UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

La Universidad Católica de Loja

ÁREA TÉCNICA

TITULO DE INGENIERO EN SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

Desarrollo de una solución web para el proceso de recolección de datos en internet a datos semánticos apoyados en la visualización.

TRABAJO DE TITULACIÓN

AUTORA: Cárdenas Cabrera, Ana Cristina

DIRECTOR: Ramírez Coronel, Ramiro Leonardo

LOJA – ECUADOR

2018

2018

APROBACIÓN DEL DIRECTOR DE TRABAJO DE TITULACIÓN

Ingeniero.

Ramiro Leonardo Ramírez Coronel

DOCENTE DE LA TITULACIÓN

De mi consideración:

El presente trabajo de titulación **Desarrollo de una solución web para el proceso de recolección de datos en internet a datos semánticos apoyados en la visualización** realizado por Ana Cristina Cardenas Cabrera ha sido orientada y revisada durante su ejecución por cuento se aprueba la presentación del mismo.

Loja, abril de 2018

f)……..

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

Yo **Ana Cristina Cardenas Cabrera** declaro ser autor del presente trabajo de fin de titulación: **Desarrollo de una solución web para el proceso de recolección de datos en internet a datos semánticos apoyados en la visualización,** de la titulación Sistemas Informáticos y Computación siendo Ramiro Leonardo Ramírez Coronel director del presente trabajo; y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales. Además, certifico que las ideas, conceptos procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

Adicionalmente declaro conocer y aceptar la disposición del Art. 88 del Estado Orgánico de la Universidad Técnica Particular de Loja que en su parte pertinente dice: “Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos, o técnicos y tesis de grado o trabajos de titulación que se realicen con el apoyo financiero académico o institucional (operativo) de la Universidad”

f.……

Autor: Ana Cristina Cardenas Cabrera

Cedula: 1105213605

DEDICATORIA

El presente trabajo es una meta personal en la que muchas personas han intervenido para este sueño, primeramente agradezco a Dios por permitirme llegar a esta etapa de mi vida y darme fuerzas para enfrentar cada obstáculo durante este camino.

Agradezco a mis padres quienes han sido un pilar fundamental en mi carrera universitaria, por el apoyo emocional entregado día a día, muchas gracias por ser unos padres ejemplares que enseñan con los actos.

A mis abuelitos y hermanos quienes han estado conmigo en este proceso maravilloso de mi vida, que con cada palabra de aliento me motivaban y me daban muchos concejos.

A mis amigos y demás familiares que sin la ayuda de ellos esta meta no se hubiera cumplido.

AGRADECIMIENTO

Agradezco a primeramente a Dios por darme esta vida tan maravillosa, a mis padres, abuelitos, tíos, amigas y amigos por todo el apoyo brindado en esta etapa de mi vida, todos han aportado en mi valores y pensamientos que nunca olvidare y los tendré presente por el resto de mi vida, además agradecer a los docentes de la universidad por todas las enseñanzas brindadas, especialmente a mi director de trabajo de fin de titulación por la confianza y el apoyo brindado.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

Actualmente la Web es uno de los principales portadores de información, una de las desventajas es que la representación de la información solo es comprensible por los seres humanos, excluyendo a las maquinas del procesamiento de los datos.

Hoy en día las herramientas que integren las funcionalidades de recolección de datos hasta la conversión RDF son escasas, es por ello se propone dar solución haciendo la integración de herramientas capaces de involucrar al usuario en todo el proceso (Scrapy, Selenium, Jquery).

Haciendo uso de la metodología SCRUM para el desarrollo del trabajo se creó una aplicación web capaz de extraer datos de páginas web que tengan filtrado de información. Una de las principales características de la propuesta es en base a los atributos propios de HTML (clases). Además contiene un módulo de mapeo de datos de CSV a serializaciones RDF, haciendo uso del API de LOV.

Para validar el proceso de recolección y transformación de datos la aplicación web pasa sobre diferentes pruebas, validando la usabilidad y el funcionamiento.

Palabras clave: Recolección, Datos, RDF, Mapeo, Transformación, Scrapy, Selenium.

ABSTARCT

Actually the Web is one of the main carriers of information, one of the disadvantages is that that the representation of information is only understandable by humans, excluding machines from data processing.

Nowadays, the tools that integrate the functionalities of data collection until the RDF conversion are insufficient, that is why it is proposed to provide a solution by integrating capable tools de involucrar al usuario en todo el proceso (Scrapy, Selenium, Jquery).

Making use of the SCRUM methodology for the work development, a web application was created capable of extracting data from web pages that contain information filtering or searching. One of the main characteristics of the proposal is based on HTML own attributes (classes). It also contains a module for mapping CSV data to RDF serializations, reusing vocabularies of the Semantic Web with the use of the LOV API.

To validate the data collection and transformation process, the web application passes over different tests, validating usability and functioning.

Keywords: Data collection, RDF, data mapping, Transformation to RDF, Scrapy.

INTRODUCCIÓN

La Web desde sus inicios se ha convertido en uno de los principales portadores de información. Desde la llegada de la Web 2.0 el usuario recibe el rol más importante ya que se convierte en el principal gestor de la información, y la interpretación solamente se realiza por los usuarios que acceden a un sitio Web. Uno de los principales problemas de esta Web es la sobrecarga de la información que es generada por el usuario, la mayoría de sitios web brinda información en un formato "no estructurado" el cual limita el uso automático de los datos. Además la Web actual posee estructuras heterogéneas y la terminología utilizada en cada sitio web conlleva a un problema de sinonimia.

Uno de los desafíos más grandes es representar la información que pueda ser entendida por máquinas y personas, es por ello que nace la Web 3.0, que busca agregar semántica a los datos mediante ontologías, una solución óptima es que la mayoría de página Web utilicen estándares sintácticos dentro de su HTML, una de las recomendaciones es utilizar el tag <meta> que ayuda en el proceso de comunicación entre quienes publican datos y quienes los consumen.

Esta investigación tiene dos objetivos principales que dan solución a dos grandes problemas de la web de hoy en día:

- **Sobrecarga de información**: Dar solución a la extracción automática de datos que se encuentran alojados en la WEB en formato HTML mediante la fusión de herramientas de recolección de datos (Scrapy, Selenium) involucrando al usuario para que realice una supervisión de los datos a recolectar, además se tiene en consideración las limitaciones existentes dentro de herramientas que brindan un servicio similar.
- **Estructuración y especificación de semantica:** Brindar al usuario un espacio de trabajo donde se permita la trasformación de datos estructurados a datos semánticos, teniendo en cuenta los principios de Linked Data. Además el proceso de dar semántica a los datos, donde se tiene en consideración aspectos importantes como: reuso de vocabularios, para cubrir este aspecto se hace uso de catálogos en línea que ayudan a realizar búsquedas de vocabularios existentes.

OBJETIVOS

Para el presente trabajo de titulación se definen los siguientes objetivos.

Objetivo general

Desarrollar una solución web para el proceso de recolección y conceptualización de datos de internet a datos semánticos apoyados en la visualización.

Objetivos específicos

- Implementar un recolector de datos de diferentes sitios web que se encuentran en formato HTML para generar un almacén de datos aplicando técnicas de extracción (Scrapy) en base a patrones de búsqueda.
- Representar los datos obtenidos en un esquema común normalizado sobre un dominio de conocimiento (creación o reuso de vocabularios).
- Automatizar el proceso de transformación de datos semánticos para el usuario.

ESTRUCTURA DEL DOCUMENTO

El presente trabajo de fin de titulación se estructura con cinco capítulos los cuales hacen referencia a agrupaciones de contenido relevante para esta investigación:

En el **Capítulo 1** se realiza la parte fundamental teórica para desarrollar el TT donde se cubre temas referentes a la Web Semántica describiendo su historia, arquitectura y conceptos claves de esta área, en esta sección se ve necesario realizar un análisis de herramientas en línea del proceso de transformación de los datos a datos enlazados, dando como resultado la parte adicional que contiene el TT.

En el **Capítulo 2** se detalla la solución propuesta para alcanzar los objetivos planteados, donde se menciona las etapas y las herramientas que se utilizan para el cumplimiento de una solución óptima, además se establece la metodología con la cual se desarrolla la aplicación.

En el **Capítulo 3** se detalla cada una de las fases propuestas en la metodología, así mismo se muestra la arquitectura con la que se construye la aplicación, además se realiza las pruebas necesarias para asegurar que el proyecto realizado cumpla con los requisitos del usuario, además se describen las conclusiones y recomendaciones del proyecto.

En el **Capítulo 4** se realiza las pruebas necesarias para validar el desarrollo de la solución al problema planteado, además se menciona los resultados obtenidos, las conclusiones y recomendaciones que se puede dar al finalizar el TT.

CAPITULO I: MARCO TEORICO

1.1. Introducción

El presente capítulo describe al marco teórico, el cual abarca temas referentes a los principales conceptos básicos que permiten el desarrollo de este trabajo:

1.2 La Web: Se realiza una introducción de la Web desde su comienzo hasta la Web actual, permitiendo la introducción al TT.

1.3 Web Semántica: Se hace referencia a los componentes principales y los estándares de representación que existen dentro de la Web, además se explica la arquitectura de la Web Semántica donde se realiza una conceptualización de cada nivel.

1.4 Vocabularios y ontologías: Dentro de esta sección se menciona la importancia de los vocabularios y ontologías dentro de la Web Semántica, además de exponer los vocabularios más relevantes hoy en día.

1.5 Herramientas: Se realiza una breve descripción de herramientas, frameworks y APIs que ayudan a realizar la recolección de datos de internet y la consulta de vocabularios existentes.

1.6 Trabajos relacionados: Se realiza un estudio de las herramientas en línea que facilitan la recolección de datos y la transformación a RDF.

1.7 Comentarios finales: Se realiza una pequeña descripción del resultado del capítulo mencionando los aspectos principales.

1.2. La Web

Cuando se habla de la Web nos referimos a información que se encuentra alojada en un determinado sitio web. Es una red de información que está representada en un lenguaje de marcado para hipertextos (HTML) el cual es un elemento básico para la construcción de una página web, la idea de gestionar la información fue propuesta por Tim Berners-Lee en 1989, donde su idea principal en ese tiempo era compartir información entre investigadores, esta idea ha sufrido muchos cambios en las que se ve reflejada la representación de la información, a continuación se detalla la evolución de la web, donde se describe características importantes de cada etapa.

Tim Berners-Lee crea la World Wide Web basada en un sistema de hipertexto para compartir documentos.Esta Web se la etiqueto como **1.0**, la cual permite clasificar información donde enlaza documentos localizados en la red. Una de las desventajas es la falta de actualización en las páginas, así mismo no existían productores de contenidos, esto llevo a cabo una nueva generación llamada **Web 2.0** o Web social, en este punto se pasa de una Web informativa a una Web donde cualquier usuario puede participar. (Zajicek, 2007) lo define al usuario como el nuevo rey de internet donde controla la era de la información y deja de ser un espectador y

consumidor de lo que ofrece internet a convertirse en creador y generador de contenidos y servicios, esta Web dio cabida a nuevos proyectos muy utilizados hoy en día como: YouTube, Wikipedia, Facebook, Blogger, Google entre otros, donde no serían nada sin la participación activa de los usuarios, una de las desventajas de esta Web es la sobrecarga y la heterogeneidad de la información, en donde Berners Lee propone mejorar la Web 2.0 remplazándola por la **Web 3.0** o Web Semántica basada en la idea de añadir metadatos semánticos y ontológicos. (Lyssania, Macías, Layla, 2009) menciona que esta Web permite estructurar la información de la manera más similar posible a como los humanos almacenan datos en el cerebro (a través de mapas cognitivos).

Uno de los enfoques muy importantes de la Web lo menciona (Butler, 2006) donde expone que la gran parte de la información del mundo es accesible a través de protocolos y formatos abiertos, en la cual se puede encontrar un sinnúmero de documentos que prestan información muy importante para el usuario, dependiendo el uso que le dé a la información, actualmente existen millones de recursos que puede producir una búsqueda en internet produciendo la heterogeneidad de la información dentro de la Web, (Hewson & Stewart, 2016) realiza una introducción al uso del internet como principal fuente de información y una mayor exploración a los documentos alojados dentro de la Web, estos documentos están escritos en un lenguaje de etiquetas denominado HTML utilizado para definir la estructura de una página web, a continuación se describe los principales estándares de a Web .

- **Identificador de Recurso Uniforme (URI):** Es el encargado de identificar los recursos de la web una manera única.
- Protocolo **de Transferencia de Hipertexto:** Permite la interacción entre cliente y servidor realizando peticiones para generar un resultado.
- **Lenguaje de Marcado de Hipertexto (**HTML**):** Es un lenguaje de marcado para elaborar páginas web, el cual utiliza etiquetas para estructurar la información.

Para realizar una petición a un servidor que aloja documentos se lo realiza a través de peticiones HTTP por medio de un navegador, el cual construye un Modelo de Objetos del Documento (DOM) este modelo fue construido para documentos válidos HTML, el objetivo del DOM es navegar por toda la estructura de un documento teniendo acceso a la información de una página web (W3C, 2014a). Dentro de la arquitectura del DOM se encuentran alojadas las XPATH las cuales ayudan a navegar en la estructura de la página web, además de ser una herramienta muy precisa para la extracción de datos en la web (Scrapy, 2016). A continuación se realiza una conceptualización más a fondo de la Web Semántica.

1.3. Web Semántica

Hoy en día la Web se presenta en formato comprensible para personas pero no por maquinas. La Web actual está limitada a la comprensión humana, en la cual existen millones de recursos que pueden ser accedidos independientemente de la situación geográfica o el idioma, (Consortium & Others, 2014) sostienen que existen dos graves problemas en la Web actual: sobrecarga de información y heterogeneidad de fuentes de información con el consiguiente problema de interoperabilidad.

(Castells, 2005) manifiesta que la Web Semántica rescata la noción de la ontología del campo de Inteligencia Artificial. Por otra parte (Gyrard, Patel, Datta, & Ali, 2017) menciona que es factible mitigar la heterogeneidad, brindando interoperabilidad semántica para construir aplicaciones que brinden soluciones inteligentes. La (Consortium & Others, 2014) define a la Web Semántica como una Web extendida dotada de mayor significado en la que cualquier usuario en internet puede encontrar respuestas a sus preguntas de forma más rápida y sencilla gracias a una información mejor definida, en cambio (Berners-Lee & Miller, 2002) señalan que “La Web Semántica es una extensión de la Web actual en la que a la información disponible se le otorga un significado bien definido que permita a los ordenadores y las personas trabajar en cooperación”.

Está basada en la idea de proporcionar en la web datos definidos y enlazados, permitiendo que aplicaciones heterogéneas localicen, integren y reutilicen la información presente en la Web”.

(Pastor Sanchez, 2011) define una filosofía de trabajo donde menciona la configuración base de trabajo de la Web Semántica:

- Utilizar modelos de metadatos para describir recursos de información.
- Uso de vocabularios RDF para representar metadatos
- Desarrollo de Esquemas RDF y ontologías para describir relaciones entre los recursos descritos e incluso entre las propiedades utilizadas para caracterizarlos
- Localización, interconexión y reutilización de fuentes de datos RDF, lo que permite su integración mediante procesos automáticos.
- Inferir nueva información a partir de las relaciones lógicas que se establecen entre los datos. De este modo se puede desarrollar herramientas y agentes inteligentes

1.3.1. Componentes principales de la Web Semántica.

Dentro de la Web Semántica existen componentes que facilitan la representación de los datos encontrados en la Web, a continuación se detalla cada uno de ellos:

1.3.1.1. RDF.

RDF fue desarrollado por la W3C. La(W3C, 2014b) lo define como “Un marco para la representación de la información en la Web” por lo tanto sirve para modelar metadatos. RDF es utilizado para identificar los recursos en la Web, RDF permite realizar afirmaciones sobre recursos, en donde cualquier cosa puede ser llamada recurso, tanto cosas concretas como abstractas. (Laufer, 2015) manifiesta que las afirmaciones son llamadas tripletas que tienen una estructura de sujeto-predicado-objeto. A continuación, se presentan las serializaciones dentro del modelo RDF, actualmente existen representaciones sintácticas de las cuales algunas de ellas son más comprensibles para personas y otras toman el enfoque adecuado para el procesamiento del computador.

- **N-Triples**

Es un formato de texto para serializar grafos RDF. La (W3C, 2001) menciona que este formato es utilizado para representar las respuestas correctas para analizar la sintaxis de RDF/XML, donde cada línea representa un triple, además menciona que el tipo de datos cadena no se especifica explícitamente.

- **JSON-LD**

Este formato provee una sintaxis JSON, además de ser una sintaxis liviana está diseñado para ser interpretado como datos enlazados, JSON-LD está destinado para ser usado en entornos de programación basados en la Web.

- **RDFa**

Este formato permite que los metadatos se incluyan en páginas Web. Utiliza una sintaxis para especificar las tripletas RDF, independientemente del tipo de datos al cual pertenece el recurso y del vocabulario utilizado.

1.3.1.2. RDFS.

Proporciona un vocabulario de modelado de datos para RDF, dotando de semántica a los datos. La (W3C, 2004a) menciona que es una extensión de RDF, la cual proporciona mecanismos para la descripción de recursos relacionados. RDFS cuenta con mecanismos que permiten especificar que determinadas URIs indican propiedades de recursos, o que determinados recursos identificados por URIs pertenecen a una determinada clase.

1.3.1.3. OWL.

RDF define un modelo de datos, basado en tripletas, para que exista una definición de jerarquías de clases, propiedades, dominios y rangos nace RDFs el cual es una extensión de RDF, así mismo es necesario definir restricciones a la información, de cómo será representada, para que todo tenga un significado, restringir el número de distintas formas de interpretación posibles respecto a un dominio, es por eso se crea OWL, lenguaje que ofrece un conjunto de restricciones para tener una única manera de representar la información.

