|  |  |
| --- | --- |
| ANÁLISIS DE LOS LENGUAJES DE REPRESENTACIÓN | Tecnológico Nacional de México  Instituto Tecnológico de Morelia  “José María Morelos Pavón”  Ingeniería del conocimiento  Unidad III  Profesor: José Omar Hernández Esquivel  Equipo:  **Reyes Rosales Yoi Andrés**  **Gonzalez Rodriguez Maricarmen Guadalupe**  **Muñoz García Rosario**  **Ramos Vega Carlos Manuel**  **Escutia Izquierdo Oscar** |

Contenido

[Introducción 2](#_Toc39259610)

[Actividad 2. ANÁLISIS DE LOS LENGUAJES DE REPRESENTACIÓN 3](#_Toc39259611)

[1. XML (Extensive Markup Language) 3](#_Toc39259612)

[2. SHOE 4](#_Toc39259613)

[3. OIL (Ontology Inference Layer) 6](#_Toc39259614)

[4.OWL (Web Ontology Layer) 8](#_Toc39259615)

[5. KIF 10](#_Toc39259616)

[6. FOAF 12](#_Toc39259617)

[7. RDF 14](#_Toc39259618)

[Herramientas de desarrollo de ontologías 17](#_Toc39259619)

[1. PROTÉGÉ 17](#_Toc39259620)

[2. SWOOP 18](#_Toc39259621)

[3. KAON 18](#_Toc39259622)

[4. JENA 19](#_Toc39259623)

[Proyecto y Herramienta seleccionada 21](#_Toc39259624)

[Criterios de Evaluación de las Ontologías. 22](#_Toc39259625)

[Conclusión 24](#_Toc39259626)

[Referencias 25](#_Toc39259627)

# Introducción

Los lenguajes ontológicos son recursos para expresar ontologías de forma comprensible por las máquinas. Algunos de ellos han surgido con la idea de la **Web Semántica**, de forma que están orientados para tal tecnología. Para tener un panorama general del impacto y uso que se le da actualmente a la ingeniería ontológica cabe mencionar que tiene un uso exhaustivo en la ingeniería del conocimiento, la inteligencia artificial y en las ciencias de la computación.

Al mismo tiempo, surge la necesidad de evaluar lo realizado. La mayoría de las metodologías empleadas para la creación de las ontologías también incluyen un apartado dedicado a la evaluación del resultado sin embargo se han tratado en el presente ensayo de manera muy concreta y general.

Se presentan siete lenguajes y cuatro herramientas ontológicos. Para cada uno de ellos se introducirá su origen y objetivos. Además de la introducción de un proyecto personal de uno de los integrantes del equipo y como una de las herramientas elegidas podría ser de gran ayuda para su actual desarrollo.

# Actividad 2. ANÁLISIS DE LOS LENGUAJES DE REPRESENTACIÓN

Lenguajes de representación

## 1. XML (Extensive Markup Language)

**XML** o **Extensive Markup Language**, es una especificación de **W3C** (World Wide Web Consortium) como lenguaje de marcado de propósito general. Esto quiere decir que, a diferencia de otros lenguajes de marcado, XML no redefine sus etiquetas, por lo que debes definir unas propias. El propósito principal del lenguaje es compartir datos a través de diferentes sistemas, como Internet.

**Origen**

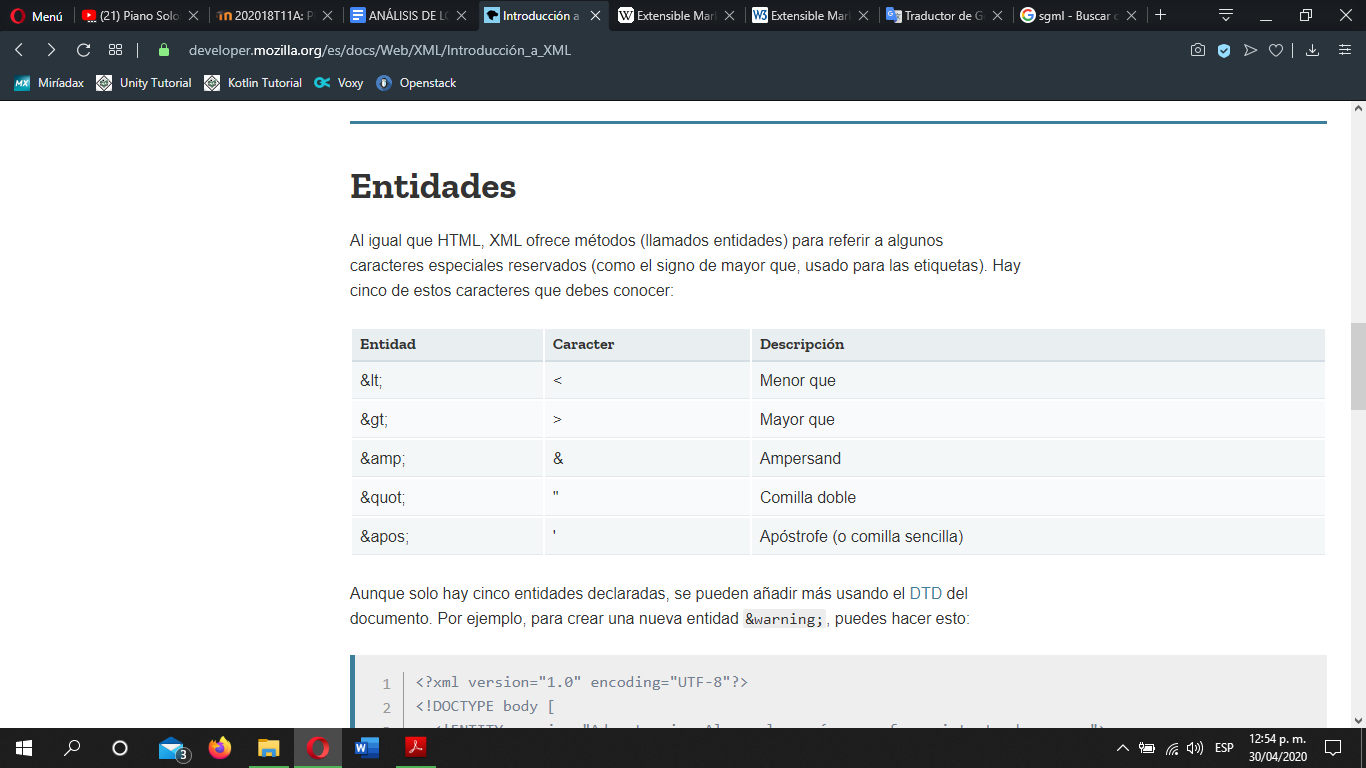
Es un formato de texto muy flexible derivado de **SGML** (Standard Generalized Markup Language) que es una ISO (8879) la cual fue originada por el lenguaje **GML** (Generalized Markup Language) inventado por IBM en los años de 1970. Hay muchos lenguajes basados en XML; Algunos ejemplos son XHTML, MathML, SVG, XUL, XBL, RSS, y RDF. Incluso, si lo deseas, puedes crear uno propio.

