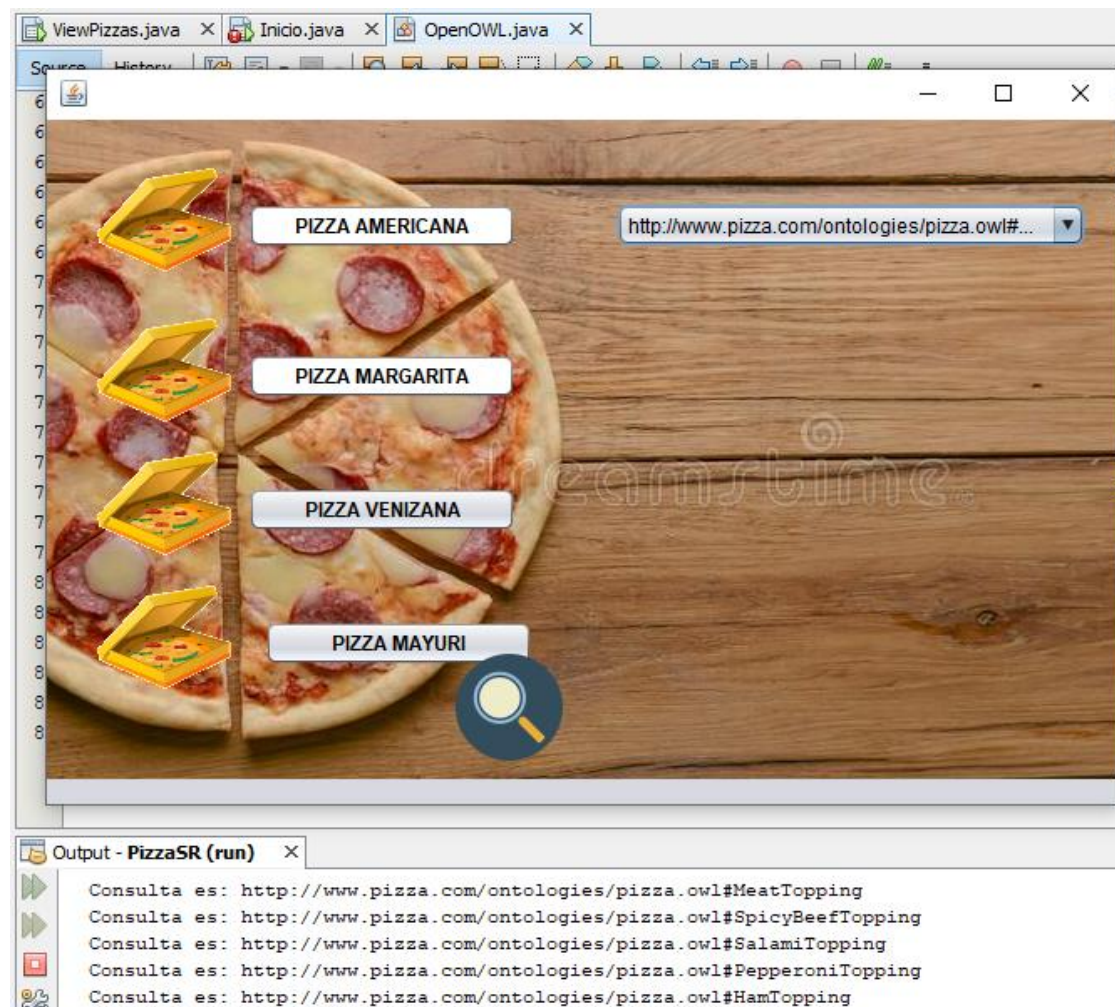
```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String nombre = "AmericanaPizza";
    objClase.setNombrePizza(nombre);

    try {
        // OntModel model = OpenOWL.OpenConnectOWL();
        String variable = objClase.getNombrePizza();
        String queryString;
        queryString = "PREFIX dp:<http://www.pizza.com/ontologies/pizza.owl#> "
            + "PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#> "
            + "PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>"
            + "SELECT (str(?x) as ?Consulta) "
            + "where { ?x rdf:resource dp:hasTopping."
            + " }";

        com.hp.hpl.jena.query.ResultSet results = OpenOWL.ExecSparql(queryString);

        while (results.hasNext()) {
```

Y el resultado obtenido es una lista de los Toppings existentes en la ontología en donde hace falta una codificación más acertada del razonador para que la consulta propuesta nos de como resultado los Toppings de pizza americana



El trabajo realizado está adaptado al tipo de ontología elaborado en el transcurso del curso y con características extras elaboradas por nuestra compañera Margory Ayala, por ende, la investigación de sistemas recomendadores y en si, el uso del razonador por defecto de la librería Jena necesita trabajo en cuanto a su implementación y correcto funcionamiento para que se adapte al tipo de proyecto realizado.

6.- BIBLIOGRAFÍA

- CARVAJAL TABASCO, S. (2017). *TFC-WEB SEMANTICA MEMORIA FINAL* [Ebook]. Retrieved from <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/11021/8/scarvajalTFC0112.pdf>
- Research, S. (2020). protégé. Retrieved 6 August 2020, from <https://protege.stanford.edu/>
- Perry, S. (2012). Conceptos básicos del lenguaje Java. Retrieved 6 August 2020, from <https://developer.ibm.com/es/tutorials/j-introjava1/>
- Avila, J. and Riofrío, X., 2014. *Sistema De Recomendación De Contenido Para TV Digital Basado En Ontologías*. [ebook] Cuenca. Available at: <http://dspace.ucuenca.edu.ec/bitstream/123456789/19875/1/tesis.pdf>
- Apache Jena - Reasoners and rule engines: Jena inference support. (2020). Retrieved 30 July 2020, from <https://jena.apache.org/documentation/inference/#owl>

**INFORME PROYECTO INDIVIDUAL: “APLICACIÓN WEB
DIDÁCTICA USANDO SPARQL, PARA RECUPERAR INFORMACIÓN
EN DBPEDIA”**