Uno de los objetivos de la Web Semántica es dotar de significado a la Web, es decir que las maquinas puedan procesar información de manera automática. (W3C, 2004b) menciona que OWL está diseñado para aplicaciones que necesiten procesar información en lugar de solo representar contenido comprensible para los seres humanos.

1.3.2. Arquitectura de la Web Semántica.

Los principales componentes de la Web Semántica son los estándares de representación XML, XML Schema, RDF, RDF Schema y OWL en la Figura 1, se presentan cada uno de los componentes principales de la web semántica los cuales son esenciales para la gestión de datos enlazados.

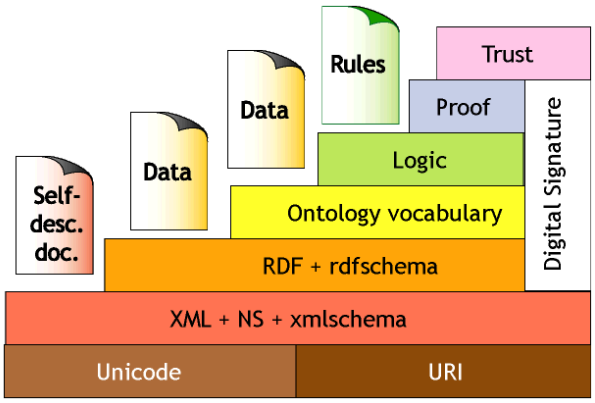


Figura 1. Arquitectura de la Web Semántica
Fuente: W3C (2014)
Elaboración: W3C(2014)

En la Tabla 1 se describe cada una de las capas de la arquitectura de la web semántica donde se detalla las partes importantes de cada componente.

Tabla 1. Descripción de la arquitectura de la Web Semántica

Nombre de Capa	Descripción
Unicode +URI	***UNICODE*** es un estándar que proporciona un número único para cada carácter, sin importar la plataforma ni el programa, es decir codifica la información en cualquier idioma

	URI es un identificador de recursos uniformes el cual permite acceder a cualquier recurso web
XML+ NS+ xmlschema	Esta capa es la más técnica de la web semántica en donde se encuentran agrupados las diferentes tecnologías que posibilitan la comunicación entre agentes
	XML ofrece un formato común para el intercambio de documentos
	NS proporciona un método para cualificar elementos y atributos de nombres
RDF+rdfschemar	Define el lenguaje universal con el que podemos expresar diferentes ideas de la web semántica, el cual define un modelo de datos para describir recursos mediante tripletas sujeto-predicado y objeto.
Ontologías	Las ***ontologías*** son especificaciones formales de un dominio de conocimiento la cual se identifica con una taxonomía.
Logic	Son las reglas formales que permiten determinar si un razonamiento se sigue de sus premisas.

Fuente: (W3C, 2001)

Elaboración: (W3C,2001)

1.4. Vocabularios y Ontologías

Hoy en día existen varias formas de representar el conocimiento. La Web Semántica establece un conjunto de vocabularios de modo que facilite el intercambio de los metadatos. La (W3C, 2004b) menciona que el lenguaje OWL de ontologías está diseñado para ser usado en aplicaciones que necesitan procesar el contenido de la información en lugar de únicamente representar información para los humanos. (Gruber, 1995) describe a una ontología como "una especificación explícita y formal sobre una conceptualización compartida" ya que las ontologías definen conceptos y relaciones de algún dominio en particular.

(Pastor Sanchez, 2011) menciona seis componentes claves para formalizar ontologías: clases, atributos, relaciones, funciones, axiomas e instancias.

- ***Clases o Conceptos:*** Son ideas que se intentan formalizar; además un concepto puede ser algo sobre lo que se dice.
- ***Atributos:*** Los atributos representan la estructura interna de los conceptos, es decir son propiedades que describen conceptos.
- ***Relaciones:*** Son enlaces entre conceptos de un mismo dominio.

- ***Instancias:*** Se las utiliza para representar elementos de un determinado concepto.
- ***Axiomas:*** Son expresiones que siempre son ciertas y que se declaran sobre relaciones que deben cumplir los elementos de la ontología,

En la Web existen “catálogos” que son de gran ayuda para la búsqueda de vocabularios entre los más destacados están: LOV, BioPortal, JoinUp.

Los vocabularios están descritos por un documento señalado por una URI y define un conjunto de clases y propiedades de un dominio determinado. En este apartado se menciona vocabularios más populares y reconocidos:

- **Dublín Core**

Es un vocabulario para la descripción de metadatos en el ámbito bibliotecario, su principal objetivo fue crear un conjunto de elementos que permite la descripción de recursos electrónicos, con el fin de facilitar la búsqueda y la recuperación (Senso & Piñero, n.d.). Por otra parte (Ortiz-Repiso Jiménez, 1999) lo define como un vocabulario que puede ser usado para la documentación de páginas Web.

- **FOAF**

Proporciona un “diccionario” de términos o también denominado vocabulario para la definición de metadatos sobre personas: permitiendo definir sus intereses, relaciones, y actividades. (Brickley & Miller, 2009) afirma que FOAF está diseñado para ser utilizado con otros “diccionarios” y para usarse con una amplia variedad de herramientas y servicios que han sido creados para la Web Semántica.

- **SKOS**

Es una recomendación de la W3C la cual ofrece una manera de representar esquemas de clasificación. Este modelo permite mapear conceptos de diferentes esquemas y agrupaciones de conceptos, (Sanchez, 2016) menciona que “SKOS se ha diseñado para crear nuevos sistemas de organización o migrar los ya existentes adaptándolos a su uso en la Web Semántica de forma fácil y rápida”

1.4.1. Linked Data.

Una de las recomendaciones de la W3C es utilizar un método de publicación de datos enlazados, (Bizer, Heath, & Berners-Lee, 2009) definen el término de Linked Data como la mejor practica para publicar y conectar datos estructurados en la Web, además de ofrecer una manera sencilla de acceder a los datos.

Berners-Lee propone una serie de 4 criterios de diseño para publicar datos en la Web con el fin de que todos los datos publicados formen parte de un espacio global:

- URIs para identificar los recursos en la web las cuales ofrecen una abstracción del lenguaje natural, además de evitar ambigüedades y así ofrecer una forma estándar y univoca para referirse a cualquier recurso.
- Aprovechar el protocolo HTTP para asegurar que cualquier recurso pueda ser buscado y accedido en la web, además se debe tener en cuenta que los URIS no son solo direcciones, son identificadores de los recursos.
- Ofrecer información sobre los recursos usando RDF.
- Incluir enlaces a otras URIs esta regla es necesaria para conectar los datos de tal manera que no se queden aislados y así se pueda compartir información con fuentes externas.

(Berners-Lee, 2010) afirma que estas reglas son expectativas de comportamiento donde romperlas no destruye nada, pero se pierde la oportunidad de conectar los datos, dado que limita la reutilización.

1.4.2. Las 5 estrellas de Linked Data.

Tim Berners-Lee sugiere un esquema de despliegue denominado “5 estrellasque ilustra una evolución de los datos hasta conseguir datos enlazados. Como se muestra en la Figura 2, el esquema está compuesto por cinco niveles donde cada evolución define el método de publicación de los datos correspondiente a un nivel.

Figura 2. Las 5 estrellas de Tim Berners-Lee
Fuente: (Berners-Lee, 2010)
Elaboración: Berners-Lee

- **Nivel 1:** Los datos deben estar en la web en cualquier formato, pero con una licencia abierta.
- **Nivel 2:** Los datos deben estar disponibles como datos estructurados legibles por máquinas XLS.

- **Nivel 3**: Los datos deben estar disponibles en formato abierto no propietario por ejemplo CSV en lugar de XLS.
- **Nivel 4**: Corresponde a todos los niveles anteriores más, utilizar estándares abiertos de W3C (RDF y SPARQL) para identificar las cosas.
- **Nivel 5:** Vincular los datos enlazados con otras fuentes para agregar contexto a la información

1.5. Herramientas

En esta sección se realiza la investigación de las herramientas necesarias para el cumplimiento de los objetivos, teniendo en cuenta el proceso de recolección de datos y la transformación de datos haciendo referencia a frameworks y APIs que ayudan a agilizar los dos procesos.

1.5.1. Técnicas para procesar documentos HTML.

1.5.1.1. DOM.

Es una interfaz de programación para documentos HTML y XML la cual facilita una representación estructurada que define la manera en que los programas pueden acceder al contenido. “El DOM es una representación completamente orientada al objeto de la página web y puede ser modificado con un lenguaje de script como JavaScript” (Mozilla, 2016).

1.5.1.2. Jquery (Manipulation DOM)

Es una librería de Java Script muy utilizada en el desarrollo web la cual permite agregar interactividad en un sitio, dentro de sus secciones se encuentra la manipulación del DOM esta herramienta sirve para analizar código HTML, el cual se lo utiliza para realizar expresiones XPATH ya que proporciona un proceso de analizar etiquetas, atributos, y datos a media que se encuentran, cada expresión XPATH encontrada se almacena para ser utilizada dentro de la extracción de datos, además esta técnica ayuda a estructurar algoritmos capaces de identificar patrones en las páginas Web.

1.5.2. Herramientas para el proceso de extracción de datos.

1.5.2.1. XPATH.

“Es un lenguaje expresivo de selección de nodos para documentos XML, que puede abstraerse como árboles de datos” (Figueira, 2017). Además de ser un lenguaje que permite construir expresiones que recorren documentos HTML, su principal función es examinar la estructura XML para acceder a los datos, el mismo contiene un modelo de datos establecido el cual consta de raíz, elemento, atributo, texto.

1.5.2.2. Framework Scrapy

Es una de las técnicas más utilizadas para recolectar información de la Web a datos estructurados. Entre sus principales características se puede señalar (Scrapy, 2017):

- Rápida y poderosa: Se definen reglas de extracción.
- Extensible: Proporciona una configuración donde se puede generar nuevas funcionalidades sin tener que modificar el código fuente.
- Portable: Está escrito en Python y puede correr en Linux, Windows, Mac.

1.5.2.3. Selenium WebDriver

Es una herramienta capaz de automatizar un navegador dependiendo a las necesidades del usuario, con pocas líneas de código se puede realizar una automatización del navegador, entre las características más importantes están:

- Tiene un API simple y concisa.
- Impulsa el navegador de manera mucho más efectiva y carga todo el contenido dinámico dentro de un sitio web.
- Compatible con la mayoría de navegadores existentes y los diferentes sistemas operativos.

1.5.3. API LOV.

LOV (Linked Open Vocabularies) consta de 527 vocabularios, 20.000 clases y casi 30.000 propiedades hasta el 2017. LOV proporcionan un acceso remoto a través de un conjunto de servicios REST.

A través de la sección vocabularios se puede enlistar y autocompletar temimos de un vocabularios**,** en la Tabla 2 se muestra todas las funcionalidades que soporta LOV.

Tabla 2. Funciones LOV

Vocabularios		
Funciones	**REST**	**Descripción**
Lista de vocabularios	http://lov.okfn.org/dataset/lov/api/v2/vocabulary/list	Proporciona una lista de todos los vocabularios
Búsqueda de vocabularios	http://lov.okfn.org/dataset/lov/api/v2/vocabulary/search?q=time	Proporciona una lista de todos los vocabularios que contengan el termino ingresado
Autocompletado	http://lov.okfn.org/dataset/lov/api/v2/vocabulary/autocomplete?q=geo	Proporciona una lista de los vocabularios por medio de un prefijo

Información	http://lov.okfn.org/dataset/lov/api/v2/vocabulary/info?vocab=foaf	Proporciona información del vocabulario ingresado

Fuente: Autor
Elaboración: Autor

1.6. Trabajos relacionados

En esta sección se analizan las diferentes herramientas relacionadas con el trabajo de fin de titulación. Actualmente son escasas las herramientas que integren las funcionalidades de recolección de datos hasta la conversión RDF. A continuación se mencionan las herramientas con mayor similitud a este proyecto, separándolas en dos grupos: herramientas en línea que ofrecen el servicio de recolección de datos y herramientas que permiten generar RDF a partir de un CSV.

Las investigaciones y herramientas en línea analizadas en esta sección se obtuvieron de Google Scholar, IEEE en base a patrones de búsqueda o palabras clave como: “CSV to RDF, convert CSV TO RDF, convert automatically to RDF”

1.6.1. Herramientas en línea para recolección de datos

1.6.1.1. Import.io.

Es una herramienta en línea que ayuda a la recolección estructurada de los datos encontrados dentro de una página web. La herramienta realiza una extracción automática, con un algoritmo capaz de encontrar etiquetas repetidas dentro del HTML, esta herramienta es muy utilizada en todos los campos ya que se acopla a cualquier sitio web, como toda aplicación tiene una variedad de ventajas y limitaciones dentro del servicio que ofrece, en la Tabla 3 se muestran algunas de las características importantes y las limitaciones que tiene la aplicación.

Tabla 3. Características y limitaciones de LOV

Características	Limitaciones
Fácil manejo ya que cuenta con una interfaz muy interactiva con el usuario. Extrae información de forma automática Permite seleccionar el los áreas de recolección Exporta los datos en un formato CSV. Gratuita	No realiza un seguimiento de enlaces; es decir si existe enlaces dentro de la información no extrae la información de los enlaces. Solamente se puede realizar la paginación si se utiliza el GET API que ofrece import.io reduciendo el uso de la herramienta a un grupo de personas que tienen conocimientos en la informática.

Fuente: Autor
Elaboración: Autor

1.6.1.2. DataScraping.

Herramienta web que permite extraer información, mediante extensiones de Google Chrome. Cuenta con una aplicación para escritorio que permite seleccionar los ítems a través de una extensión de Google Chrome; además, ofrece realizar el proceso de recolección sin tener que aprender técnicas de extracción de datos, en la Tabla 4 se muestran las características y limitaciones que ofrece esta aplicación.

Tabla 4. Características y limitaciones de DataScraping

Características	Limitaciones
Extrae la información que el usuario selecciona. Permite seleccionar el HTML, texto o css de un elemento Exporta los datos en un formato CSV, JSON. Permite gestionar la data almacenada en la nube.	El usuario necesita tener conocimientos técnicos para utilizar la herramienta. No realiza un seguimiento de enlaces; es decir si existe enlaces dentro de la información no extrae la información de los enlaces. No permite realizar paginación; solo extrae información de la página principal.

Fuente: Autora
Elaboración: Autora

1.6.1.3. Extractly.

Es una herramienta en línea que da control total al usuario, y seleccionar los elementos que requiere extraer. Esta herramienta ofrece extraer los datos de cualquier sitio web, sin embargo algunas URLs ingresadas no pueden ser procesadas por la herramienta. En la Tabla 5 se muestran las características y limitaciones de esta herramienta

Tabla 5. Características y limitaciones de Extractly.

Características	Limitaciones
-Extrae información que el usuario selecciona. - Permite guardar un historial de las XPATH - Se puede descargar los datos en formato JSON	-La interfaz de usuario de la herramienta es muy técnica, es decir al momento de seleccionar ítems se genera código. -No realiza un seguimiento de enlaces. -No permite realizar paginación.

Fuente: Autora
Elaboración: Autora

1.6.1.4. Comparación de herramientas

En base a las herramientas investigadas dentro de la sección “Recolección de información” se han encontrado tres herramientas que ofrecen el servicio de recolección de datos. A continuación se realiza un análisis comparativo basado en una serie de indicadores que sirve para realizar un resumen tanto de las limitaciones y las características de las herramientas, los indicadores fueron creados a partir de las características y limitaciones que presentan las herramientas estudiadas.

Descripción de indicadores encontrados

- **Interfaz amigable:** Evalúa la usabilidad de la herramienta.
- **Renderizar contenido dinámico:** Evalúa fallos frente a diseño de páginas actuales.
- **Seguimiento de enlaces:** Evalúa el comportamiento de realizar crawler.
- **Paginación:** Evalúa criterios básicos de un crawler.
- **Exportación de datos:** Evalúa que permita realizar una exportación en formatos abiertos.

La Tabla 6 realiza la comparación de cada herramienta frente a los indicadores creados, dando como resultado una matriz que permite encaminar el trabajo de TT tomando en cuenta ciertos indicadores que hoy en día no ofrecen las herramientas de recolección de datos.

Tabla 6 Análisis de herramientas de recolección de datos vs indicadores encontrados

Herramienta	Interfaz amigable	Renderizar contenido JS	Seguimiento de enlaces	Paginación	Exporta a formatos abiertos
Import.io	**X**			**X**	**X**
DataScraping	**X**				**X**
Extractly	**X**				**X**
Solución TT	**X**	**X**	**X**	**X**	**X**

Fuente: Autora
Elaboración: Autora

1.6.2. Herramientas para la transformación de datos RDF

1.6.2.1. Mapeo semi-automatico de fuentes estructuradas de la Web Semántica

(Knoblock et al., 2012) presenta un proceso semiautomático que permite convertir fuentes estructuradas según el modelo definido por una ontología, según los autores el mapeo

automático muchas de las veces es ambiguo, para superar este problema ofrece una interfaz gráfica al usuario que permite refinar o modelar los modelos de forma interactiva, la idea principal de esta investigación es dar semántica al proceso de conversión.

Características

- Usa los principios de Linked Data
- Los usuarios pueden definir o reutilizar vocabularios

1.6.2.2. Marco de conversión CSV2RDF

(Ermilov, Bis, & Stadler, 2012) construyeron una aplicación para la transformación de datos tabulares encontrados en PublicData.eu a RDF. La generación a RDF requiere un esfuerzo adicional como diseño de vocabularios, reutilización y mapeo de vocabularios. Esta aplicación busca transformar los datos en formatos normalizados facilitando la descripción semántica, vinculación e integración de datos mediante un enfoque que es compartido por la máquina y el usuario. Para cumplir su propósito CSV2RDF, utiliza las siguientes herramientas:

- ***Sparqlify-CSV:*** Para la conversión a RDF.
- ***Lenguaje de mapeo (Sparqlify-ML):*** Proporciona medios para definir "vistas RDF" sobre una base de datos relacional o una fuente CSV.