**Características**

XML fue diseñado para almacenar y transportar datos y para ser autodescriptivo. La mayoría de las aplicaciones basadas en XML, se espera que sigan funcionando con normalidad a pesar de agregar nuevas etiquetas o removerlas en una nueva versión de etiquetado.

Para ser válido, un documento XML necesita cumplir ciertas reglas de semántica que son generalmente definidas en un esquema XML o en una **Definición de Tipo de Documento (DTD)**. Un documento que contiene una etiqueta no definida es inválido. Por ejemplo, si nunca definimos la etiqueta <warning>, el documento anterior no sería válido.

**Entidades**. Al igual que HTML, XML ofrece **métodos** (llamados entidades) para referir a algunos caracteres especiales reservados (como el signo de mayor que, usado para las etiquetas). Las 5 entidades declaradas son:



Aunque sólo hay cinco entidades declaradas, se pueden añadir más usando el DTD del documento.

XML es generalmente para propósitos descriptivos, pero hay maneras de presentar datos XML. Si no defines una manera específica de renderizar XML, el contenido puro de XML es mostrado en el navegador.

Una forma de estilizar la salida de XML es especificando CSS (Cascading Style Sheets). También hay otra manera más poderosa de presentar XML: Extensible Stylesheet Language Transformations (XSLT) que puede ser usado para transformar XML a otros lenguajes como HTML. Esto hace a XML increíblemente versátil.

## 2. SHOE

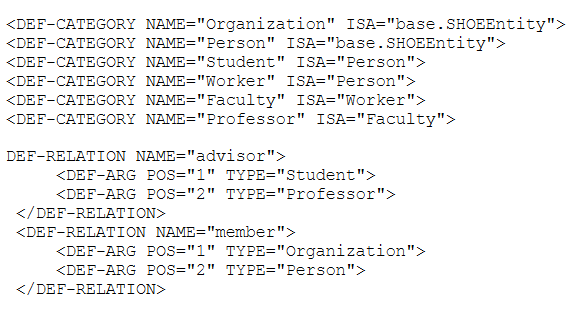
El Simple HTML Ontology Extension (SHOE) fue desarrollado en la Universidad de Maryland como una extensión del HTML que incluye conocimiento semántico en documentos web. La última versión de este lenguaje adapta su sintaxis a XML. Se distingue entre otros por ser el primer lenguaje de etiquetado para ontologías web. Este lenguaje añade etiquetas necesarias para la representación semántica de datos en las páginas web. Así pues con la ayuda de este nace lo que se conoce como la web semántica, actualmente la web semántica se define como las actividades desarrolladas por World Wide Web (www) para la creación de tecnologías de publicación de datos que sean más legibles y sencillas de entender para las aplicaciones informáticas, es decir que para un usuario este tipo de concepto facilitara sus búsquedas de una manera más ordenada ya que el futuro de la web es tener más contenido, más semántica, de forma que la búsqueda de la información deseada se vuelva mucho más sencilla e intuitiva.

El lenguaje permite definir las clases e inferencias necesarias para un mejor entendimiento.

De acuerdo con el modelo ontológico SHOE usa las siguiente terminología y clasificación:

* **Ontología**: Hace referencia a las clasificaciones válidas para las instancias y relaciones entre ellas y elementos. Las Ontologías de este lenguaje son un conjunto de reglas que definen qué tipos de declaraciones pueden hacer los documentos de SHOE y que significa cada uno de la sus declaraciones. **Como ejemplo podemos tomar en cuenta que una ontología de SHOE define como entidad de datos a una “PERSONA” esa entidad tendrá atributos como por ejemplo “nombre” o “edad”.**
* **Categoría**: Refieren a etiquetas para *anotar documentos web* que se suscriban a una o más ontologías, declarar entidades y crear declaraciones sobre esas entidades bajo las reglas establecidas por las ontologías. **Esto también puede entenderse como manejo de clases, las categorías pueden tener relaciones con otras en las que se puede llegar a establecer una herencia si una instancia se clasifica en una categoría, entonces claves clasificadas en esta categoría pueden rellenar posiciones en relaciones definidas por esa categoría o alguna de sus categorías padres.**
* Instancia: También se puede conocer como el objeto o entidad que puede ser clasificado y contenido por una categoría e incluido como argumento de alguna relación. Las instancias también pueden tener instancias, se les conoce como substancias. También existe la posibilidad de declarar instancias constantes.
* Relaciones: Son las que definen una relación entre instancias y otras instancias o datos. **Una relación ocurre entre cero o varios elementos llamados argumentos; si una relación se define para un conjunto de argumentos, éste permite que los documentos SHOE declaren esta relación entre instancias de esos argumentos. Los argumentos se ordenan explícitamente.**
* Reglas: Definen clasificaciones o relaciones válidas en la ontología y documentos.

Un ejemplo sencillo de la especificación de SHOE podría ser el siguiente:



Como se aprecia en la imagen anterior se tienen categorías que clasifican el tipo de persona/usuario y que establece relaciones que definen qué tipos de miembros o tutor son cada uno.

**SHOE se centra en ofrecer una base de conocimiento semántico dentro de la web, y tiene varias formas de hacer uso de los datos adquiridos en la misma, está orientado principalmente a facilitar el trabajo a los robots de la web y a los agentes inteligentes de manera que hagan nuestra vida un poco más fácil.**

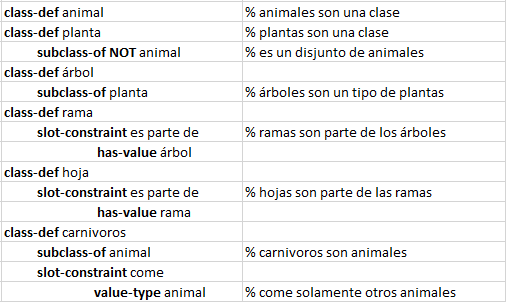
## 3. OIL (Ontology Inference Layer)

Se trata de una representación basada en la Web y las leyes de inferencia para ontologías, combina el uso de primitivas de modelado desde lenguajes basados en frames (clases) con la semántica formal y los servicios de razonamiento provistos por descripciones lógicas (axiomas). Es compatible con RDFS e incluye una semántica precisa para describir el significado de términos e información.

Oil unifica tres importantes aspectos provistos por diferentes comunidades:

* Semántica formal y soporte eficiente para razonamiento
* Enriquecimiento epistemológico de las primitivas
* Propuesta estándar para notación de intercambio

Un ejemplo de la aplicación de esta ontología es la siguiente:



Una ontología expresada en OIL consiste en una lista de definiciones de clases y definiciones de slots, puede verse “slot” como análogo al de atributo existente en el paradigma de orientación a objetos o modelo entidad relación.