Según el marco que sigue la herramienta, las columnas son usadas en el proceso de conversión como identificadores para las propiedades respectivas.

1.6.2.3. LevelUp CSV to RDF

Es una herramienta en línea que ayuda a la conversión de datos en una serializacion RDF/XML. Realiza la transformación de los datos tomando las cabeceras de un archivo CSV. Mediante el proceso de trasformación, los recursos son descritos mediante un proceso de enriquecimiento. Para transformar los datos se carga un archivo CSV y se Ingresan las configuraciones (prefijo, tipo de dato).

Entre sus principales características están:

- Proporciona una interfaz gráfica amigable para el usuario.
- Permite exportar datos a las diferentes serializaciones RDF.
- Da control al usuario para que establezca el prefijo y las configuraciones de transformación.
- El flujo de transformación es intuitivo.

1.6.2.4. Open Refine (Extensión RDF Refine)

Es una herramienta gratuita muy poderosa para trabajar con datos desordenados, además permite realizar limpieza y análisis de los datos. Mediante la extensión RDF se puede crear el modelado de datos RDF. Las características más importantes se muestran a continuación:

- ***Exportación RDF:*** Para la exportación de datos RDF se requiere hacer un proceso de definir URI, prefijos, clases, propiedades. Esta herramienta realiza sugerencias de las posibles clases y propiedades dependiendo a la data y así definir un esqueleto para convertir datos a RDF.
- ***Reconciliación de la data:*** Realiza la reconciliación de tres formas: basada en SPARQL, reconciliación basada en búsqueda de texto y reconciliación usando Sindice, si se encuentra dos o más recursos para un determinado dato se alerta al usuario para realizar una reconciliación manual.

En base al análisis realizado de los trabajos relacionados, se tiene una visión global de las herramientas que proporcionan servicio de crawler y la generación de RDF.

1.6.2.5. Comparación de herramientas de transformación de datos

En base a las herramientas investigadas dentro de la sección “Transformación de datos” se han encontrado varias herramientas que realizan la transformación de datos estructurados a RDF. En esta investigación se hace énfasis en la tecnología que utiliza cada herramienta como las características principales de cada una de ellas. Dentro del alcance establecido se requiere realizar una aplicación que permita transformar datos CSV o estructurados a serializaciones RDF tomando en cuenta el reuso de vocabularios, dando como indicadores de esta comparación los siguientes:

- **Interfaz amigable:** Evalúa la usabilidad de la herramienta.
- **Reuso de vocabularios:** Evalúa el uso de los vocabularios existentes dentro de la Web Semántica.
- **Exportaciones a las Serializaciones RDF:** Evalúa el comportamiento al convertir CSV a RDF.

Dentro de la Tabla 7 se realiza la comparación de cada herramienta frente a los indicadores creados dando como resultado una matriz que permite encaminar el trabajo de TT tomando en cuenta ciertos indicadores que hoy en día no ofrecen las herramientas de transformación de datos.

Tabla 7 Análisis de herramientas de transformación de datos vs indicadores encontrados

Herramienta	Interfaz amigable	Re-uso de vocabularios	Exportaciones a las serializaciones RDF
Mapeo semi-automatico de fuentes estructuradas de la Web Semántica	X		X
Marco de conversión CSV2RDF			X
Level Up CSV to RDF	X		X
Open Refine (Extensión de RDF Refine)	X	X	X
Trabajo de TT	X	X	X

Fuente: Autora
Elaboración: Autora

1.7. Comentarios finales

Al finalizar el capítulo 1 se tiene un dominio del tema asegurando que la investigación tenga bases estables para continuar con el desarrollo de la investigación, tomando en cuenta herramientas que ayudan a solventar el problema planteado, además se realizó la investigación de trabajos relacionados identificando indicadores faltantes dentro del medio, teniendo en cuenta que esta fase es crucial para definir el alcance del trabajo.

CAPITULO 2: PROPUESTA DE LA SOLUCIÓN

El propósito de este capítulo es presentar la problemática, y la solución tomando en cuenta la sección 1.5 donde se encuentra las herramientas relacionadas al trabajo de fin de titulacion dando como resultado un modelo acoplado a los requerimientos del usuario, este capítulo consta de 4 secciones:

2.1 Problemática: En este apartado se describe la problemática que existe actualmente, tomando en cuenta la sección de trabajos relacionados ubicados en la sección 1.5 donde se establecen indicadores de las limitaciones de las herramientas estudiadas.

2.2 Modelado de solución: Se define la solución, tomando en cuenta los problemas actuales. Además se realiza el flujo que tendrá la aplicación frente a la recolección de datos como en la transformación a serializaciones RDF.

2.3 Metodología de desarrollo: En este apartado se define una metodología para la gestión y desarrollo del proyecto, de tal manera que se cumplan con los requerimientos planteados.

2.4 Resultado del capítulo: Se realiza un breve resumen de los resultados obtenidos en esta sección, mencionando aspectos claves para continuar con la investigación.

2.1. Problemática

Se estableció dos partes para definir la problemática tomando en cuenta los trabajos investigados en el capítulo 1:

- **Sobrecarga y falta de estructura de la información**: Dentro de la investigación en el capítulo 1, muchos autores señalan que la heterogeneidad de las páginas web y el inadecuado uso de etiquetas HTML es uno de los grandes problemas de la Web actual, donde existen millones de recursos alojados en la Web que contienen información valiosa para el usuario. Para acceder a dicha información es necesario realizar peticiones HTTP. Además la información que se encuentra en formato no estructurado la cual está limitada a la toma de decisiones. El proceso de recolectar datos no estructurados lleva consigo el uso de herramientas informáticas que ayudan a extraer información de una página web, tomando en cuenta que hoy en día existen herramientas que facilitan a usuario la forma de recolectar datos, pero poseen limitaciones que son necesarias para tener una buena estructura de datos.
- **Datos legibles para las maquinas:** Uno de los desafíos más grandes es representar la información que pueda ser entendida por máquinas y personas, es por ello que nace la Web 3.0, la cual busca agregar semántica a los datos mediante ontologías (esquema de conocimiento). Además la Web actual posee estructuras heterogéneas y la terminología utilizada en cada sitio Web conlleva a un problema de sinonimia.

2.2. Modelo de solución

Se propone dividir el problema en dos grandes funcionalidades como: recolección de datos y trasformación a RDF. En la Figura 3 se ilustran tres etapas que se ha creído conveniente utilizar, cada etapa indica las herramientas necesarias para realizar la semi-automatización de la recolección de datos hasta la conversión a RDF.

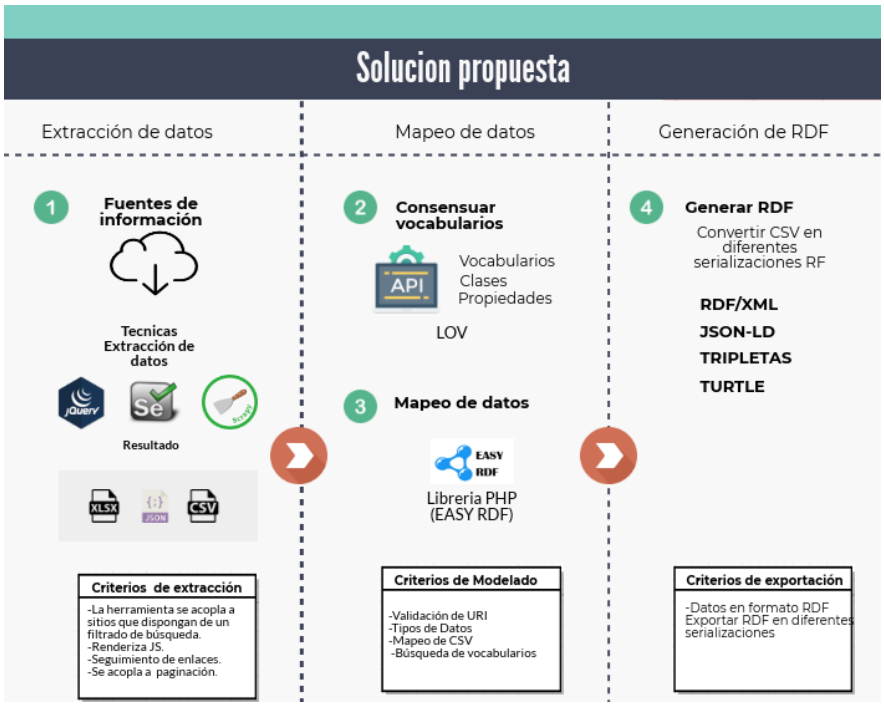


Figura 3. Solución propuesta
Fuente: Autora
Elaboración: Autora

A continuación, se detalla cada fase propuesta, donde se realiza un flujo del como la aplicación aborda los objetivos de cada etapa.

2.2.1. Recolección de datos

Dentro del presente trabajo este problema se lo mitiga con la participación del usuario ya que este proceso de recolección debe ser acoplado a cualquier sitio web, el usuario asume el rol más importante donde puede establecer parámetros o áreas que requiere recolectar de manera interactiva.

En esta etapa se establece el alcance de la recolección de datos, la Web posee estructuras heterogéneas en la representación de la información, a continuación se definen los definir criterios de diseño de la aplicación que tienen que tener las paginas web para realizar la recolección de manera exitosa:

- Páginas web que dispongan filtrado o búsqueda de información mediante un criterio.
- Para realizar un crawler exitoso y mostrar los ítems que pueden ser recolectados es necesario que dentro de cada enlace la información se encuentre en un formato semiestructurado.
- Dentro de la sintaxis HTML el atributo a recolectar debe de tener la clase de cada etiqueta permitiendo establecer expresiones XPATH fiables.

Para la recolección de la información se hace uso de múltiples técnicas y herramientas como: Scraping, Selenium y jquery, donde se ve conveniente fusionar estas técnicas para el cumplimiento de los objetivos planteados, las cuales facilitan la extracción de información que se encuentra alojada en la Web, desde este punto se crea una aplicación web que permita al usuario extraer información teniendo en cuenta criterios de selección como: datos relevantes, , uso de la información.

- **Flujo de la solución**

Uno de las principales características que se abarca en esta solución web es la interacción que tiene el usuario con la aplicación, por ello se utilizan tecnologías amigables para el usuario como es HTML jquery, JavaScript, las cuales permiten realizar el desarrollo enfocado a las interacciones de usuario. En la Figura 4, se muestra el flujo de la solución propuesta para la recolección de datos tomando en cuenta diferentes aspectos como:

- El uso inadecuado de etiquetas HTML en la representación del contenido no es estándar.
- La mayoría de páginas web cargan su contenido de manera dinámica lo cual dificulta la extracción de datos.
- La paginación dentro de cada sitio web es utilizado de diferentes formas, dificultando la localización de manera automática para realizar el seguimiento de enlaces.

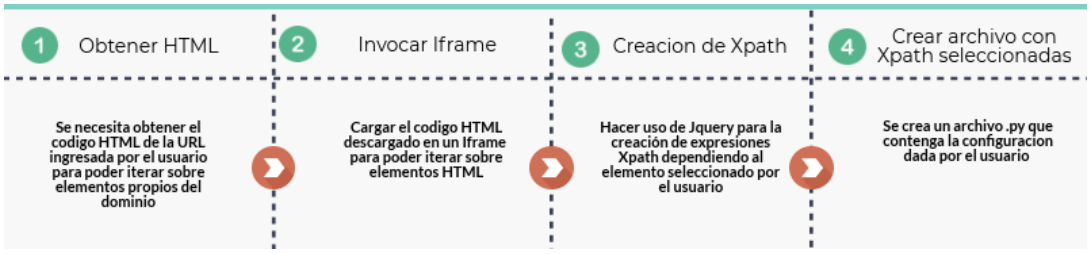


Figura 4. Solución para la recolección de datos
Fuente: Autora
Elaboración: Autora

- **Obtener HTML:** Desarrollar una manera óptima de obtener el código HTML con sus respectivas hojas de estilos e imágenes.
- **Desarrollar un script para iterar sobre elementos HTML:** Desarrollar una interfaz gráfica capaz de involucrar al usuario en la selección del contenido.
- **Desarrollar un script para formar XPATH (expresiones que recorren un XML):** Realizar un algoritmo capaz de formar expresiones XPath tomando en cuenta atributos como la clase.
- **Determinar un formato estándar para la recolección de datos:** La solución óptima de recolectar datos es seguir el esquema dado por Scrapy donde plantea:
    - Búsqueda de bloque contenedor
    - Selección de ítems
    - Selección de paginación si es necesario

2.2.2. Transformación de datos

Para el mapeo de los datos recolectados, el usuario posee el principal rol, en la cual se tiene varias actividades como: búsqueda de términos y mapeo de datos.

Para llegar a la etapa de buscar vocabularios se necesita clasificar los metadatos o determinar los nombres de las clases, una vez establecidos los parámetros se comienza a buscar vocabularios existentes que contengan términos acoplados a cada metadato, en el apartado siguiente se detalla el flujo del mapeo de los datos CSV a serializaciones RDF.

- ***Búsqueda de términos***

Esta actividad tiene como fin realizar búsquedas de vocabularios acoplados a las clases o subclases de los datos recolectados. En el presente trabajo se hace uso de un catálogo de vocabularios denominado LOV (Linked Open Vocabularies) el cual cuenta con un API que permite realizar consultas sobre vocabularios, clases y términos.

- ***Mapeo de datos***

Una vez seleccionados los vocabularios a utilizar, establecidos en la actividad anterior se necesita la intervención y conocimiento por parte del usuario para realizar un modelo óptimo de acuerdo a sus necesidades, el cual tendrá una interfaz amigable para que el usuario interactúe con la aplicación, de tal manera que tenga un área de trabajo para construir un modelo formal.

- **Flujo de a solución mapeo de datos**

Para realizar el mapeo de los datos es necesario tener como entrada un archivo estructurado CSV delimitado por comas, donde se especifique cada columna con su respectivo metadato, además es necesario que la columna 1 sea el recurso que se describe. En la Figura 5 se muestra el proceso del mapeo de los datos con cada término ontológico seleccionado por el usuario.

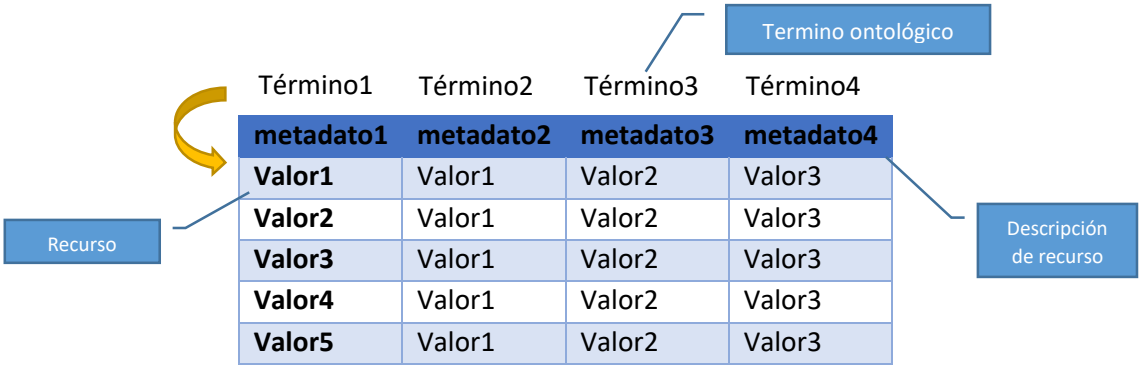


Figura 5. Solución mapeo de datos CSV
Fuente: Autora
Elaboración: Autora

- **Termino ontológico:** Se establece el reuso de términos desde el catálogo LOV o la creación de un término en específico teniendo en cuenta la URI del usuario.
- **Descripción de recurso**: Es el nombre del metadato que describe cada columna.
- **Recurso:** Término que se va a describir y que consta en la primera columna del CSV.

Para realizar el mapeo de los datos es necesario tener una estructura con los elementos descritos en la parte superior, dentro de la solución los datos de entrada (CSV) se trabajan por posiciones: la primera columna es el recurso que se va a describir, teniendo en cuenta los límites y las posiciones de cada elemento, a partir de cada grafo se incluye los atributos necesarios de cada recurso. En la Figura 6 se muestra el grafo que se crea por cada fila de la Figura 5, teniendo en cuenta el tipo de datos de cada atributo del CSV, en donde el tipo de dato que es soportado dentro de la aplicación son: cadena, recurso, entero, fecha considerando los estándares de representación dado por la W3C.

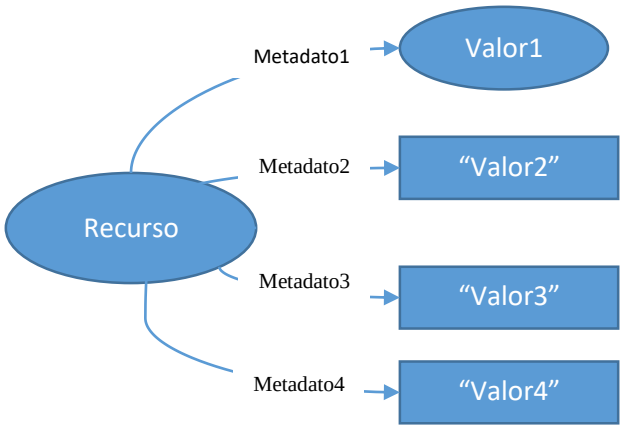


Figura 6. Grafo por cada fila recorrida
Fuente: Autora
Elaboración: Autora

2.3. Metodología de desarrollo

2.3.1. SCRUM

(Schwaber & Sutherland, 2013) describen todo el ciclo de uso de la metodología SCRUM, desde conceptos claves hasta el flujo de la metodología ágil, que permite gestionar un proyecto, por lo que responde con rapidez y da un buen resultado, Entre sus características más importantes se tiene:

- Adoptar una idea total de la realización del producto, en lugar de la planificación y ejecución completa del proyecto.
- Enfocarse más en las zonas de solapamiento, en lugar de realizar una tras otra en un ciclo de cascada.

Dentro SCRUM se realizan tres fases fundamentales, para su desarrollo. A continuación se detalla cada una de ellas

2.3.1.1. Planeación.

La fase de planeación consiste en analizar características propias del proyecto, utilizando historias de usuario las cuales describen los requerimientos delusuario, además en esta fase se establecen tiempos estimados de desarrollo tomando en cuenta dos aspectos importantes como: la prioridad y la complejidad de los entregables. Los resultados de esta fase son:

- **Historias de usuario y criterios de aceptación:** Se definen los requerimientos establecidos sobre cómo debe actuar el sistema mediante un evento.
- **ProductBacklog:** Es un documento que contiene descripciones genéricas de todas las funcionalidades y requerimientos deseables donde se realiza una priorización según sea su valor.
- **SprintBacklog:** Se describen de manera ordenada, los entregables del proyecto (Sprints), así mismo, por cada entregable se establece el tiempo que toma realizar cada entregable y el tiempo estimado en días.

2.3.1.2. Desarrollo de Sprint

En esta fase se desarrolla cada entregable cumpliendo criterios como tiempo, calidad y costos. Para el desarrollo de cada sprint se realiza un proceso de validación que se ajusta a cada proyecto:

2.3.1.3. Cierre.

Esta fase correspondiente a la fase final del proyecto implica realizar pruebas sobre todo el sistema para asegurar la calidad y la funcionalidad de todos los requerimientos establecidos en la fase de planeación.

2.4. Resultados del capítulo

Como resultado de este capítulo se ha diseñado la propuesta que cubre el proceso de extracción de datos y transformación de datos, teniendo en cuenta el alcance del TT. Se ha descrito el flujo de la aplicación y las diferentes herramientas y APIs que serán de ayuda para el cumplimiento de los objetivos. Además se presenta la metodología ágil Scrum para el desarrollo del proyecto ya que permite estimar tiempo y avances de manera periódica.

CAPITULO 3: DESARROLLO DE LA SOLUCIÓN

En el presente capítulo se describe el desarrollo de la solución al problema planteado tomando en cuenta el capítulo 2 (definición de la solución) y la metodología de desarrollo SCRUM. Este capítulo detalla las fases de la metodología investigada la cual abarca 3 fases:

3.1 Planeación: Se realiza tres entregables iniciales propuestos por la metodología: Historias de Usuario, ProductBacklog y SprintBacklog.

3.2 Desarrollo: Se hace referencia a la fase 2 de la metodología, donde se detalla cada entregable desarrollado hasta su culminación.

3.3 Resultados del capítulo Se realiza un breve resumen de los resultados obtenidos del capítulo, mencionando aspectos claves para finalizar la investigación.

3.1. Planeación

3.1.1. Historias de usuario.

Para la elaboración de las historias de usuario, se realizó reuniones periódicas con el director de tesis para definir el alcance que debe tener el proyecto, teniendo en cuenta el capítulo 1 de trabajos relacionados para más detalle revisar el Anexo 1 Historias de usuario.

3.1.2. Product Backlog

Una vez obtenidos los requerimientos es necesario definir cómo se va a desarrollar cada funcionalidad priorizando características del proyecto ver Anexo 2 ProductBacklog.

3.1.3. SpintBacklog

Se detallan las tareas por cada iteración así mismo se estima el tiempo de desarrollo en días de cada sprint ver Anexo 3 SprintBacklog.

3.2. Desarrollo

En esta sección se detalla el desarrollo de cada Sprint definido en el Anexo 3 detallando características importantes de cada entregable. Cada Sprint tiene un Alias asociado a su nombre es por eso se lo detallará de acuerdo al alias que corresponde.

3.2.1. Ingreso a la aplicación web (R001)

3.2.1.1. Diseño de la arquitectura.

Para el diseño de la aplicación web se construye una arquitectura acoplada a las funcionalidades del TT, la misma cubre aspectos generales de una arquitectura cliente servidor ya que es una arquitectura distribuida. Esta arquitectura tiene un flujo donde los clientes acceden a los servicios que proporciona el servidor a través de llamadas a procedimientos remotos usando el protocolo HTTP. En la Figura 7 se muestra la arquitectura de la aplicación web.

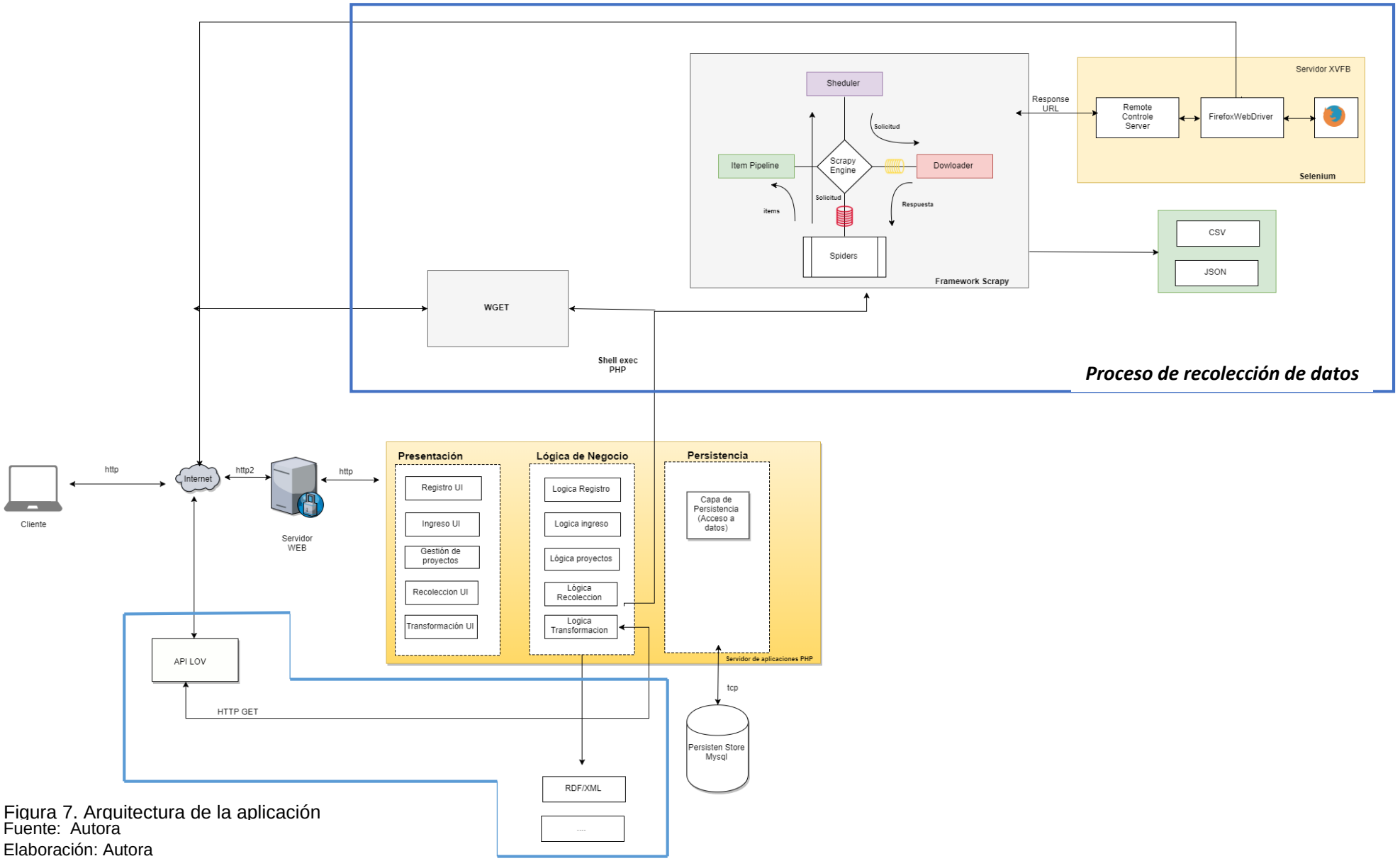


Figura 7. Arquitectura de la aplicación
Fuente: Autora
Elaboración: Autora

El flujo dentro de la arquitectura se realiza mediante peticiones a un servidor web el cual aloja las diferentes funcionalidades de la página web.

Servidor de aplicaciones PHP:

Se implementa en PHP. Posee una estructura de tres capas que permite trabajar por separado las mejoras y actualizaciones de los componentes. Además es una arquitectura apta para desarrollar una aplicación cliente/servidor. A continuación se detalla cada una de ellas.

- **Capa de presentación:** Se encarga de presentar la información al usuario y la gestión de sus interacciones. Abarca cinco componentes de interfaz de usuario como:
    - Registro e ingreso que es indispensable dentro de una aplicación web que preste un servicio,
    - Gestión de proyectos.
    - Interfaz de recolección de datos.
    - Interfaz de trasformación de datos.
- **Capa de lógica de negocio:** Se implementa la lógica de la aplicación, donde se encuentran un archivo específico por cada componente de presentación. En esta capa se requiere utilizar la Shell y el framework Scrapy para la recolección de datos, el mismo se lo integra haciendo una llamada a la función “exec”, que simula el comportamiento de una terminal, también es conveniente utilizar selenium que ayuda a renderizar páginas web que cargan el contenido dinámicamente.

    Módulo recolección de datos: Dentro del módulo recolección de datos es necesario integrar tecnologías capaces de trabajar en conjunto para extraer información de páginas web. A continuación se hace referencia al uso de las tecnologías utilizadas en este módulo:

    - **Wget:** Se lo utiliza para bajar páginas web.
    - **Framework scrapy:** Se hace uso de spiders, ítems, download y la ingeniería propia de scrapy para recolectar datos estructurados de cualquier página web. Al igual que toda herramienta cuenta con limitaciones y una de ellas es renderizar páginas dinámicas. Por esta razón se lo acopla con Selenium que ayuda a renderizar páginas teniendo como resultado datos estructurados.
    - **Selenium WebDriver dentro de un servidor XVFB:** Ejecuta un controlador que hace posible que el navegador se maneje automáticamente en memoria a través de un webdriver.

Módulo transformación de datos: Dentro de este módulo se ve necesario utilizar APIs que den un servicio de búsqueda de vocabularios ontológicos que existen dentro de la Web. Se hace uso del API de LOV mediante llamada HTTP GET para obtener resultados, además de utilizar EasyRDF que es una librería PHP para generar serializaciones RDF.

- **Capa de persistencia:** Está directamente relacionada con la capa de lógica de negocio para realizar peticiones a la base de datos.

3.2.1.2. Diseño de base de datos

Para el almacenamiento de los datos efectuados por parte del usuario se crea una base de datos la cual consta de 5 tablas como se muestra en la Figura 8.

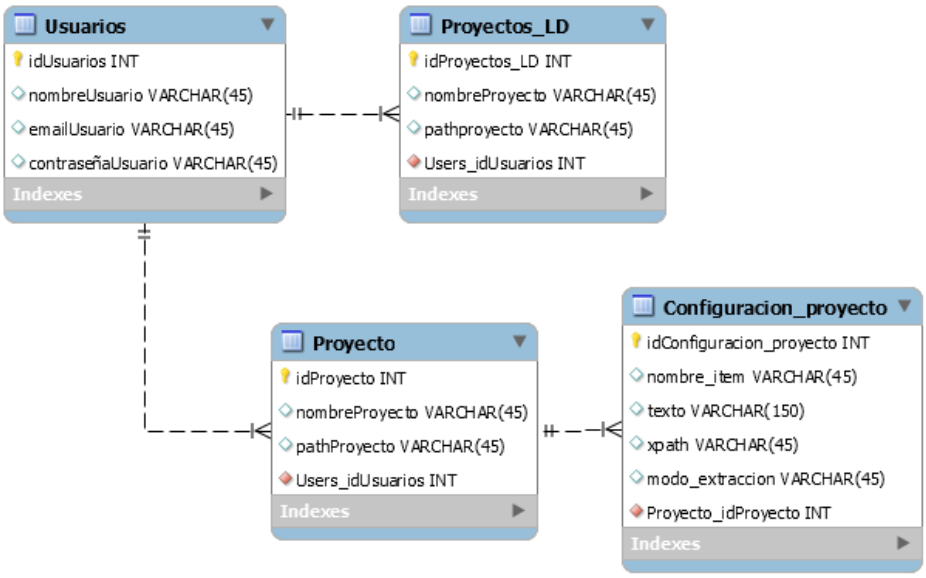


Figura 8. Diseño de base de datos
Fuente: Autor
Elaboración: Autor

Descripción de tablas

- **Tabla Usuarios:** Almacena información de los usuarios que se registran en la aplicación. Como atributos principales cuanta con una clave principal denominada “id Usuarios”, además, cuenta con atributos específicos que describen al usuario como nombre, email y contraseña.
- **Tabla Proyecto:** Contiene información general de los proyectos de recolección de datos generados por el usuario, el campo pathProyecto almacena la ruta del proyecto por cada usuario.
- **Tabla Configuración Proyecto:** Contiene información generada por el usuario para la recolección de datos como: nombre del ítem, texto, xpath y método de extracción,

la cual sirve para tener un historial de los cambios generados por el usuario en cada proyecto.

- **Tabla Proyectos_LD:** Almacena información de la localización del archivo RDF generado por el usuario.

En el Anexo 4 Diccionario de base de datos, se encuentra el diccionario de datos donde se establece una lista organizada de los datos que pertenecen a la aplicación.

3.2.1.3. Diseño del flujo de la aplicación

Dentro de la solución web es conveniente realizar la gestión de usuarios y de proyectos, es por eso se crea una aplicación amigable para el usuario. En la Figura 9 se muestra el flujo de la aplicación propuesta dentro de la solución.

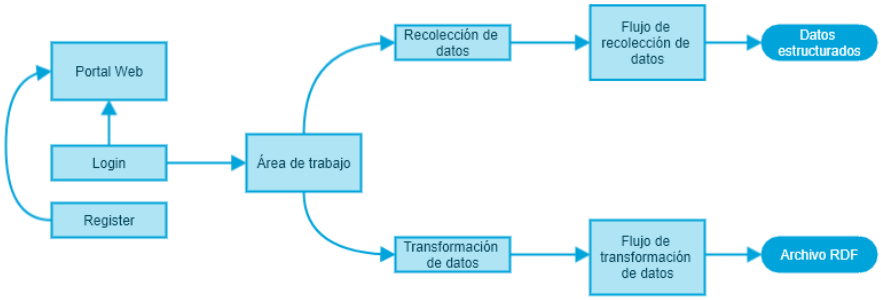


Figura 9. Diseño del fluio de la solución
Fuente: Autor
Elaboración: Autor

Descripción de Flujo:

- La primera etapa es el registro del usuario en el portal web donde es necesario obtener una credencial única por cada usuario.
- Una vez registrado, el usuario puede acceder a las funcionalidades del portal.
- Dentro de la aplicación el usuario puede realizar la recolección de los datos de un sitio web o transformar los datos estructurados a datos semánticos.
- En el apartado “Recolección de datos” se describe el flujo de instrucciones que el usuario debe tener en cuenta para que la extracción sea exitosa. Como resultado final de este componente se obtienen los datos estructurados en formatos CSV.
- En el apartado “Transformación de datos” es necesario tener como entrada un archivo estructurado CSV delimitado por comas para realizar el proceso de búsqueda de ontologías acoplada a la data dando como resultado un archivo RDF.

3.2.1.4. Desarrollo de la interfaz web

Desde este apartado se inicia la descripción del prototipo de la aplicación web donde se utiliza tecnologías como HTML, CSS, BOOTSTRAP, JS y JQUERY las cuales ayudan a tener un

contenido agradable para el usuario. A continuación se muestra la creación de las interfaces de los principales modulos de la aplicación.

- **Modulo principal**

  En la Figura 10 se muestra el portal principal que contiene secciones que engloba las funcionalidades del proyecto, ofreciendo al usuario una interfaz sencilla pero intuitiva de lo que realiza la aplicación, dentro de esta interfaz se toma en cuenta aspectos generales del alcance del proyecto.

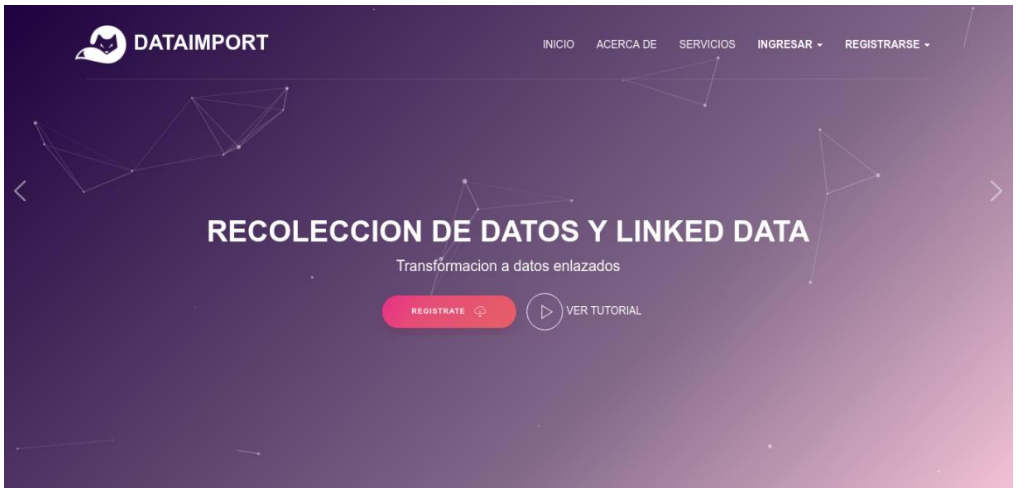


Figura 10. Diseño de interfaz principal
Fuente: Autor
Elaboración: Autor

- **Modulo Ingreso y registro**

En la Figura 11 se muestran las interfaces de registro e ingreso para la aplicación, en donde registro se toma en cuenta un identificador único por cada usuario, así mismo para la sección de ingreso tendrá que acceder por medio del usuario y clave registrada.