Por claridad, primero se introducirá el concepto de class-expression. Una class-expression (expresión de clase) puede ser el nombre de una clase, una slot-constraint, o cualquier combinación compleja de expresiones de clase usando los operadores lógicos AND, OR y NOT. Una definición de clase asocia un nombre de clase con una descripción y consta de los siguientes componentes, (algunos se han omitido por simplicidad):

**El type** de la definición, que puede ser primitivo o definido, por defecto se asume primitivo. Cuando una definición de clase es de tipo primitivo las restricciones que delimitan su definición (slot-constraints) son consideradas necesarias pero no suficientes en el sentido de la pertenencia de un miembro a dicha clase. En caso de tipo definido, las restricciones tienen carácter de necesarias y suficientes. Por ejemplo si se define la clase primitiva elefante como una subclase de animal con una restricción (slot-constraint) estableciendo que su color debe ser gris, entonces todas las instancias de elefante deben ser necesariamente de color gris, pero pueden existir animales grises que no sean instancias de elefante.

**Subclass-of:** Una lista de una o más expresiones de clase (class expressions), de manera que la clase en definición, debe ser una subclase de cada una de las expresiones de la lista.

**Slot-constraint**: Una lista de cero o más (slot-constraints), de manera que la clase en definición, debe ser una subclase de cada una de las clases definidas por los slot-constraints existentes en la lista. Una slot-constraint es una lista de restricciones que se aplican a un slot. Un slot es una relación binaria, esto es que sus instancias son pares de individuos, pero también es implícitamente una definición de clase, cuyas instancias son aquellos individuos que satisfacen la restricción establecida.

Los principales componentes de una slot-constraint son:

• **name** – un string indicando el nombre del slot

**• has-value** – una lista de una o más expresiones de clase, de manera que cada instancia de la clase definida por la slot- constraint debe estar relacionada por la relación slot a una instancia de cada una de las expresiones de clase presentes en la lista.

**• value-type** – una lista de una o más expresiones de clase. Si una instancia de la clase definida por la slot-constraint está relacionada por la relación slot a algún individuo x, entonces x debe ser una instancia de cada expresión de clase de la lista.

El modelado de primitivas y su semántica es uno de los aspectos a tratar en la definición de OIL, respecto de su sintaxis, se atendió a los estándares que actualmente posee la web.

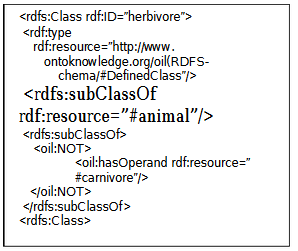
Oil tiene una sintaxis bien definida en XML basada en DTD’s y XML Schema. Por otra parte OIL es una extensión de RDF y RDFS. Con respecto a ontologías, RDFS provee dos importantes contribuciones: una sintaxis estandarizada para denotar ontologías y un conjunto estándar de primitivas de modelado tales como las relaciones instance-of y subclass-of.

Ejemplo de expresión OIL:

En ella se define herbivore como una clase, la cual es una subclase de animal y disjunta de carnivore.

En resumen, OIL tiene algunas ventajas, está apropiadamente asentado sobre lenguajes de la Web, como XML Schemas y RDFS (ver fig. 3.6) y ofrece diferentes niveles de complejidad.

Atendiendo a sus primitivas de modelado, OIL refleja cierto consenso entre áreas como Description Logics y sistemas basados en frames.



## 4.OWL (Web Ontology Layer)

El Lenguaje de Ontologías Web (OWL) supone una recomendación del conocido World Wide Web Consortium (W3C). En palabras del profesor Horrocks, OWL es un "lenguaje de representación del conocimiento descriptivo y basado en lógica". El OWL se deriva de DAML+OIL. Lenguaje de representación del conocimiento descriptivo y basado en lógica.

Está diseñado para usarse cuando la información contenida en los documentos necesita ser procesada por programas o aplicaciones, en oposición a situaciones donde el contenido solamente necesita ser presentado a los seres humanos, puede usarse para representar explícitamente el significado de términos en vocabularios y las relaciones entre aquellos términos.

Owl posee más funcionalidades para expresar el significado y semántica que XML, RDF Y RDFS, además ofrece la posibilidad de representar contenidos de las web interpretable por máquina.

OWL proporciona un lenguaje que utiliza la conexión proporcionada por [RDF](http://www.w3.org/TR/2004/REC-rdf-primer-20040210/) para añadir las siguientes capacidades a las ontologías:

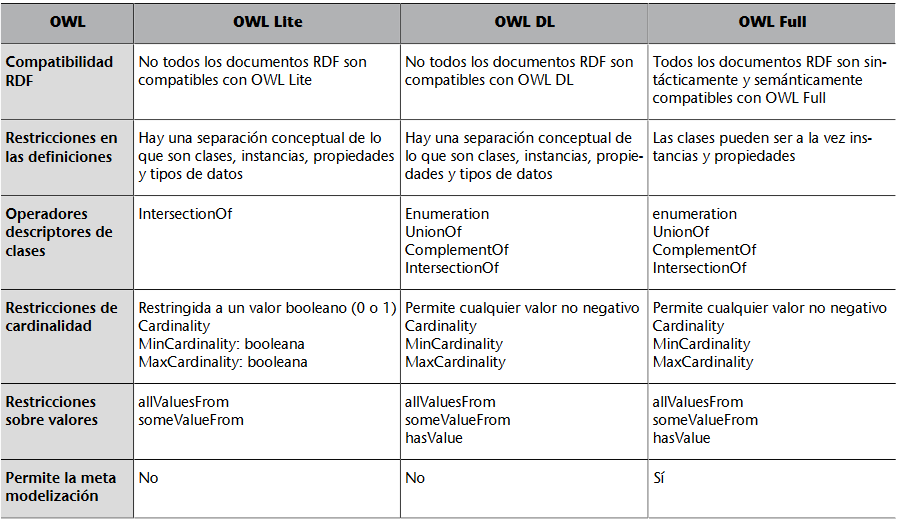
* Capacidad de ser distribuidas a través de varios sistemas
* Escalable a las necesidades de la Web
* Compatible con los estándares Web de accesibilidad e internacionalización
* Abierto y extensible

Permite definir:

* Clases (conceptos)
* Propiedades
* Propiedades de tipos de datos
* Propiedades de objetos
* Individuos (instancias)
* Contiene una serie de predicados predefinidos para la definición de ontologías: clases equivalentes.
* Pueden definirse clases a partir de su descripción:
* Enumeraciones (rojo, verde o azul)
* Restricciones (todos los individuos que tienen más de dos hijos)
* Enunciados lógicos (Persona que no es estudiante y tiene ojos azules)

OWL permite definir ontologías que pueden ser utilizadas a través de diferentes sistemas para compartir información específica de un dominio, como pueden ser finanzas, medicina, deporte, etc. Las ontologías se encargan de definir los conceptos necesarios para describir y representar un determinado dominio y las relaciones existentes entre estos conceptos. En este sentido, mediante OWL se pueden definir y aplicar restricciones sobre clases (conceptos), propiedades de clases y relaciones entre clases.