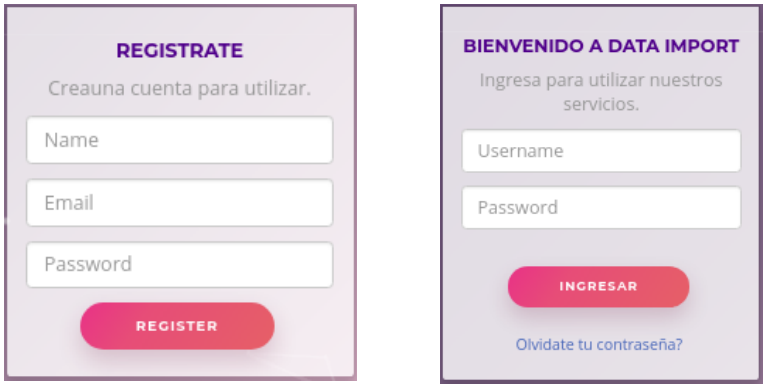


Figura 11. Diseño de interfaz eegistro e ingreso
Fuente: Autor
Elaboración: Autor

- **Modulo recoleccion de datos (scrapy-crawler)**

En esta sección se desarrolla la interfaz web en donde el usuario podrá extraer la información de manera iterativa . En la Figura 12 se muestra la maquetación de este componente donde se definen tres secciones importantes: sección instrucciones se detalla el flujo a seguir para obtener una recolección exitosa, sección Iframe se recupera el .html de la página web ingresada incluyendo los estilos css para la selección del contenido HTML, en la sección 3 se encuentra una tabla de toda la información que el usuario ingrese

Figura 12. Diseño de interfaz recoleccion de datos (Scrapy-Crawler)
Fuente: Autor
Elaboración: Autor

- **Desarrollo de interfaz resultado (recoleccion de datos)**

En la Figura 13 se ilustra la interfaz web para mostrar el contenido recolectado, el cual se hace uso de tablas para proyectar la información, teniendo en cuanta que se puede realizar filtros de búsqueda y exportaciones a diferentes formatos.

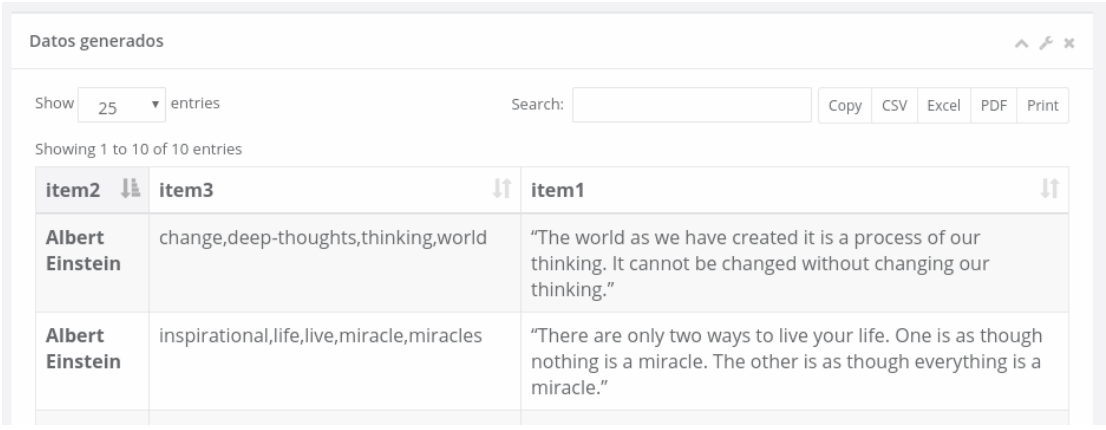


Figura 13. Diseño de interfaz presentacion de resultados

Fuente: Autor
Elaboración: Autor

- **Diseño de interfaz Transformación de datos**

  En la Figura 14 se muestra la interfaz web para el proceso de transformación de datos la cual está compuesta por cuatro secciones, cada una de ellas sigue un flujo para realizar la transformación de datos, para el uso de este modeulo es necesario tener conocimiento de la estructura que contiene las serializaciones RDF para poder realizar un mapeo de los datos de acuerdo a los metadatos de el CSV ingresado, asi mismo del tipo de datos que requiere cada valor. .

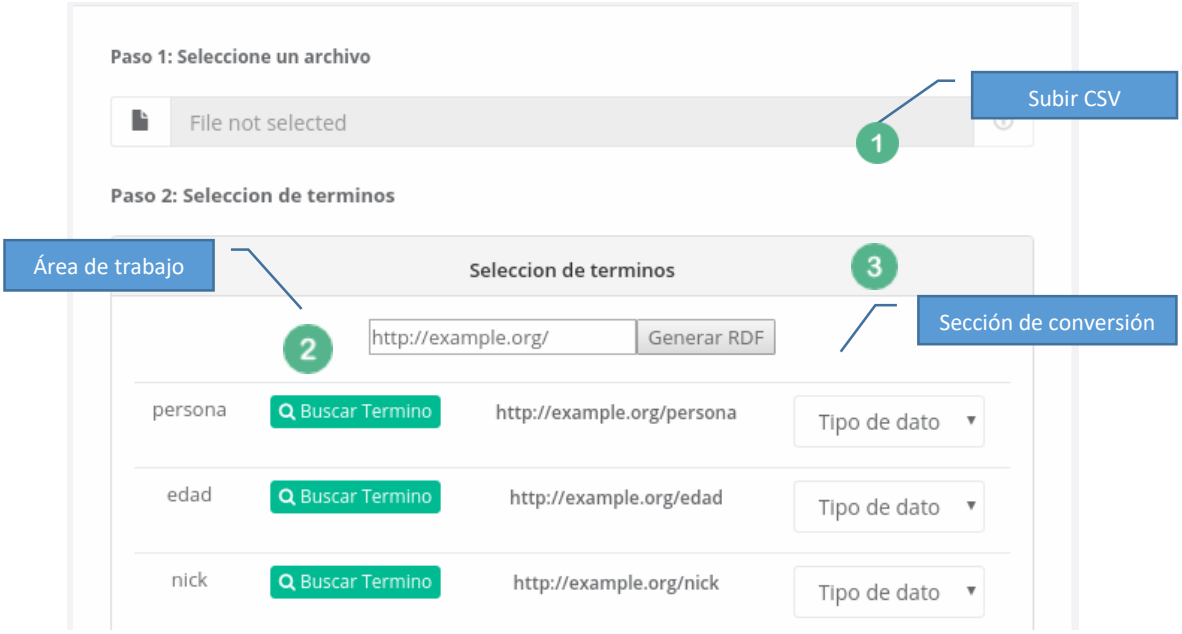


Figura 14. Diseño de interfaz transformación de datos
Fuente: Autor
Elaboración: Autor

- **Diseño de interfaz resultado de transformación de datos**

  En la Figura 15 se muestra la salida del mapeo de los datos. Se compone de una serie de botones en la cual el usuario pueda realizar la transformación a una socialización RDF, además de contar con la opción de descarga de cualquier tipo de serializaciónes.

Figura 15. Diseño de interfaz resultado de transformación de datos
Fuente: Autor
Elaboración: Autor

En el Anexo 5 titulado Desarrollo de interfaz web, se detalla cada una de las interfaces mostrando el flujo desde el portal principal de la aplicación, hasta las funciones primordiales del mismo

3.2.2. Modulo principal (I001)

3.2.2.1. Desarrollo del módulo registro.

Dentro del servidor se tiene una carpeta denominada **"usuarios_dataimport"** para el almacenamiento de archivos generados por el usuario; cada usuario que se registra en la aplicación se reserva un espacio de trabajo, con el identificador único (email), dentro de esta carpeta el usuario puede almacenar n proyectos de recolección de datos. En la Figura 16 se muestra el fragmento de código que genera cada registro de usuario, puntualizando dos aspectos importantes:

- Almacenamiento en base de datos
- Proceso de gestión de directorios

```
1  //función insert en BDD
2  function registro($nombre,$email,$clave){
3          include("../manejar_conexion.php");
4          $res=$miconexion->consulta("INSERT into Users
5          (`nombreUsuario`,`emailUsuario`,`contrasenaUsuario`)
6           VALUES('$nombre','$email','$clave')");
7      }
8  //Creacion de carpeta por cada usuario (servidor)
9      $path="$url_doc/$email";
10     $old = umask(0);
11     mkdir($path);
12     umask($old);
```

Figura 16. Código para realizar la sección registro

Fuente: Autor
Elaboración: Autor

3.2.2.2. Desarrollo módulo login

Se realiza una consulta directamente a la base de datos para comprobar credenciales ingresados por el usuario. En la Figura 17 se muestra el código que permite realizar el logueo de un usuario, así mismo las variables de sesión que se habilitan por cada usuario.

```
1  //Crea variables de sesión
2      if ($lista[0]!=NULL){
3          session_start();
4          $_SESSION['id']=$lista[0];
5          $_SESSION['nombreUsuario']=$lista[1];
6          $_SESSION['emailUsuario']=$lista[2];
7      }
```

Figura 17. Código para módulo de ingreso

Fuente: Autor
Elaboración: Autor

3.2.3. Módulo recolección de datos (E001)

3.2.3.1. Desarrollo de la gestión URL.

Para las configuraciones necesarias se realiza la gestión de la URL ingresada por el usuario en este apartado se hace uso de wget ya que es una herramienta libre que sirve para descargar contenidos desde servidores web. En la Figura 18 se muestra el código fuente para ejecutar WGET.

```
#Bajar paginas web con wget(imagenes,estilos)
$cmd="wget  --no-directories \
      --recursive \
      --page-requisites \
      --html-extension \
      --convert-links\
      --restrict-file-names=windows \
      --domains hola.org --no-parent\
      - P '$proyecto/scrapy/scrapy/' \
      --user-agent='Mozilla/5.0 \
        (Macintosh; Intel Mac OS X 10.9; rv:32.0) \
        Gecko/20100101 Firefox/32.0' "." "'.$url.'"';
```

Figura 18. Comando WGET para bajar páginas web

Fuente: Autor
Elaboración: Autor

En la Tabla 8 se realiza una descripción del comando wget.

Tabla 8. Descripción de comando WGET

Comando	Descripción
--no-directories	No crea jerarquía de directorios.
--recursive	Descarga recursivamente los archivos.
--page-requisitas	Descarga archivos necesarios para que la página funcione correctamente.
--html-extension	Retorna el archivo como un .HTML
--convert-links	Convierte enlaces locales en hipervínculos visibles.
--user-agent	Envía encabezados a las solicitudes HTTP.

Fuente: Autor
Elaboración: Autor

La solución propuesta al usar WGET está limitada a la descarga de páginas estáticas es decir, no cubre el contenido dinámico. Por esta razón se realizó una solución adicional capaz de dar al usuario una sección en la que pueda cargar un archivo HTML.

Dentro de la gestión URL es necesario editar el archivo .HTML con el fin de agregar estilos para acceder al contenido HTML, en la Figura 19 se muestra el fragmento de código para insertar css dentro del archivo descargado.

```
//Insertar link en archivo .html
 $path_to_file = "$proyecto/scrapy/scrapy/scrapy.html";
 $file_contents = file_get_contents($path_to_file);
 $file_contents = str_replace("</head>",$estilos,$file_contents);
 file_put_contents($path_to_file,$file_contents);
```

Figura 19. Fragmento de código para agregar estilos
Fuente: Autor
Elaboración: Autor

3.2.3.2. Desarrollo de XPATH

Para el manejo interactivo se usaron métodos propios de Jquery en donde a través de eventos como ***.mouseover (), .mouseout (), .click ()*** se logró tener una interfaz agradable para el usuario. En la Figura 20 se muestra el fragmento de código que ayuda al recorrido del documento HTML.

```
$(inframeDoc).mouseover(function (event) {
  class_valido[0]=$(event.target).attr('class');
  if(class_valido!=""){
    longi=class_valido[0].length;
    var ps=class_valido[0].slice(-1)
  }
  if(ps==" "){
    $(event.target).addClass('clase_espacio');
  }
  $(event.target).addClass('outline-element');
  }).mouseout(function (event) {
     $(event.target).removeClass('outline-element');
  }).click(function (event) {
      //Proceso Xpath
   });
```

Figura 20. Código para recorrer DOM
Fuente: Autor
Elaboración: Autor

Para la creación de expresiones Xpath se hace uso de métodos propios de Jquery que permiten la manipulación del DOM; este proceso inicia en la iteración del usuario al seleccionar un elemento del iframe. En la Tabla 9 se muestran los métodos y sus características de la implementación.

Tabla 9. Métodos JQuery para recorrer el DOM

Método	Uso
.nodeName	Recupera el nombre del nodo seleccionado por el usuario.
.attr('class')	Recupera el nombre de la clase del nodo seleccionado.
.parent()	Recupera el padre del elemento seleccionado para tener una expresión xpath más precisa.
.children()	Recupera los nodos hijos del elemento seleccionado para recorrer etiquetas HTML hasta llegar al texto.
.text()	Recupera el texto de la etiqueta HTML si es vacío el script recorre los demás nodos hijos hasta encontrar texto

Fuente: Autor
Elaboración: Autor

En el Anexo 7 Script para desarrollar XPATH se detallan los métodos generados dentro del script.

3.2.3.3. Desarrollo de algoritmo de recolección de datos (SCRAPY)

Para la elaboración del algoritmo de recolección de datos se determinó una estructura general para cualquier proyecto donde se tiene que considerar que las expresiones Xpath se deben generar de manera correcta. En el desarrollo de esta sección se determina el uso de la tecnología Selenium que ayuda en la automatización de las páginas dinámicas y el manejo de la paginación construyendo una archivo .py que en donde se establecen cuatro secciones.

- **Librerías e ítems**

Se establecen librerías para integrar scrapy con selenium permitiendo la fusión de las dos poderosas herramientas. Además contiene una clase permitiendo definir items para recolectar datos. En la Figura 21 se ilustra las librerías que contiene cada archivo .py

```
1  //Librerias e items
2  # coding=utf-8
3  import scrapy
4  from selenium import webdriver
5  from selenium.webdriver.common.keys import Keys
6  import time
7  from selenium.common.exceptions import NoSuchElementException
8  from pyvirtualdisplay import Display
9  display = Display(visible=0, size=(800, 600))
10 display.start()
11 class product_spiderItem(scrapy.Item):
12     item1=scrapy.Field()
13     ...
14     pass
```

Figura 21. Librerías e ítems para integrar Scrapy y selenium
Fuente: Autora
Elaboración: Autora

- **Configuración general**

Contiene variables necesarias como la URL, paginación y número de páginas y los xpaths a recolectar, además de estas variables contiene funciones selenium que ayudan a realizar peticiones para renderizar la URL ingresada tomando en cuenta tiempos de espera para realizar una scrapeado exitoso. Independientemente de cómo carga el contenido, la página Selenium se encarga de esperar 4 segundos para realizar un scroll dentro del webdriver ingresado. En la Figura 22 se muestra el fragmento de código para tener las configuraciones necesarias para recolectar datos.

```
//Clase spider
class ProductSpider(scrapy.Spider):
    name = "product_spider"
    start_urls = ["http://quotes.toscrape.com/"]
    COUNT_MAX = 3
    count = 0
    def __init__(self):
        self.driver = webdriver.Firefox()
    #Integracion Scrapy y Selenium
    def parse(self, response):
        self.driver.get(response.url)
        paginacion="Next →"
        page_number = 1+1
        #Proceso recoleccion
        while True:
            sel = scrapy.Selector(text=self.driver.page_source)
            for iteracion in sel.xpath('//div[@class="quote"]'):
                item = product_spiderItem()
                item['item1']=iteracion.xpath().extract()
                item['item2']=iteracion.xpath().extract()
                item['item3']=iteracion.xpath().extract()
                yield item
```

Figura 22. Configuraciones generales para recolectar datos

Fuente: Autor
Elaboración: Autor

- **Paginación**

Contiene métodos selenium que ayudan a rastrear la paginación ***"self.driver.find_element_by_link_text()"*** permitiendo realizar la busqueda de paginación por el texto del link, en la Figura 23 se muestra el fragmento de código que permite realizar la paginación el mismo tiene la capacidad de extraer links a partir del texto buscado. Se hace uso de funciones que permiten manejar el navegador haciendo clics. Cuando la condición sea falsa se realiza un corte dentro del Selenium web driver para dejar de ejecutar este proceso

Para el control de memoria y recursos dentro del servidor se agrega el cierre de los componentes utilizados: webdriver, display xvfb por cada ejecución.

```
//Manejo de paginación
self.count = self.count + 1
link = self.driver.find_element_by_link_text(str(paginacion))
try:
   if(self.count < self.COUNT_MAX):
      link.click()
      print self.driver.current_url
      page_number += 1
   else:
      break
except:
   break
def closed(self, reason):
display.stop()
self.driver.quit()
self.driver.close()
```

Figura 23. Configuraciones de paginación
Fuente: Autor
Elaboración: Autor

3.2.3.4. Desarrollo del algoritmo recolección de datos (CRAWLER)

A diferencia del algoritmo anterior cambia la lógica de configuración del proyecto ya que se realiza una solución obteniendo los links y los almacena dentro de un arreglo para recorrerlos posteriormente. Esta solución es óptima por lo que en un principio se definien las reglas del scrapeado. En la Figura 24 se muestra el fragmento de código necesario para realizar el crawler

```
//Solución Crawler
while True:
   #Almacenamiento de links
   linklist = []
   for link in self.driver.find_elements_by_xpath():
       linklist.append(link.get_attribute('href'))
   #Recorriendo los links y extrayendo informacion
   for link in linklist:
       self.driver.get(link)
       print self.driver.current_url
       sel = scrapy.Selector(text=self.driver.page_source)
       item = product_spiderItem()
       item['item1']=sel.xpath().extract()
       item['intem2']=sel.xpath().extract()
       yield item
       self.driver.back()
```

Figura 24. Configuraciones generales (CRAWLER)
Fuente: Autor
Elaboración: Autor

3.2.3.5. Implementación del algoritmo (Scrapy y Crawler).

Se hace uso del comando ***"shell_exec"*** para correr el archivo.py generado por el usuario dentro de las configuraciones del servidor es necesario tener instalado herramientas

necesarias para ejecutar el .py. En la Tabla 10 se muestra cada una de las herramientas con su respectiva versión estable.

Tabla 10. Herramientas utilizadas para la recolección de datos

Herramienta	Versión
Python	2.7
Selenium	3.8.1
PyvirtualDisplay	2.7

Fuente: Autor
Elaboración: Autor

Dentro de la carpeta de cada usuario se crea un archivo denominado “codigo1.py” el cual almacena las configuraciones necesarias para la recolección de datos. En la Figura 25 se muestra la línea de comando para ejecutar scrapy. Además se tomó en cuenta los archivos “log” que generan estos lenguajes dando como opción válida eliminar los logs después de la recolección de datos.

```
1 //Ejecución desde PHP
2 exec("cd $rf
3        && /var/www/env/bin/scrapy
4        runspider codigo1.py -o datos.csv 2>&1");
5  exec("rm -rf  $rf/scrapy/");
6  unlink("$rf/codigo1.py");
7  unlink("$rf/codigo1.pyc");
8  unlink("$rf/geckodriver.log");
```

Figura 25. Implementación de algoritmos
Fuente: Autor
Elaboración: Autor

3.2.4. Módulo de transformación de datos (V001)

3.2.4.1. Consumir vocabularios LOV.

Para consumir vocabularios LOV se realizan peticiones “GET” a través de Ajax. En la Figura 26 se muestra en fragmento de código para utilizar el servicio REST.

```
1 //Consumir LOV Vocabularios
2  $.ajax({
3  type: "GET",
4  url: "http://lov.okfn.org/dataset/lov/api/v2
5    /term/search?q="+singleValues+"&page_size=100",
6  success: function (result) {
7     #Proceso de seleccion
8
9  });
```

Figura 26. Consumir vocabularios LOV

Fuente: Autor
Elaboración: Autor

Se envía dos parámetros utilizados: el número de páginas de resultados y el valor a buscar tomando en cuenta la sintaxis válida

3.2.4.2. Desarrollo de mecanismo para generar RDF

Para generar RDF a partir de un archivo CSV es necesario realizar un mapeo junto con librerías capaces de poder crear grafos a partir de entradas. En esta sección se desarrolló una secuencia de instrucciones que sirve para construir serializaciones RD. Dentro del servidor PHP se encuentra un archivo denominado “rdf.php” que permite realizar la construcción de diferentes serializaciones. En la Tabla 11 se menciona los métodos utilizados dados por EasyRdf.

Tabla 11. Métodos utilizados de EasyRDF

Método	Descripción
public object resource($uri)	Crea un recurso para ser almacenado dentro de un grafo, se lo utiliza para posterior realizar una descripción de cada recurso dado por el CSV.
public integer set (string $resource, string $property, mixed $value)	Establece valores literales, fechas y número tomando en cuenta las representaciones establecidas para generar RDF.
EasyRdf_Literal_Date($valor)	Proporciona el formato necesario para representar un valor tipo fecha.
EasyRdf_Literal_Integer($valor)	Da el formato necesario para representar un valor tipo entero.

Fuente: Autor.

Elaboracion: Autor

En el Anexo 8 Archivo generador RDF se describe la estructura y los métodos que se utilizaron para la conversión de datos.

3.2.5. Manuales del aplicativo

Dentro de la documentación del aplicativo se tiene en consideración manuales de usuario.

3.2.5.1. Manual de usuario

El objetivo del manual es brindar al usuario asistencia para usar el aplicativo el cual contiene un flujo desde el ingreso de la aplicación hasta las funcionalidades primordiales. En el Anexo 10 Manual de usuario se especifica el documento.

3.2.5.2. Manual de programador

Se describe el uso de la tecnología y los archivos de configuración necesarios para el correcto funcionamiento de la aplicación; además, se describen aspectos claves de los módulos principales de la aplicación. En el Anexo 11, Manual de programador se especifica los módulos desarrollados.

3.3. Resultados del capítulo

En este capítulo se presentó el desarrollo de la solución tomando en cuenta la metodología Scrum. Se tiene la aplicación funcionando bajo los requerimientos establecidos por el usuario, además se realizó el acoplamiento de nuevos requerimientos dentro del aplicativo, adicionalmente se implementó la aplicación en Digitalocean, el proceso de la implementacion se lo describe en el siguiente capitulo mencionando la discponibilidad del mismo.

CAPITULO 4: PRUEBAS Y VALIDACIÓN

En este capítulo se explican diferentes pruebas que fueron diseñadas para verificar el funcionamiento y calidad de la aplicación. Estas pruebas son utilizadas en cualquier tipo de desarrollo informático utilizando herramientas para realizar la automatización de las mismas, bajo estándares establecidos. Este capítulo está compuesto por seis secciones:

En la **sección 4.1** Pruebas unitarias, describe una prueba por cada sprint detallado en la sección anterior.

En la **sección 4.2** Pruebas de caja negra se comprueba que toda la aplicación trabaje bien en conjunto ingresando datos a la aplicación.

En la **sección 4.3** Pruebas de rendimiento consisten en determinar la velocidad en ejecutar una tarea, en donde se mide los tiempos de respuesta de los dos módulos principales.

En la **sección 4.4** Análisis de código, analiza el código estático a través de Sonarqube para generar métricas de código.

En la **sección 4.5** Despliegue de la aplicación web, describe los pasos a seguir para implementar la aplicación web dentro de DigitalOcean.

En la **sección 4.6** Comentarios finales, se presenta un resumen de lo realizado en el capítulo enfatizando las secciones más importantes.

4.1. Pruebas unitarias

Se comprueba el funcionamiento de cada componente de la aplicación, tomando en cuenta la metodología utilizada, en la fase de desarrollo de cada sprint se realizó las pruebas independientes utilizando PHPUnit con el objetivo de aislar componentes y detectar errores en el código de manera individual. En la Tabla 12 se realiza un resumen de las pruebas realizadas a cada componente, detallando el nombre del sprint, las entradas para el funcionamiento y las salidas, y el estado de aceptación de cada prueba.

Tabla 12. Pruebas unitarias

	Nombre	**Entradas**	**Salidas**	**Validación**	
Sprint Registro e ingreso	Registro a la aplicación	Método		Registro en la base de datos y creación de directorio único	Validado
		Registro			
		Parámetros			
		nombre	Anita		
		email	ana@gmail.com		
		clave	ana1995		
		Método			Validado

	Ingreso a la aplicación	Ingresar		Consulta a la base de datos	
		Parámetros			
		email	ana@gmail.com		
		clave	ana1995		
Sprint Recolección de datos Scrapy	Gestión URL Proceso automático	Método		Accede al directorio del usuario y descarga la página web	Validado
		gestio_url			
		Parámetros			
		email	ana@gmail.com		
		name_project	proyecto1		
		URL	http:quotes.scrape.com		
	Gestión URL Proceso manual	Método		Accede al directorio del usuario y descomprime archivos enviados por el usuario	
		gestion_url_manual			
		Parámetros			
		email	ana@gmail.com		
		URL	http:quotes.scrape.com		En espera
		archivo .zip	archivo .zip		
	Gestión archivo .py	Método		Accede al directorio del usuario y crea el archivo .py	Validado
		escribir archivo			
		Parámetros			
		ruta	../email/proyecto/		
		código	"xpath insertadas por el usuario"		
	Recolección de datos	Método		Accede a la carpeta del usuario y ejecuta el archivo codigo1.py (el proceso de recolección se lo verifico manualmente)	Validado
		Recolección			
		Parámetros			
		ruta	../email/proyecto/codigo1.py		
Consumir vocabularios LOV	Mapeo de archivo CSV	Método		Extrae las cabeceras del archivo CSV	Validado
		Mapeo CSV			
		Parámetros			
		archivo CSV	archivo.csv		
	Generación de socializaciones	Método		Accede a la carpeta del usuario y ejecuta el archivo codigo1.py (el proceso de recolección se lo verifico manualmente)	Validado
		Generar RDF			
		Parámetros			
		vocabulario	foaf:name		
		fila CSV	Ana		

Fuente: Autor
Elaboración: Autor

4.2. Pruebas de caja negra

En este apartado se evalúa el comportamiento de la aplicación para asegurar el cumplimiento de los requerimientos planteados en la fase inicial.

- **Pruebas recolección de datos**

Dentro de estas pruebas se toma en consideración páginas web que se acoplan con los requerimientos necesarios para realizar una recolección de datos exitosa; las mismas se acoplan a las restricciones de la aplicación web, donde se debe de tener un filtrado de información. En la Tabla 13 se muestra el número de pruebas y el proceso que se acopló la página web, el proceso varia si una página web es compatible con las restricciones en la página web.

Tabla 13. Datos de entrada a la aplicación

	#	Pagina	Dominio	Ítems	Proceso	Estado
Recolección de datos	1	http://quotes.toscrape.com/	Quotes	4	Scrapy/Crawler	OK
	2	http://ieeexplore.ieee.org	IEEE	5	Scrapy	OK
	3	https://www.booking.com	BOOKING	6	Scrapy/Crawler	OK
	4	http://www.imdb.com/	IMBD	7	Scrapy/Crawler	OK
	5	https://www.hoteles.com/	Hoteles	6	Scrapy/Crawler	OK
	6	https://www.ebay.com/	Ebay	5	Scrapy/Crawler	OK
	7	https://scholar.google.com.ec	GoogleScholar	4	Scrapy	OK
	8	http://cnnespanol.cnn.com	CNN	5	Scrapy	OK
	9	https://www.peru.travel	PeruTravel	7	Scrapy/Crawler	OK
	10	https://mercadolibre.com.ec	Mercado Libre	4	Scrapy/Crawler	OK

Fuente: Autor
Elaboración: Autor

Prueba página Booking

En la Tabla 14 se detalla el resumen de la URL ingresada que pertenece al dominio de booking. En la Figura 27 se ilustra el resultado de la recolección de datos de la prueba realizada a la página booking.com sitio que muestra información de los hoteles encontrados bajo un criterio de búsqueda.

Tabla 14. Prueba de caja negra módulo recolección de datos

Indicador	Resultado	Observaciones
Numero de ítems	5	No soporta paginación ya que carga el contenido mediante Ajax
Soporta paginación	No soporta	
Soporta Crawler	Valido	
Numero de paginas	1	
Numero de filas recolectadas	15	

Fuente: Autor
Elaboración: Autor

Descripcion	Nombre	Clificacion	DescripcionC	Comentarios
El Romar Royal Hotel se encuentra en Loja y ofrece alojamiento, WiFi gratuita y un restaurante. El alojamiento alberga un bar. Hay aparcamiento privado gratuito en el establecimiento.	Romar Royal Hotel	9,6 , 9,3	Excepcional , Ubicación	44 comentarios
El Dulce Hogar se encuentra en Loja y cuenta con jardín y terraza. El establecimiento facilita aparcamiento privado gratuito. Los alojamientos incluyen terraza y zona de salón comedor.	Dulce Hogar	9,5	Excepcional	39 comentarios
El Hotel Howard Johnson Loja ofrece bañera de hidromasaje, piscina, gimnasio y habitaciones con conexión Wi-Fi gratuita. Sirve desayunos y cuenta con 2 restaurantes.	Hotel Howard Johnson Loja	9,0	Fantástico	218 comentarios

Figura 27. Resultado de recolección de datos

Fuente: Autor
Elaboración: Autor

En el Anexo 9, se ilustran algunos resultados obtenidos de la extracción de datos por el usuario teniendo en consideración los indicadores propuestos en la fase 1 del proyecto.

- **Pruebas de transformación de datos**

Se establece un CSV con una lista de personas con atributos propios de cada persona. En la Tabla 15, tomando en cuenta los vocabularios existentes dentro de la Web Semántica. Actualmente existe un vocabulario denominado “FOAF” que ayuda a describir personas y sus actividades, la aplicación además de realizar el reuso de vocabularios se puede extender a utilizar el vocabulario del usuario. La búsqueda de vocabularios son peticiones al catálogo LOV lo cual facilita la búsqueda de términos por el usuario.

Nombre: foaf:name, **Apellido:** foafLastName, **Edad:**foaf:age ,**Direccion:**voabulario_propio, **Fecha de nacimiento:** schemaBirtdate, **Telefono:**foaf:phone

Tabla 15. CSV para generar RDF.

Nombre	Apellido	edad	dirección	Fecha de nacimiento	Teléfono	País
Jeampier	Ruiz	21	Loja	21/06/1996	72677325	Chile
Juan	López	22	Loja	22/06/1995	72677465	Ecuador
Alejandra	Cardenas	21	Guayaquil	23/06/1996	46758784	Colombia
Pedro	Carrión	22	Quito	24/06/1995	26789099	Ecuador
Carolina	Cabrera	21	Loja	25/06/1996	72557689	Ecuador
Carlos	Hidalgo	22	Loja	26/06/1995	72557689	Perú
Javier	Sánchez	22	Piñas	27/06/1995	42734234	Ecuador
Teresa	Rio frio	22	Loja	28/06/195	72507689	Venezuela

Fuente: Autor
Elaboración: Autor

Se realizó la búsqueda de vocabularios como se muestra en la Figura 28 donde se ingresa a la aplicación las palabras claves definidas por el usuarioposteriormente se ingresó los datos a la aplicación y se realizó el mapeo conjunto con los valores establecidos.

Figura 28. Búsqueda de vocabularios
Fuente: Autor
Elaboración: Autor

Los resultados obtenidos dentro de esta etapa son las serializaciones que el usuario puede exportar. En la Figura 29 se muestra la salida del proceso de transformacion de datos.

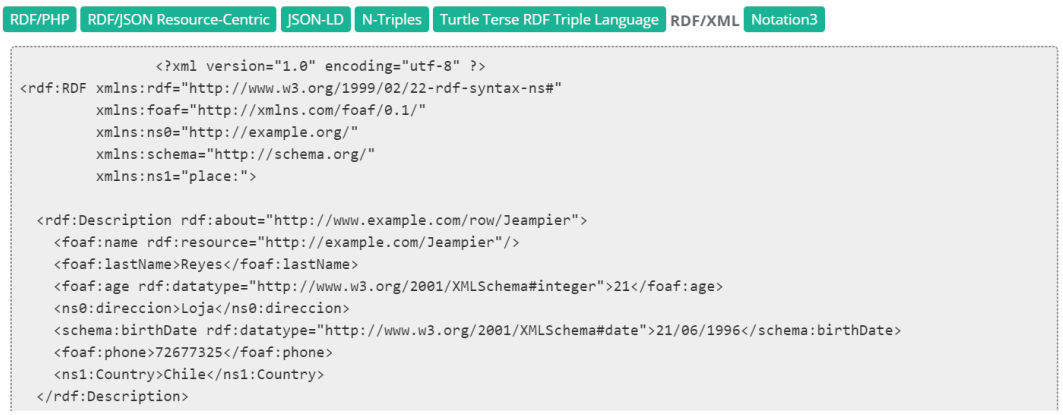


Figura 29. Resultado de la fase transformación de datos
Fuente: Autora
Elaboración: Autora

Para validar la generación de seriaizaciones se hace uso del servicio validator service de la W3C, donde se tiene como entrada la trasformación de los datos dada por la aplicacion. Al finalizar la validación el servicio genera tripletas de la información ingresada. En la Figura 30 se muestra el fragmento del resultado obtenido en el validador.

Validation Results

Your RDF document validated successfully.

Triples of the Data Model

Number	Subject	Predicate	Object
1	http://www.example.com/row/Jeampier	http://xmlns.com/foaf/0.1/name	http://example.com/Jeampier
2	http://www.example.com/row/Jeampier	http://xmlns.com/foaf/0.1/lastName	"Reyes"
3	http://www.example.com/row/Jeampier	http://xmlns.com/foaf/0.1/age	"21"^^http://www.w3.org/2001/XMLSchema#integer
4	http://www.example.com/row/Jeampier	http://example.org/direccion	"Loja"
5	http://www.example.com/row/Jeampier	http://example.org/Fechanacimeinto	"21/06/1996"^^http://www.w3.org/2001/XMLSchema#date
6	http://www.example.com/row/Jeampier	http://example.org/Telefono	"72677325"
7	http://www.example.com/row/Jeampier	place:Country	"Chile"

Figura 30. Validación RDF/XML
Fuente: Autora
Elaboración: Autora

4.3. Pruebas de rendimiento

El objetivo principal de estas pruebas es determinar la velocidad en ejecutar una tarea. En donde se toma los tiempos de respuesta de los dos módulos principales. En la Tabla 16 se detallan las pruebas de rendimiento que se realizaron.

Tabla 16. Descripción pruebas de rendimiento

Modulo	Especificación
Recolección de datos	-Se evalúa el tiempo de respuesta para obtener la página HTML dentro del iframe. -Se evalúa el tiempo de respuesta al cargar archivos .zip que contienen el archivo html. - Se evalúa el tiempo de respuesta que la aplicación tarda en recolectar datos de una determinada página.
Transformación de datos	-Se evalúa el tiempo de búsqueda de un determinado vocabulario, especificando la palabra clave. -Se evalúa el tiempo de transformación de datos a las diferentes serializaciones.