OWL se presenta en tres niveles de lenguajes: Lite, DL y Full, cada uno con diferentes niveles de expresividad. De este modo, se puede elegir uno u otro según las necesidades de expresividad y eficiencia de nuestra aplicación.



Aunque hay muchos factores que influyen a la hora de escoger el sub-lenguaje OWL más apropiado para una aplicación concreta, mostramos a continuación unas reglas muy simples que pueden ayudar a guiar la elección:

La elección entre OWL Lite y OWL DL se puede basar en el hecho de si la sintaxis OWL Lite es suficiente o no para lo que necesitamos.

La elección entre OWL DL y OWL Full se puede basar en el hecho de si es más importante llevar a cabo razonamiento automático en la ontología, o si por el contrario, conviene tener más capacidad de expresión y facilitar así la modelización del conocimiento (por ejemplo, si hay que definir clases de clases).

## 5. KIF

Es un lenguaje lógico estándar que describe objetos dentro de un sistema computacional. Es el lenguaje utilizado para el intercambio de conocimiento de computadoras. Define un conjunto de objetos, definiciones y relaciones cuyo significado es fijo, por ejemplo, números y funciones aritméticas. Sin embargo los usuarios tienen la libertad de definir significados de cualquier otro símbolo que no esté predefinido.

Este formato para intercambio de conocimiento permite expresar datos simples, por ejemplo, las siguientes oraciones codifican dos tuplas en una base de datos. El primer argumento es el número de seguro social, el segundo es el departamento en el que la persona trabaja y el tercero es el salario de la persona:

**(salario 016-46-3956 compras 8900)**

**(salario 233-23-2342 desarrollo técnico 5000)**

KIF incluye una variedad de operadores lógicos para ayudar a codificar información lógica( como : negaciones, disyunciones, reglas, cuantificadores, entre otros). El siguiente ejemplo es una oración compleja en KIF que representa que cualquier número x elevado a la n es positivo si x es real y n es par

**(<=(>(expt ?x ?n )0 )(número-real ?x)(número-par ?n))**

Algo que distingue a KIF es la habilidad de codificar conocimiento acerca del conocimiento , usando los operadores ? y, además de un vocabulario relacionado. Por ejemplo la siguiente oración afirma que el agente **joe** puede manejar solicitudes de información sobre salario. El uso de comas señala que las variables no se deben tomar literalmente. Sin las comas, esta oración diría que **joe** solo puede manejar la oración **(salario ?x ?y ?z)** en lugar de sus instancias:

**(maneja joe ?(salario, ?x,?y,?z))**

KIF también puede ser utilizado para describir procedimiento, ósea, escribir programas para agentes.

CARACTERÍSTICAS

* Tiene semántica declarativa
* Es completo
* Permite representar metaconocimiento
* Capacidad de ser traducido
* Legibilidad
* Usabilidad

LÓGICA

* Términos funcionales **(+ 2 3**)
* Términos lógicos **(if(>1 2)1 (>2 1) 2 0)**
* Bottom
* True y False
* Equations y inequalities **(= term ) (/= term term)**
* Sentencias relacionadas **(> 1 2**)
* Sentencias lógicas **(=> a b)**
* Sentencias cuantificadas  **forall exists**
* Definiciones  **(defobject s :=t)**

Listas

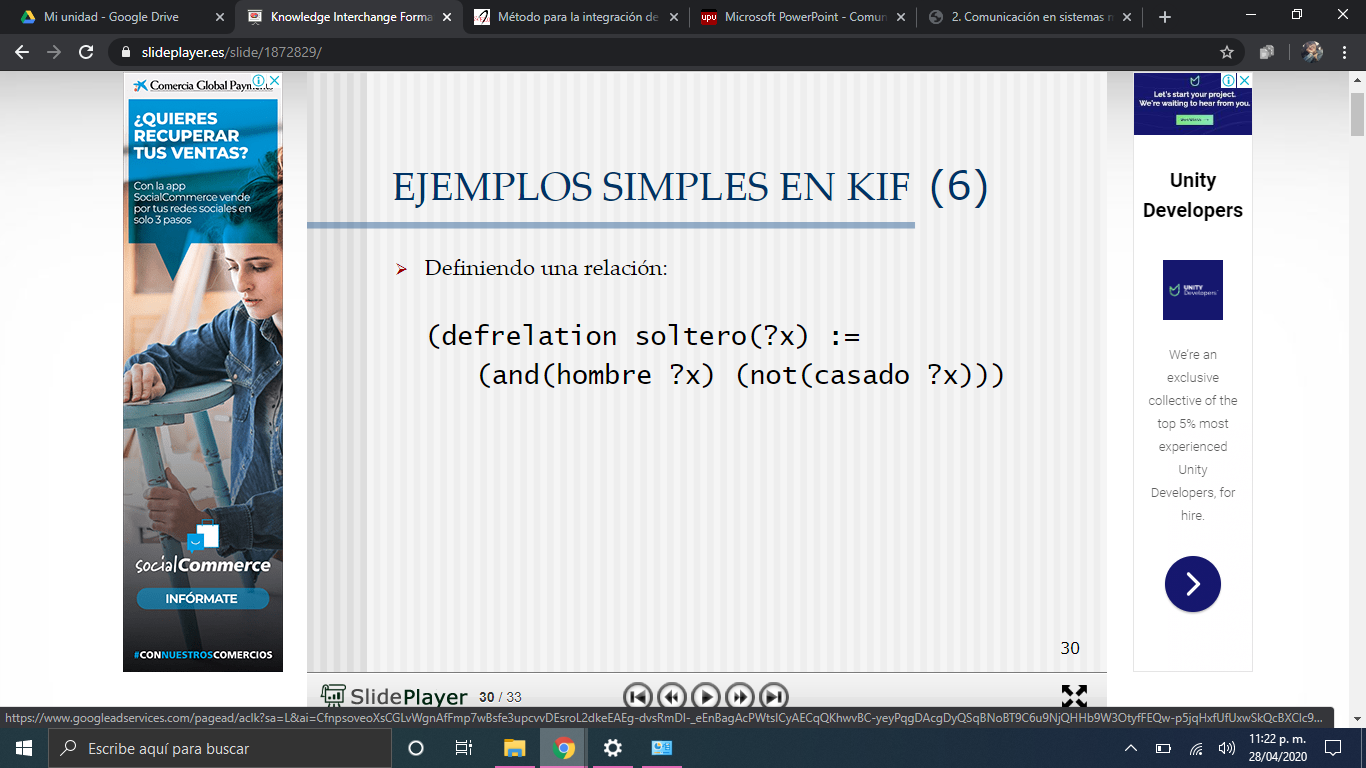
Secuencia de objetos. Los objetos de una lista no necesitan expresiones KIF

**(listof obj1 (listof lob1 lob2 lob3) obj3)**

Funciones para trabajar con listas:

* first
* rest
* last
* reverse
* append
* sublist

EJEMPLO:

****

## 6. FOAF

El vocabulario Friend of a Friend (FOAF) surgió a comienzos del año 2000. Es un vocabulario adecuado para la definición de metadatos sobre personas, sus intereses, sus relaciones y sus actividades. El vocabulario cuenta con un conjunto central de clases (primera letra en mayúscula) y propiedades (primera letra en minúscula):

• Agent – cosas que realizan algo, puede tratarse de personas, organizaciones, robots, etc. Cuenta con las subclases “Person”, “Organization” y “Group”.