Fuente: Autora
Elaboración: Autora

Para el cálculo de los tiempos de respuesta de la aplicación se hace uso de la función time() de PHP donde se establece el tiempo inicial y tiempo final de una determinada tarea.

4.3.1. Modulo recolección de datos

Las pruebas de rendimiento de este módulo están ligadas a los contenidos que cargan cada página web, para realizar las pruebas de rendimiento se establece una lista de URLs mencionadas en la Tabla 13.

4.3.1.1. Obtener página HTML con WGET.

En la Tabla 17 se muestran los tiempos que tarda la aplicación en obtener el HTML y presentarlo dentro del iframe, el tiempo varía de acuerdo a los requerimientos de cada página.

Tabla 17. Prueba e rendimiento obtener HTML con WGET

#	Pagina	Tiempo
1	http://quotes.toscrape.com/	2.51 segundos
2	http://ieeexplore.ieee.org	No soporta WGET
3	https://www.booking.com	4.81 segundos
4	http://www.imdb.com/	5.65 segundos
5	https://www.hoteles.com/	3.63 segundos
6	https://www.ebay.com/	3.38 segundos
7	https://scholar.google.com.ec	2.90 segundos
8	http://cnnespanol.cnn.com	3.23 segundos
9	https://www.peru.travel	No soporta WGET
10	https://mercadolibre.com.ec	3.71 segundos

Fuente: Autora
Elaboración: Autora

4.3.1.2. Carga de archivos .zip a la aplicación-

En el apartado anterior se evalúa el tiempo de descarga de cada página con el comando wget. Dentro de la aplicación existe un apartado de subir archivos .html para ciertas páginas que contengan contenido dinámico en la Tabla 18 se muestra los tiempos de carga de archivos .zip que contiene el .html de cada página web.

Tabla 18. Prueba de rendimiento carga de archivos .zip

#	Pagina	Tiempo
1	http://ieeexplore.ieee.org	10.03 segundos
2	https://www.peru.travel	9.58 segundos

Fuente: Autora
Elaboración: Autora

El tiempo de carga de los archivos .zip contra el comando WGET varia notoriamente ya que se necesita subir los archivos necesarios para cargar dentro del iframe, los tiempos de carga van en relación a los requerimientos de cada página web.

4.3.1.3. Recolectar datos.

Para evaluar el tiempo de recolección de datos se establece el número de filas a recolectar teniendo en cuenta el número de ítems.

- **Recolección de datos con 10 filas afectadas.** En la Tabla 19 se muestra la URL con las filas recolectadas y el tiempo que tarda en recolectar los datos.

Tabla 19. Prueba de rendimiento recolección de datos 10 filas

#	Pagina	Filas	Scrapy	Crawler
1	http://quotes.toscrape.com/	10	8.56 segundos	9.16 segundos
2	http://ieeexplore.ieee.org	10	7.59 segundos	10.9 segundos
3	https://www.booking.com	10	8.95 segundos	9.52 segundos
4	http://www.imdb.com/	10	8.49 segundos	8.58 segundos
5	https://www.hoteles.com/	10	7.58 segundos	7.56 segundos
6	https://www.ebay.com/	10	9.65 segundos	8.45 segundos
7	https://scholar.google.com.ec	10	5.65 segundos	5.65 segundos
8	http://cnnespanol.cnn.com	10	8.75 segundos	9.52 segundos
9	https://www.peru.travel	10	9.45 segundos	8.47 segundos
10	https://mercadolibre.com.ec	10	8.56 segundos	8.26 segundos

Fuente: Autora
Elaboración: Autora

- **Recolección de datos con 50 filas afectadas** en la Tabla 20 se muestra la URL con las filas recolectadas y el tiempo que tarda en recolectar los datos.

Tabla 20. Prueba de rendimiento recolección de datos 50 filas

#	Pagina	Filas	Srapy	Crawler
1	http://quotes.toscrape.com/	50	19.56 segundos	20.40 segundos
2	http://ieeexplore.ieee.org	50	25.85 segundos	29.25 segundos
3	https://www.booking.com	50	22.56 segundos	25.36 segundos
4	http://www.imdb.com/	50	12.45 segundos	15.23 segundos
5	https://www.hoteles.com/	50	-	-
6	https://www.ebay.com/	50	13.65 segundos	15.65 segundos
7	https://scholar.google.com.ec	50	23.25 segundos	25.36 segundos
8	http://cnnespanol.cnn.com	50	-	-
9	https://www.peru.travel	50	28.34 segundos	32.25 segundos
10	https://mercadolibre.com.ec	50	22.07 segundos	24.37 segundos

Fuente: Autora
Elaboración: Autora

- **Recolección de datos con 100 filas afectadas** en la Tabla 21 se muestra la URL con las filas recolectadas y el tiempo que tarda en recolectar los datos.

Tabla 21. Prueba de rendimiento recolección de datos 100 filas

#	Pagina	Filas	Scrapy	Crawler
1	http://quotes.toscrape.com/	100	36.83 segundos	32.31 segundos
2	http://ieeexplore.ieee.org	100	43.59 segundos	45.23 segundos
4	http://www.imdb.com/	100	12.17 segundos	15.65 segundos
5	https://www.hoteles.com/	100	-	-
6	https://www.ebay.com/	100	21.15 segundos	23.52 segundos
7	https://scholar.google.com.ec	100	20.58 segundos	22.25 segundos
8	http://cnnespanol.cnn.com	100	-	-
9	https://www.peru.travel	100	54.25 segundos	57.25 segundos
10	https://mercadolibre.com.ec	100	28.87 segundos	31.25 segundos

Fuente: Autora
Elaboración: Autora

4.3.2. Módulo transformación de datos.

Las pruebas de rendimiento de este módulo tienen el objetivo de proveer dos funciones importantes: tiempo de búsqueda de vocabulario y transformación de los datos a las diferentes serializaciones RDF

4.3.2.1. Búsqueda de vocabulario.

Consiste en obtener una lista de términos de 100 resultados desde el API de LOV para realizar el match a las diferentes columnas de un CSV. En la Tabla 22 se muestra el tiempo que requiere la búsqueda de un determinado término.

Tabla 22 Prueba de rendimiento transformación de datos (búsqueda de termino)

Palabra clave	Tiempo	Termino encontrado
Name	2.01 segundos	OK
Last name	1.47 segundos	OK
Age	1.57 segundos	OK
Address	1.59 segundos	OK
Phone	1.45 segundos	OK
Country	1.45 segundos	OK

Fuente: Autora
Elaboración: Autora

4.3.2.2. Transformación a serializciones RDF.

Se evalúa el tiempo que tiene la aplicación en realizar el mapeo de los datos del CSV ingresado a las diferentes serializaciones. En la Tabla 23 se muestra el tiempo que toma en transformar los datos.

Tabla 23. Prueba e rendimiento transformación de datos (serializaciones RDF)

Salida	Tiempo
Retorna las diferentes serializaciones de representación RDF	3.25 segundos

Fuente: Autora
Elaboración: Autora

4.4. Análisis de código

Para evaluar el código fuente de la aplicación web se utiliza sonarQube la cual es una plataforma libre que ayuda al análisis del código estático, la misma ayuda a recolectar métricas para mejorar el código. En la Figura 31 se ilustra el resultado del análisis tomando en cuenta la categorización que utiliza sonarQube. Como se puede observar se obtiene un 23.2% en código duplicado por razones del módulo recolección de datos donde se hace la división de scrapy y crawler los cuales utilizan código similar

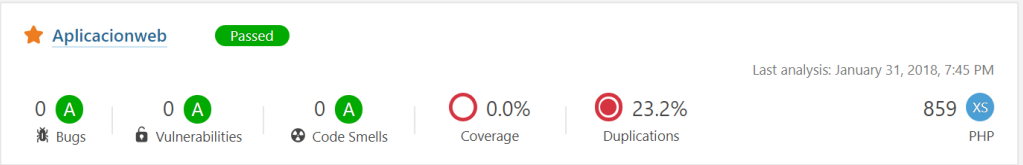


Figura 31. Análisis de código SonarQube
Fuente: Autor
Elaboración: Autor

- **Métricas evaluadas**

  Las métricas evaluadas dentro de la herramienta SonarQube realiza una categorización de tal manera que los desarrolladores lleguen al nivel “A”. A continuación, se describe cada una de las métricas generadas por SonarQube.

  - **Confiabilidad:** Se calcula la probabilidad de que se produzca un error; la herramienta realiza una categorización de 5 niveles desde la “A-E”
  - **Seguridad:** Se determina la seguridad de la aplicación, de igual manera, realiza una categorización dependiendo del número de vulnerabilidades encontradas en la aplicación.
  - **Mantenibilidad:** Determina que tan mantenible es la aplicación, tomando en cuenta las deficiencias que se encuentran en el código, así como malos hábitos de programación

Además se obtiene las métricas de tamaño de la aplicación tomando en cuenta lo desarrollado en PHP. En la tabla 24 se ilustran los porcentajes analizados por sonarQube.

Tabla 24. Métricas de tamaño

Ítem	Tamaño
Líneas de código	1.099
Funciones	32
Archivos	25
Líneas comentadas	257

Fuente: Autor
Elaboración: Autor

4.5. Despliegue de la aplicación web

Para la implementación de la aplicación web se adquiere el servicio de un servidor virtual de Digital Ocean, el cual facilita la escalabilidad y el rendimiento de la aplicación[1]. Dentro de la configuración realizada se enfatiza los siguientes puntos:

- ***Configuración inicial del servidor:*** Se crea las configuraciones necesarias para iniciar sesión dentro del droplet de Digital Ocean.
- ***Instalación de Apache, MYSQL, y PHP:*** A través de la consola del servidor se instala los paquetes necesarios para alojar la aplicación.
- ***Configuración de phpmyadmin:*** Se realiza la instalación y la configuración para administrar la base de datos.
- ***Creación de entorno virtual para alojar herramientas necesarias para recolectar datos:*** Se realiza la instalación de paquetes necesarios para la recolección de datos.

Dentro del Anexo 11 Manual de programador se detalla la tecnología y el entorno virtual de la aplicación.

4.6. Comentarios finales

Una vez culminada la fase de pruebas se determina si la aplicación cumple con los requerimientos establecidos por el usuario. Las pruebas realizadas en esta sección cubren criterios de funcionalidad, rendimiento y de calidad:

- **Pruebas de funcionalidad:** Se realizaron pruebas a distintas páginas web que contengan un filtrado de información dando como resultado observaciones de páginas que requieren un proceso adicional para completar una recolección de datos exitosa.
- **Pruebas de rendimiento:** Las pruebas de rendimiento se las realizoó a los dos módulos principales de la aplicación.

[1] *Disponible en: http://dataimport.latinubes.net.*

    - **Módulo recolección de datos:** El proceso que toma en ejecutar las herramientas necesarias para la recolección de datos es un tiempo promedio independientemente del número de filas a recolectar.
    - **Módulo de transformación de datos:** Involucra el tiempo de búsqueda de términos desde LOV y el mapeo del CSV a las diferentes serailizaciones.
- **Análisis de código:** Se evalúa el código estático en SonarQuebe el cual permite obtener métricas de calidad.

CONCLUSIONES

Después de finalizar el presente trabajo de TT y cumpliendo con los objetivos planteados se han tenido los siguientes resultados:

- Investigación de trabajos relacionados el cual permitió conocer indicadores faltantes dentro de las aplicaciones existentes hoy en día, permitiendo enfocar una solución a las limitaciones de las herramientas.
- Desarrollo de una arquitectura capaz de cumplir con los requerimientos del usuario, utilizando herramientas que ayudan a facilitar el proceso de recolección y transformación de datos.
- Desarrollo de un algoritmo capaz de generar expresiones XPATH para recolectar datos de páginas web, tomando en cuenta el árbol DOM de cada sitio web.
- Desarrollo de un algoritmo capaz de extraer datos de páginas web que cuenten con un filtrado de información.
- Desarrollo de un algoritmo capaz de realizar un mapeo de datos estructurados a serializaciones RDF.

Como conclusiones importantes tenemos:

- La integración de herramientas como Scrapy y Selenium permitió de una manera más precisa el desarrollo de un algoritmo capaz de extraer datos de diferentes páginas web ya que cuenta con funciones muy poderosas que ayudan al análisis de un HTML.
- El uso de la tecnología Jquery para armar expresiones XPATH analizando el DOM de un HTML permitió realizar un análisis de los elementos encontrados tomando en cuenta niveles superiores e inferiores de cada elemento.
- La mayoría de páginas web no utilizan estándares establecidos por la W3C dificultando tener un proceso fiable al momento de recolectar datos.
- Para tener una buena recolección de datos es necesario tener una supervisión del usuario, ya que muchas páginas web contienen fallos dentro de su HTML.
- La implementación de la metodología de desarrollo SCRUM facilito la gestión del desarrollo permitiendo que el proceso de cada etapa se realice bajo los tiempos estimados, tomando en cuenta la calidad del producto.
- El uso de catálogos de vocabularios semánticos ayudan a crear un esquema común normalizado sobre un dominio de conocimiento.
- La fase de pruebas en aplicaciones web permiten conocer los fallos de la aplicación realizando mejoras en el código (estándares de programación) y errores comunes dentro de la aplicación.

RECOMENDACIONES

Durante el proceso de desarrollo del TT se obtienen experiencias que pueden ayudar a mejorar los proyectos relacionados.

- Mantener la arquitectura diseñada dentro del servidor ya que ayuda a añadir nuevos módulos sin afectar lo desarrollado.
- La aplicación se aloje en un servidor con buenos recursos para evitar congestionamiento de la aplicación al momento de realizar la recolección de datos.
- Buscar mejores procedimientos para obtener el HTML de una página web teniendo en consideración archivos necesarios para cargar la página web.
- Desarrollar nuevas funcionalidades que permitan mejorar la forma en que el usuario recolecta los datos.
- Integrar la validación de las serializaciones RDF dentro del aplicativo.
- Integrar funcionalidades avanzadas de mapeo de datos, creando un grafo que esté vinculado a la descripción de más recursos.
- Mejorar la estructura de expresiones XPATH teniendo en cuenta la ausencia de atributos como la clase y el id de cada elemento.
- Permitir la limpieza de los datos dentro de la aplicación.

BIBLIOGRAFÍA

Berners-Lee, T. (2010). Linked Data - Design Issues. *World Wide Web Consortium*. Retrieved from https://www.w3.org/DesignIssues/LinkedData

Berners-Lee, T., & Miller, E. (2002). Semantic Web. *October*, (51).

Bizer, C., Heath, T., & Berners-Lee, T. (2009). Linked data-the story so far. *International Journal on Semantic Web and Information Systems*, *5*(3), 1–22. https://doi.org/10.4018/jswis.2009081901

Brickley, D., & Miller, L. (2009). FOAF Vocabulary Specification 0 . 91 Status of This Document. *World Wide Web Internet And Web Information Systems*, (November 2007), 3–6.

Butler, D. (2006). The web-wide world. *Nature*, *439*(7078), 776–778. https://doi.org/10.1038/439776a

Castells, P. (2005). La web semántica. *Sistemas Interactivos Y Colaborativos En La Web*, 195–212.

Consortium, W. W. W., & Others. (2014). Guía Breve de WebSemántica.

Ermilov, I., Bis, A., & Stadler, C. (2012). Crowd-Sourcing the Large-Scale Semantic Mapping of Tabular Data. *Web Science*, 1–10.

Figueira, D. (2017). Satisfiability of XPath on data trees To cite this version : HAL Id : hal-01670363 Satisfiability of XPath on data trees.

Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing? *International Journal of Human-Computer Studies*, *43*(5–6), 907–928. https://doi.org/10.1006/ijhc.1995.1081

Gyrard, A., Patel, P., Datta, S. K., & Ali, M. I. (2017). Semantic Web Meets Internet of Things and Web of Things: [2nd Edition]. *Proceedings of the 26th International Conference on World Wide Web Companion*, 917–920. https://doi.org/10.1145/3041021.3051100

Hewson, C., & Stewart, D. W. (2016). Internet Research Methods. *Wiley StatsRef: Statistics Reference Online*, 1–6. https://doi.org/10.1002/9781118445112.stat06720.pub2

Knoblock, C. A., Szekely, P., Ambite, J. L., Goel, A., Gupta, S., Lerman, K., … Mallick, P. (2012). Semi-automatically mapping structured sources into the semantic web. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7295 LNCS*, 375–390.

https://doi.org/10.1007/978-3-642-30284-8_32

Laufer, C. (2015). Guia_Web_Semantica. Retrieved from http://ceweb.br/guias/web-semantica//es/capitulo-4/

Lyssania, Macías, Layla, M. (2009). Los recursos de la Web 2.0 para el manejo de información académica. *Revista Fuente*, *1*(1), 18–27. Retrieved from http://fuente.uan.edu.mx/publicaciones/01-01/los_recursos_de_la_web_2.0_para_el_manejo_de_informacion_academica.pdf

Mozilla. (2016). Introducción - Referencia DOM de Gecko _ MDN. Retrieved from https://developer.mozilla.org/es/docs/Referencia_DOM_de_Gecko/Introducción

Ortiz-Repiso Jiménez, V. (1999). Nuevas perspectivas para la catalogación: Metadatos Versus Marc. *Revista Española de Documentación Científica*, *22*(2), 198–219. https://doi.org/10.3989/redc.1999.v22.i2.338

Pastor Sanchez, J. A. (2011). *Tecnologias de la Web Semantica*.

Sanchez, P. (2016). SKOS.