• Person – entidad central del vocabulario: representa a las personas.

• name – cadena de caracteres con un nombre.

• title – forma de tratamiento, como, por ejemplo, “Sr.”, “Sra.”, etc.

• img – una imagen que representa a una persona.

• depiction (depicts) – relaciona alguna cosa a una imagen.

• familyName – describe parte del nombre de una persona (apellido).

• givenName – describe parte del nombre de una persona (primer no

• knows – relaciona a dos personas.

• based\_near – relación espacial entre dos cosas.

• age – la edad de la persona.

• made (maker) – alguna cosa hecha por alguien.

• primaryTopic (primaryTopicOf) – principal tópico de un documento.

• Project – un proyecto.

• Organization – una organización.

• Group – un grupo.

• Member – un miembro de un grupo.

• Document – un documento.

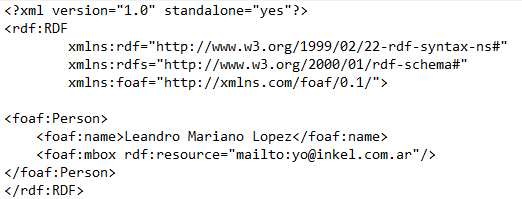
• Image – una imagen.

Además de ese núcleo central, existe una extensión de clases y propiedades relacionadas con las características sociales de la Web, tales como “nick”, “mbox” (e-mail), “homepage”, “publications”, “account”, etc.

Ahora la cuestión es, ¿cómo reconocemos cada elemento? aquí una breve explicación de cómo se hace.

**Identificar personas**

Existen en el mundo cientos de miles de personas que comparten los mismos atributos, como el nombre, color de ojos, pelo, altura, intereses, etc. Sabiendo esto, ¿cómo podemos llegar a distinguir de quién estamos hablando? ¿Como identificar de un modo unívoco a alguien? Pues bien, como FOAF es un proyecto basado en internet, los desarrolladores buscaron algún atributo propio de los usuarios de internet, y lo encontraron: el email. Este funciona como identificador único de alguien. Por ejemplo:



**Expresando relaciones.**

Como se mencionó anteriormente FOAF define relaciones de personas con personas, personas con gustos, etc. Esto se hace con la propiedad knows, por ejemplo:



La persona asociada al email *yo@inkel.com.ar*, de nombre *Leandro Mariano López*, **conoce** a la persona asociada al email *jim@jibbering.com*, cuyo nombre es *Jim Ley*. Como vemos es bastante simple, pero en eso se basa también su efectividad: expresa correctamente, y sin ambigüedades quien conoce a quien.

## 7. RDF

El *lenguaje RDF* es un lenguaje para representar información sobre recursos en la World Wide Web.

Este también se utiliza para representar metadatos pero sobre los recursos web tales como títulos, autor, modificaciones de los datos de la página web, copyright y otras licencias de información sobre documentos web, así como la disponibilidad para algunos recursos compartidos. Se trata de un modelo de datos para objetos (recursos).

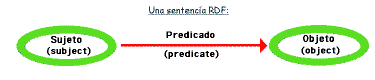
Gracias a la Web Semántica y a la utilización de lenguajes de Metadatos como el RDF la usabilidad y aprovechamiento de la Web y sus recursos interconectados aumentará gracias a una serie de factores:

* Los documentos etiquetados con información semántica (estilo la etiqueta <meta> de HTML utilizada para facilitar el trabajo a los robots). Se pretende que esta información sea interpretada por el ordenador con una capacidad comparable a la del lector humano. El etiquetado puede incluir Metadatos descriptivos de otros aspectos documentales o protocolarios.
* Los vocabularios comunes de Metadatos (ontologías) y mapas entre vocabularios permiten a quienes elaboran los documentos disponer de nociones claras de cómo deben etiquetarlos para que los agentes automáticos puedan usar la información contenida en los Metadatos (por ejemplo el metadato autor debería significar “autor de la página” y no “autor del objeto descrito en la página”)
* Los agentes automáticos que realizan tareas para los usuarios de estos metadatos de la Web Semántica.
* Los servicios web que proporcionan información a los agentes (por ejemplo un servicio de garantías a quien un agente pudiera consultar sobre si un comercio electrónico tiene un historial de mal servicio o de generar correo basura).

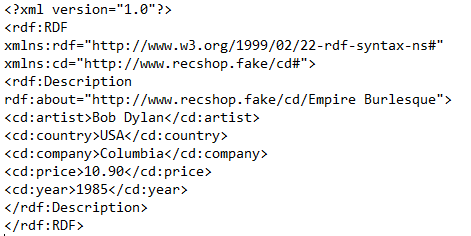
**Estructura**

Este modelo se basa en la idea de convertir las declaraciones de los recursos en expresiones con la forma sujeto-predicado-objeto. El elemento de construcción básica en RDF es el "triplete" o sentencia, que consiste en dos nodos (sujeto y objeto) unidos por un arco (predicado), donde los nodos representan recursos, y los arcos propiedades.

Por ejemplo una sentencia podría expresar el hecho de que el autor (predicado) del libro "En la cripta" (sujeto) fue el escritor H.P. Lovecraft (objeto). Encadenando estos tripletes se construyen grafos o redes semánticas para la web.



En el siguiente ejemplo se utilizará RDF para describir un CD de música:



El documento está basado en XML, con su declaración en la primera línea.

Tras ella, la raíz del documento RDF <rdf:RDF>. Se especifican ahí las URLs de los namespaces xmlns:rdf y xmlns:cd.

El elemento <rdf:Description> contiene la descripción del recurso identificado mediante el atributo rdf:about. Los elementos <cd:artist>, <cd:country>, <cd:company>, etc. son las propiedades del recurso.

**Lenguaje de consulta.** RDF cuenta con un lenguaje que permite ejecutar búsquedas bastantes complejas sobre un grafo mediante una sintaxis sencilla, sin embargo, no se ha estandarizado este lenguaje por lo que existen varias propuestas que pueden servir para ejecutar dichas consultas. Las más sobresalientes y que han prosperado son:

* RDQL, por Hewlett Packard, posiblemente el más extendido.
* RQL, por el instituto ICS-FORTH de Grecia.
* SeRQL, por la empresa holandesa Administrator.
* SPARQL**.**

# Herramientas de desarrollo de ontologías

## 1. PROTÉGÉ

Es una plataforma open source que provee una creciente comunidad con una suite de herramientas para construir modelos de dominio y aplicaciones basadas en conocimiento con ontologías.

Con la amplia adopción de plataformas Web 2.0 y la adopción gradual de ontologías y tecnologías de Web Semántica en el mundo real, necesitamos herramientas de desarrollo de ontología que se adapten mejor a las nuevas formas de interactuar, construir y consumir conocimiento. Los usuarios de hoy dan por sentado la creación de contenido basado en la web y la colaboración en línea. WebProtégé adopta estas herramientas para generar mayores beneficios.

Características

* Soporta el lenguaje de ontologías web OWL 2.
* Con una interfaz simple por default, la cual brinda acceso a los constructores comúnmente usados en OWL.
* Seguimiento completo de cambios e historial de revisiones.
* Herramientas de colaboración, como compartir y notificaciones por email.
* Formularios web personalizables para edición específica de aplicación / dominio
* Soporte para editar ontologías OBO (Open Biological Ontologies)
* Múltiples formatos para cargar y descargar ontologías (RDF/XML, Turtle, OWL/XML, OBO y otros)

Protégé permite:

* Modelar una ontología de las clases que describen un tema particular.
* Creación de una herramienta de adquisición de conocimiento para recoger conocimiento.
* Entrar en casos específicos de datos y de la creación de una base de conocimiento.
* La ejecución de usos.

Actualmente se está utilizando en **medicina clínica**, y **ciencias biomédicas**, aunque puede ser utilizado en cualquier campo donde los conceptos se pueden modelar como jerarquía de clases.

## 2. SWOOP

Es un editor de ontologías **OWL** ligero, anteriormente se mantenía sólo en la Universidad de Maryland, pero ahora se desarrolla en conjunto con la Universidad de Manchester, IBM Watson Research y Clark & Parsia.

Es un editor y navegador que toma de navegador web estándar su paradigma básico de interfaz. Swoop incluye muchas de las características familiares de un navegador web, como una barra de direcciones y botones de historial, marcadores, navegación hipertextual, etc., y los aplica al problema de navegar y editar ontologías basadas en la Web.

Características:

* Es sencillo cargar ontologías desde la web y navegar dentro y entre ellas.
* Se pueden cargar múltiples ontologías al mismo tiempo.
* Las ontologías, clases, propiedades e individuos se representan de manera accesible y de alto nivel.
* Uno puede "ver la fuente" de ontologías y sus entidades en una serie de sintaxis comunes (por ejemplo, RDF / XML, la sintaxis abstracta OWL, Turtle).
* Los razonadores OWL se pueden integrar para subsunción, comprobación de coherencia, etc. Los razonadores predeterminados incluyen un razonador simple similar a RDFS y Pellet, un razonador de cuadros lógicos de descripción.
* Gestión de cambios de ontología con amplios mecanismos de deshacer y deshacer
* Compartir anotaciones sobre ontologías utilizando el protocolo Annotea. También adjunte y distribuya conjuntos de cambios de ontología con anotaciones
* Buscar en múltiples ontologías y 'encontrar todas las referencias' de una entidad llamada OWL
* Comparar entidades usando un Titular de Recursos
* Exportar ontologías directamente a una tienda remota WebDav

## 3. KAON

KAON (The Karlsruhe Ontology and Semantic Web Tool Suite) es un gestor de ontologías de código abierto. Incluye un conjunto de herramientas para crear y gestionar ontologías. Esta herramienta constituye un RDF o El Marco de Descripción de Recursos (del inglés Resource Description Framework, **RDF**).

Su licencia es GPL, y está disponible para plataformas de Windows y Linux, este sistema está desarrollado en lenguaje de programación JAVA.

**El objetivo más importante para KAON es el razonamiento escalable y eficiente de las ontologías para su construcción de la mejor manera.** La herramienta proporciona 2 niveles de gestión:

* OI-Modeler: Es el redactor de la ontología y proporciona la ayuda para la creación y mantenimiento de las ontologías. Las ontologías se pueden corregir simultáneamente por varios usuarios así que **soporta desarrollo concurrente y colaborativo**. **La navegación en una ontología se proporciona con metáforas basadas en gráficos y árboles**. OI-Modeler también proporciona una plataforma abierta y extensible para convivir con otros módulos de software para ontologías. Por lo tanto los usuarios pueden tener acceso a un registro de las ontologías de KAON y a los algoritmos vía OI-Modeler.
* KAON porta: Proporciona un marco simple para navegar y buscar ontologías con browsers de la Web. Mientras que KAON porta se ha desarrollado sobre todo como un frame, espera ser adaptado a los propósitos individuales cambiando la disposición (y posiblemente un cierto código), es completamente funcional como una herramienta independiente.
* KAON 2 es el sucesor del proyecto KAON. La principal diferencia con KAON 1 es el lenguaje de ontología compatible: KAON 1 utilizó una extensión patentada de RDFS, mientras que KAON2 se basa en **OWL-DL y F-Logic**. KAON 2 es un sistema nuevo por lo tanto no es compatible con versiones pasadas de KAON 1.

**Características importantes de KAON 2**

* Proporciona una API para la gestión programática de las ontologías OWL-DL, SWRL y F-Logic.
* Un motor de inferencia para responder consultas conjuntas (expresadas usando la sintaxis SPARQL),
* Una interfaz DIG, que permite el acceso desde herramientas como Protégé,
* Un módulo para extraer instancias de ontología de bases de datos relacionales.
* KAON2 es gratuito para las universidades para uso académico no comercial, sin embargo para fines comerciales, hay una versión comercial de KAON2 llamada OntoBroker OWL.

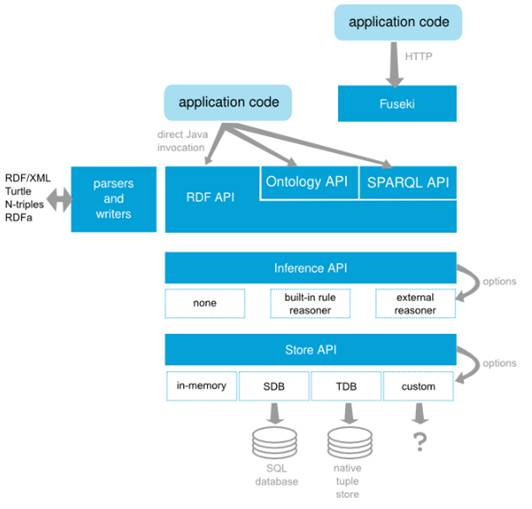
## 4. JENA

JENA es un framework de la Web Semántica(extensión de la web actual dotada de conocimiento) de código abierto para Java. Tiene una API que extrae datos y escribe grafos RDF y ontologías OWL. Los modelos representan un grafo en JENA. Se pueden crear modelos utilizando datos desde URLs, ficheros, bases de datos o combinando distintas fuentes. Para realizar consultas se usa SPARQL(Lenguaje de consulta para RDF).

Es gratuito y de código abierto, utilizado para construir aplicaciones de datos y web semánticos.

Su arquitectura incluye :

* API para trabajar (leer, procesar, escribir) ontologías RDF y OWL
* Motor de inferencia para razonar sobre ontologías RDF Y OWL
* Estrategias de almacenamiento flexible para almacenar tripletas RDF en memoria o fichero
* Motor de consultas compatible con especificación SPARQL



Jena es una API de Java que se puede utilizar para crear y manipular gráficos RDF como este. Jena tiene clases de objetos para representar gráficos, recursos, propiedades y literales. Las interfaces que representan recursos, propiedades y literales se denominan Recurso, Propiedad y Literal respectivamente. En Jena, un gráfico se llama modelo y está representado por la interfaz Modelo.

# Proyecto y Herramienta seleccionada

**Nombre de Proyecto: MerKdo**

Una aplicación que permite ordenar y pagar órdenes de insumos y despensa a domicilio de tiendas y mercados locales cercanos al cliente o dentro de su misma región con la finalidad de apoyar a los negocios locales a vender sus productos y a los consumidores a comprar su despensa desde casa sin la necesidad de salir a buscar en medio del toque de queda por la contingencia producida por el virus que está atacando a la población en la actualidad.

**Herramienta: Jena**

Al ser una herramienta de código libre y que además admite distintos tipos de entradas de datos desde ficheros, bases de datos y enlaces, puede procesar información de muchas fuentes, proporcionando datos estadísticos y encuestas, lo cual enriquecería el desarrollo del estudio de mercado, las necesidades de la población de compra y venta de insumos de cada área, la oferta y demanda y la frecuencia con la que se pueden realizar los envíos. Con esta información se puede definir un modelo estratégico para generar mantener informados a los vendedores sobre el volumen de ventas aproximado que tendrán para que puedan adecuar la cantidad de inventario que manejan y evitar pérdidas así como para planificar las rutas, fechas y horarios de entregas buscando siempre las rutas más rápidas que se traduzcan en menor consumo de combustible o gasto de envío que será abarcado por los clientes.

La evaluación realizada por el proceso ontológico mostrará áreas de oportunidad y mejora con la retroalimentación de los compradores, la adición de nuevos vendedores y recomendaciones para posicionar nuevos puntos de venta estratégicos según las necesidades de la sociedad en cuestión del área facilitando los productos a los consumidores y la venta de insumos a los vendedores

Este framework nos ayudará a almacenar patrones de comportamiento de los consumidores, esto con el fin, de implementar métodos que nos ayuden a ofertar los insumos de una manera más adecuada, estos patrones pueden almacenarse de acuerdo con el horario de compra, día, la cantidad y el tipo de productos que los usuarios consultan en la aplicación, entre otros.

# Criterios de Evaluación de las Ontologías.

Los criterios de evaluación de una ontología son muy similares a los de la aprobación de un sistema de tecnología, se debe tener un juicio técnico y tener en cuenta los requerimientos, y objetivos de la ontología.

La evaluación tiene como tarea principal la validación de esta desde la construcción de definiciones, requerimientos y las respuestas que se han establecido, se refiere a las definiciones de la ontología y como estas lo modelan con tal de cumplir con el objetivo para las que fueron creadas. Otros puntos importantes por considerar son:

* Contenido y organización
* Lenguaje Utilizado para codificación
* Metodología de Desarrollo
* Herramientas de Software
* Costo y Construcción

Es importante mencionar que también están consideradas diferentes tipos de metodologías, algunos autores han decidido agrupar los métodos esenciales de evaluación en función a la filosofía principal que motiva dicha evaluación, la que consideramos más importante es el empleo de criterios clásicos en la recuperación de información. La primera variante utiliza dos métricas estándar en la recuperación de información para valorar la ontología son la **precisión**: proporción de material relevante recuperado ya sea por medio de clases o subclases o por medio de objetos asociados a éstas, y **exhaustividad**: capacidad de la ontología para recuperar objetos.

El principal problema que presentan estos métodos de evaluación es la necesidad de un entrenamiento previo de los observadores, para que puedan valorar las conclusiones de manera objetiva. En muchas ocasiones, los resultados no son los más factibles, debido a que tanto precisión como exhaustividad son dos cuestiones que se deben interpretar desde el punto de vista de la persona que interactúa con el sistema.

La segunda variante se emplea una ontología base, que es comparada con la ontología que se desea generar. La idea sobre la que se construye ese sistema es el de evaluar el cuerpo de los textos empleados en la creación de la ontología en relación con las instancias de dicha ontología, construyendo un clúster que permita valorar el nivel de consistencia semántica alcanzado.

Los resultados deberían revelar si la metodología empleada para el diseño de la ontología ha sido la idónea, o si el cuerpo con el que se ha trabajado cumple con los estándares correctos, pero no se centra demasiado en valorar el nivel de recuperación de información obtenido, o las inconsistencias semánticas.

# Conclusión

En la actualidad existen diferentes formas para realizar procesos de administración de conocimiento desde la generación hasta la aplicación de este. El uso de lenguajes de representación es una forma viable para comprender cómo está organizada la información que poseemos también desarrollar ontologías de forma cooperativa basado en la integración de piezas de conocimiento. Las ontologías son un tópico común de investigación en varias comunidades, tales como ingeniería del conocimiento, procesamiento de lenguaje natural, sistemas de información cooperativos, integración inteligente de información y gestión del conocimiento, proveen una comprensión compartida y consensuada del conocimiento de un dominio que puede ser comunicada entre personas y sistemas heterogéneos. Las ontologías son teorías formales acerca de un dominio de discurso y por eso requieren de un lenguaje lógico formal para ser expresadas. En el área de IA se han desarrollado muchos lenguajes para este fin. El objetivo de estos procesos de integración es obtener ontologías que contengan el mejor y mayor conocimiento posible, de forma que dichas ontologías puedan ser consideradas apropiadas para tener el dominio de la información en cuestión.

Siempre que se hayan construido buenas ontologías, se pueden usar para desarrollar sistemas de gestión del conocimiento. También se han identificado diferentes herramientas que son eficientes para representar el conocimiento en opinión como equipo hemos llegado a la conclusión en que JENA se ha convertido en una herramienta escalable que puede ser usada en cualquier tipo de ámbito para la representación de la información este tipo de herramientas consideramos el punto más necesario para los desarrolladores de sistemas de gestión del conocimiento para su creación. El objetivo de la ingeniería ontológica es la creación de esquemas conceptuales que permiten desarrollar la representación formal de un grupo de conceptos y sus relaciones entre ellos también siendo estas el punto más fuerte para trabajar actualmente.

Finalmente otro punto importante es el mismo éxito de la web semántica, La idea es que los datos puedan ser utilizados y comprendidos por los ordenadores sin necesidad de supervisión humana, de forma que los agentes web puedan ser diseñados para tratar la información situada en las páginas web de manera semiautomática. Los lenguajes de representación tienen semánticas que generan una base para razonar sobre el significado de una expresión, sirviendo en el modelado de ontologías, que ofrecen la unión de datos entre comunidades descriptivas, siendo comprensibles por las máquinas, en donde su mayor virtud está en la estructuración de los contenidos en la web.

# Referencias

1. Colaboradores de MDN. (2019). Introducción a XML. abril 30, 2020, de MDN Web Docs Sitio web: [https://developer.mozilla.org/es/docs/Web/XML/Introducción\_a\_XML](https://developer.mozilla.org/es/docs/Web/XML/Introducci%C3%B3n_a_XML)
2. Extensive Markup Language. (s.f.). En Wikipedia. Recuperado el 30 de abril del 2020 de <https://es.wikipedia.org/wiki/Extensible_Markup_Language>
3. Extensive Markup Language (XML). (s.f.). En W3C Information and Knowledge Domain. Recuperado el 30 de abril del 2020 en <https://www.w3.org/XML/>
4. Fernández , R. (s. f.). Herramientas y lenguajes para la definición de ontologías en la web . Recuperado 27 de abril de 2020, de <http://personales.upv.es/ccarrasc/doc/2004-2005/LaWebSemantica-FernandezRoigSetal/www/tresHerr.html>
5. Simple HTML Ontology Extension, S. (s. f.). SHOE. Recuperado 27 de abril de 2020, de <http://www.cs.umd.edu/projects/plus/SHOE/>
6. VARGAS MONTOYA, H. (s. f.). HERRAMIENTA INFORMÁTICA BASADA EN ONTOLOGÍAS PARA LA CLASIFICACIÓN AUTOMÁTICA DE INFORMACIÓN DE SEGURIDAD . Recuperado 27 de abril de 2020, de <https://repository.eafit.edu.co/bitstream/handle/10784/2770/Hector_VargasMontoya_2005.pdf?sequence=7&isAllowed=y>
7. S. A (s. f) Comunicación entre agentes Recuperado 28 de abril de 2020, de <http://www.upv.es/sma/teoria/sma/Comunicaci%F3n%20entre%20agentes.pdf>
8. S. A(s, f) Capítulo 2. Comunicación en sistemas de agentes y bibliotecas digitales. Recuperado 28 de abril de 2020, de <http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/barceinas_g_a/capitulo2.pdf>
9. Apache Jena (s. f.) Apache Jena : An Introduction to RDF and the Jena RDF API. Recuperado 29 de abril de 2020 de <http://jena.apache.org/tutorials/rdf_api.html>
10. Lapuente Lamarca, M (2018). La web semántica: Hacia la web semántica. Recuperado 29 de abril de 2020 de <http://www.hipertexto.info/documentos/web_semantica.htm>
11. Benítez Andrades, J (2014) Blog: Investigación: Jena, DLEJena y JOINT. Recuperado 29 de abril de 2020, de <http://phd.jabenitez.com/2014/06/27/investigacion-jena-dlejena-y-joint/>
12. WebProtégé. (s.f.). En ProtegeWiki. Recuperado el 30 de abril del 2020 de <https://protegewiki.stanford.edu/wiki/WebProtege>
13. Tudorache T., Nyulas C., Noy N. F. & Musen M. A.. (2011). WebProtégé: A Collaborative Ontology Editor and Knowledge Acquisition Tool for the Web. abril 30, 2020, de IOS Press Sitio web: [http://www.semantic-web-journal.net/sites/default/files/swj210\_1.pd](http://www.semantic-web-journal.net/sites/default/files/swj210_1.pdf)f
14. Hurtado, M. (s. f.). Lenguaje RDF. Recuperado 1 de mayo de 2020, de <http://lawebdelosdatos.blogspot.com/p/lenguaje-rdf.html>
15. Vocabularios y Ontologías. (s. f.). Recuperado 1 de mayo de 2020, de <https://ceweb.br/guias/web-semantica/es/capitulo-6/>
16. Duran, J., Conesa, J. & Clarisó, R.. (s.f). Ontologías y Web Semántica. 30/04/2020, de Universidad Oberta de Catalunya Sitio web: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/69005/1/Representaci%C3%B3n%20del%20conocimiento_M%C3%B3dulo%202_Ontolog%C3%ADas%20y%20web%20sem%C3%A1ntica.pdf>
17. Silva, L.. (Enero 2004). Representación de ontologías en la web semántica. 30/04/2020, de Instituto de Informática – Universidad de Federal do Rio Grande do Sul (UFRGS) Sitio web: <https://www.researchgate.net/publication/228541846_REPRESENTACION_DE_ONTOLOGIAS_EN_LA_WEB_SEMANTICA>
18. (s.a). ((s.f)). Herramientas y lenguajes para la definición de ontologías en la web. 30/04/2020, de upv.es Sitio web: <http://personales.upv.es/ccarrasc/doc/2004-2005/LaWebSemantica-FernandezRoigSetal/www/tresHerr.html#OWL>
19. Parra, J.. (2013). Temario de Bibliotecas Universitarias (Galicia). Madrid España: Lulu. <https://books.google.com.mx/books?id=KvHCDwAAQBAJ&pg=PA602&lpg=PA602&dq=oil+lenguaje+de+representacion&source=bl&ots=fbh4IODNq3&sig=ACfU3U3XwcbTmdlWVy4AUywfmpwOTSaDrg&hl=es-419&sa=X&ved=2ahUKEwiikc_CyZHpAhVM-6wKHRdPDdY4ChDoATAFegQICRAB#v=onepage&q=oil%20lenguaje%20de%20representacion&f=false>
20. Talavera Pereira, Rosalba, & Marcano Aular, Yelitza Josefina. (2007). Los lenguajes de representación semántica y su uso en la construcción de ontologías. *Revista de Ciencias Sociales*, *13*(1), 59-71. Recuperado en 01 de mayo de 2020, de <http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S1315-95182007000100005&lng=es&tln=es>.
21. (s.a). (diciembre 11, 2009). SWOOP. 30/05/2020, de w3c semantic Web Sitio web: <https://www.w3.org/2001/sw/wiki/SWOOP>
22. Kalyapur, A., Parsia, A. & Sirin, E.. (Enero 4, 2006). Swoop. 01/05/2020, de University of Maryland Sitio web: <https://github.com/ronwalf/swoop>
23. (s.a). (Abril 27, 2008). Swoop. 01/05/2020, de semanticweb.com Sitio web: <https://www.semanticweb.org/wiki/Swoop.html>