Schwaber, K., & Sutherland, J. (2013). La Guía de Scrum. *Scrumguides.Org*, *1*, 21. Retrieved from http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf

Scrapy. (2016). XPath Tutorial Exemple 1. Retrieved from http://zvon.org/xxl/XPathTutorial/Output_fre/example1.html

Senso, J., & Piñero, A. de la R. (n.d.). Evolución del Dublin Core Metadata Initiative. *Ugr.Es*, 1–26. Retrieved from http://www.ugr.es/~jsenso/curriculum/dcmi.pdf

W3C. (2001). N-Triples. Retrieved from https://www.w3.org/2001/sw/RDFCore/ntriples/

W3C. (2004a). RDF Schema. *WikiPedia*. Retrieved from http://www.w3.org/TR/rdf-schema/

W3C. (2004b). Vista General del Lenguaje de Ontologías Web (OWL). Retrieved from https://www.w3.org/2007/09/OWL-Overview-es.html

W3C. (2014a). Html 5.

W3C. (2014b). Rdf 1. Retrieved from https://www.w3.org/TR/rdf11-concepts/

Zajicek, M. (2007). Web 2.0. *Proceedings of the 2007 International Cross-Disciplinary Conference on Web Accessibility (W4A) - W4A '07*, 35. https://doi.org/10.1145/1243441.1243453

ANEXOS

Anexo 1 Historias de usuario

	Enunciado de la historia				Criterios de aceptación			
Identificador (ID) de la historia	**Rol**	**Característica / Funcionalidad**	**Razón / Resultado**	**Número (#) de escenario**	**Criterio de aceptación (Título)**	**Contexto**	**Evento**	**Resultado / Comportamiento esperado**
00-0000-0001	Usuario	Necesito ingresar a la aplicación	Con la finalidad de tener acceso a la herramienta	1	Autenticación	En caso que el usuario ingrese usuario y contraseña a la aplicación	Cuando haga clic en un botón	El sistema debe realizar una consulta para verificar la existencia del usuario y permitir acceso.
				2	Restricciones de ingreso	En caso que el usuario ingrese usuario y contraseña incorrecta	Cuando haga clic en un botón	El sistema debe alertar al usuario que sus credenciales son incorrectas.
00-0000-0002	Usuario	Necesito registrarme a la aplicación	Con la finalidad de tener acceso a la herramienta	1	Almacenamiento	En caso que el usuario ingrese una URL.	Cuando se ingrese la URL del sitio	El sistema debe soportar sitios Web dinámicos como estáticos

00-0000-0003	Usuario	Necesito extraer datos de internet	Obtener un almacen de datos.	1	Acoplado a páginas tanto estáticas como dinámicas	En caso que el usuario ingrese una URL.	cuando se ingrese la URL del sitio	El sistema debe soportar sitios Web dinámicos como estáticos
				2	Listar metadatos encontrados	En caso de que la página tenga metadatos encontrados.	Cuando se ingrese la URL del sitio	El sistema debe mostrar al usuario los metadatos encontrados en el sitio Web,
				3	Exportar información en formatos como CSV, XLS	En caso que el usuario seleccione los metadatos que requiere.	Cuando seleccione los metadatos de la página	El sistema debe permitir exportar la información en formatos como: CSV, XLS

00-0000-0004	Usuario	Necesito almacenar y consultar datos recolectados	Con la finalidad de que el usuario reutilice datos recolectados	1	Almacenamiento	En caso que el usuario requiera guardar los datos recolectados	Cuando haga clic en un botón	El sistema debe permitir realizar el almacenamiento de los datos recolectados
				2	Consulta de datos	En caso que el usuario necesite reutilizar los datos	Cuando haga clic en un botón	El sistema debe permitir realizar búsqueda de los datos recolectados

						recolectados con anterioridad.		
00-0000-0005	Usuario	Necesito establecer vocabularios	Con la finalidad de reusar vocabularios existentes y si no existen definir clases	1	Buscar vocabularios,	En caso que existan vocabularios dentro del catálogo LOV que se acoplen al dominio.	cuando se agregue un objeto (circulo, cuadrado, flecha)	El sistema debe permitir interactuar con el usuario de tal manera que al momento de crear una clase, propiedad o instancias se pueda buscar vocabularios de LOV
				2	Definir URI, terminos	En caso que ningún vocabulario se acople al dominio	cuando se agregue un objeto (círculo, cuadrado, flecha)	El sistema debe permitir interactuar con el usuario de tal manera que al momento de crear una clase, propiedad o instancias se pueda establecér el vocabulario definido por el usuario
				3	Asignar columna de datos	En caso que el usuario necesite generar el RDF	Cuando haga clic en un botón	El sistema debe permitir seleccionar las columnas de datos

								y asignarlas a cada objeto
00-0000-0007	Usuario	Necesito que exista un mecanismo para generar RDF a partir de ID 5 y 6	Con la finalidad de establecer un marco para definir metadatos	1	Generar RDF	En caso que el usuario requiera convertir los datos a RDF	Cuando haga clic en un botón	El sistema debe permitir generar RDF a partir de los datos ingresados

Anexo 2 Product Backlog

Identificador (ID) de la Historia	Enunciado de la Historia	Alias	Estado	Prioridad
00-000-0001	Como un usuario, necesito registrarme e ingresar a la aplicación web, con la finalidad de tener acceso a funcionalidades dentro de la aplicación	R001	TERMINADO	MEDIA
00-000-0003	Como un usuario, necesito extraer datos de internet, con la finalidad de dar semántica a los datos	E001	TERMINADO	ALTA
00-000-0004	Como un usuario, necesito almacenar y consultar datos recolectados, de que el usuario reutilice los datos.	A002	TERMINADO	MEDIA
00-000-0005	Como un usuario, necesito establecer vocabularios (nuevo, reuso) y asignar data, con la finalidad de reusar vocabularios existentes y si no existen definir un vocabulario.	V001	TERMINADO	ALTA
00-000-0006	Como un usuario, necesito exportar los datos a diferentes serializaciones RDF.	S01	TERMINADO	ALTA
00-000-0006	Como usuario necesito manuales de usuario y de programador.	M001	TERMINADO	ALTA

Anexo 3 Sprint Backlog

Identificador (ID) de ítem de Product backlog	**Tarea**	**Estatus**	**Días estimados totales**
R001	Diseño de la arquitectura	Terminado	2
	Diseño de base de datos	Terminado	1
	Diseño del flujo de la aplicación	Terminado	2
	Diseño de la interfaz web	Terminado	10
I001	Desarrollo del módulo Login	Terminado	3
	Desarrollo del módulo registro	Terminado	3
E001	Desarrollo de gestión URL	Terminado	7
	Desarrollo de Xpath	Terminado	10
	Desarrollo de algoritmo de recolección de datos(SCRAPY)	Terminado	7
	Desarrollo del algoritmo recolección de datos (CRAWLER)	Terminado	7
	Implementación del algoritmo (SCRAPY y CRAWLER)	Terminado	7
V001	Consumir vocabularios en LOV	Terminado	3
	Desarrollo de mecanismo para generar RDF	Terminado	3
	Despliegue de interfaz Web	Terminado	5
M0001	Documentación de manual de usuario Documentación de manual de programador	Terminado	3

Anexo 4 Diccionario de datos

Tabla usuarios: Almacena la información de los usuarios registrados en la aplicación.

Tabla Usuarios					
Llave	Nombre	Campo	Tipo	Tamaño	Descripción
PK	ID de usuario	idUsuarios	INT		Almacena el código único por cada usuario
	Nombre de usuario	nombreUsuario	TEXTO	45	Almacena el nombre del usuario
	Email de usuario	email Usuario	TEXTO	45	Almacena el email del usuario
	Contraseña del usuario	contrasenaUsuario	TEXTO	45	Almacena la contraseña del usuario

Tabla Proyecto: Almacena la información de los proyectos generados por el usuario.

Tabla Proyecto					
Llave	Nombre	Campo	Tipo	Tamaño	Descripción
PK	ID del proyecto	idProyecto	INT		Almacena el código único por cada proyecto
	Nombre del proyecto	nombreProyecto	TEXTO	45	Almacena el nombre del proyecto
	Ruta del proyecto	pathProyecto	TEXTO	45	Almacena la ruta del proyecto
FK	ID de usuario	Users_idUsuariios	INT		Almacena el ID del usuario

Tabla configuración proyecto: Almacena la información de los eventos generados por el usuario.

Tabla Configuraciones_Proyecto					
Llave	Nombre	Campo	Tipo	Tamaño	Descripción
PK	ID de la configuracion	idConfiguracion_proyecto	INT		Almacena el código único por cada proyecto
	Nombre del item	Nombre_item	TEXTO	45	Almacena el nombre del item
	Texto seleccionado	Texto	TEXTO	150	Almacena el texto de la etiqueta
	Expresión Xpath	Xpath	TEXTO	45	Almacena la expresión Xpath
	Modo de extracción	Modo_extraccion	TEXTO	45	Almacena el modo de extracción
FK	ID de proyecto	Proyecto_idProyecto	INT		Almacena el ID del usuario

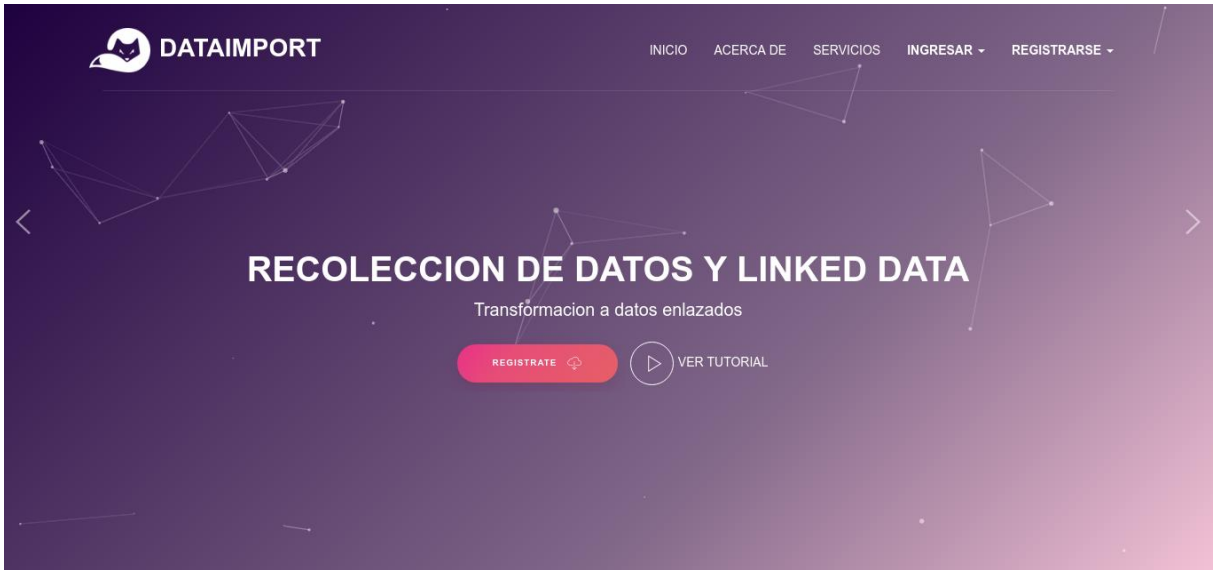
Tabla proyectos LD: Almacena la información de los datos ingresados por el usuario.

Tabla Proyecto_LD					
Llave	Nombre	Campo	Tipo	Tamaño	Descripción
PK	ID del proyecto	IdProyecto_LD	INT		Almacena el código único por cada proyecto
	Nombre del proyecto	nombreProyecto	TEXTO	45	Almacena el nombre del proyecto
	Ruta del proyecto	pathProyecto	TEXTO	45	Almacena la ruta del proyecto
FK	ID de usuario	Users_idUsuariios	INT		Almacena el ID del usuario

Anexo 5 Desarrollo de la interfaz Web

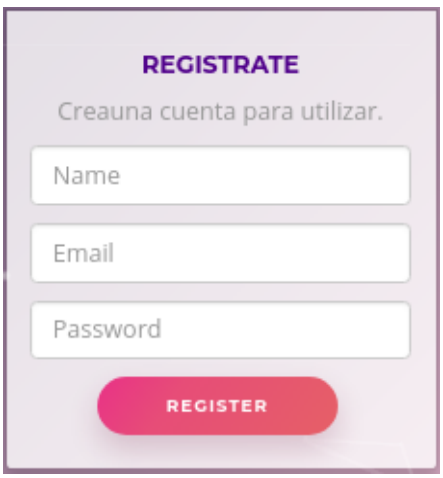
Desarrollo de interfaz principal

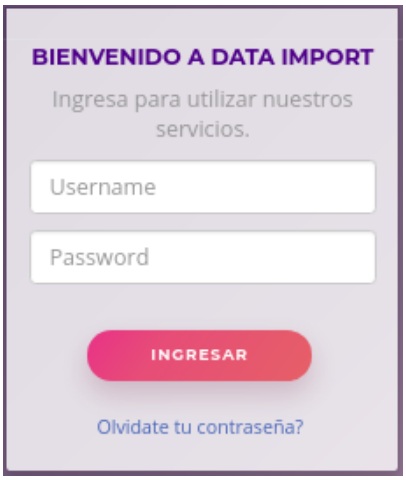
El portal principal contiene secciones que engloba las funcionalidades del proyecto, ofreciendo al usuario una interfaz sencilla pero intuitiva de lo que realiza la aplicación, dentro de esta interfaz se toma en cuenta aspectos generales que.



Desarrollo de interfaz registro e ingreso

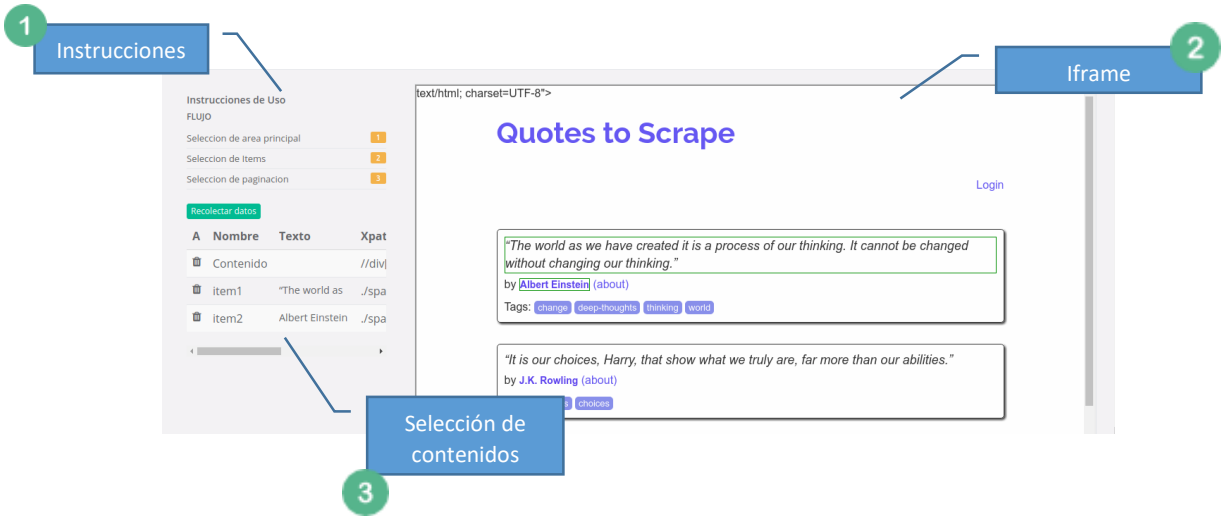
Dentro de la interfaz para la sección registro se toma en cuenta un identificador único por cada usuario, así mismo para la sección de ingreso tendrá que acceder por medio del usuario y clave registrada.





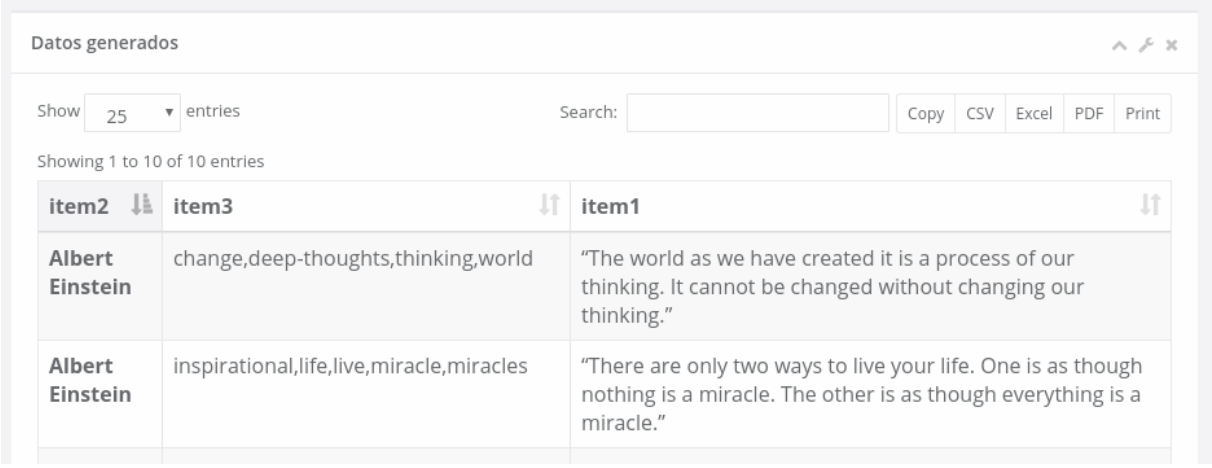
Desarrollo de la interfaz área de trabajo (Scrapy- Crawler)

En esta sección se desarrolla la interfaz web en donde el usuario podrá extraer la información de manera iterativa en la siguiente se muestra la maquetación de este componente donde se definen tres secciones importantes: sección instrucciones se detalla el flujo a seguir para obtener una recolección exitosa, sección Iframe se recupera el .html de la página web ingresada incluyendo los estilos css para la selección del contenido HTML, en la sección 3 se encuentra una tabla de toda la información que el usuario ingrese.



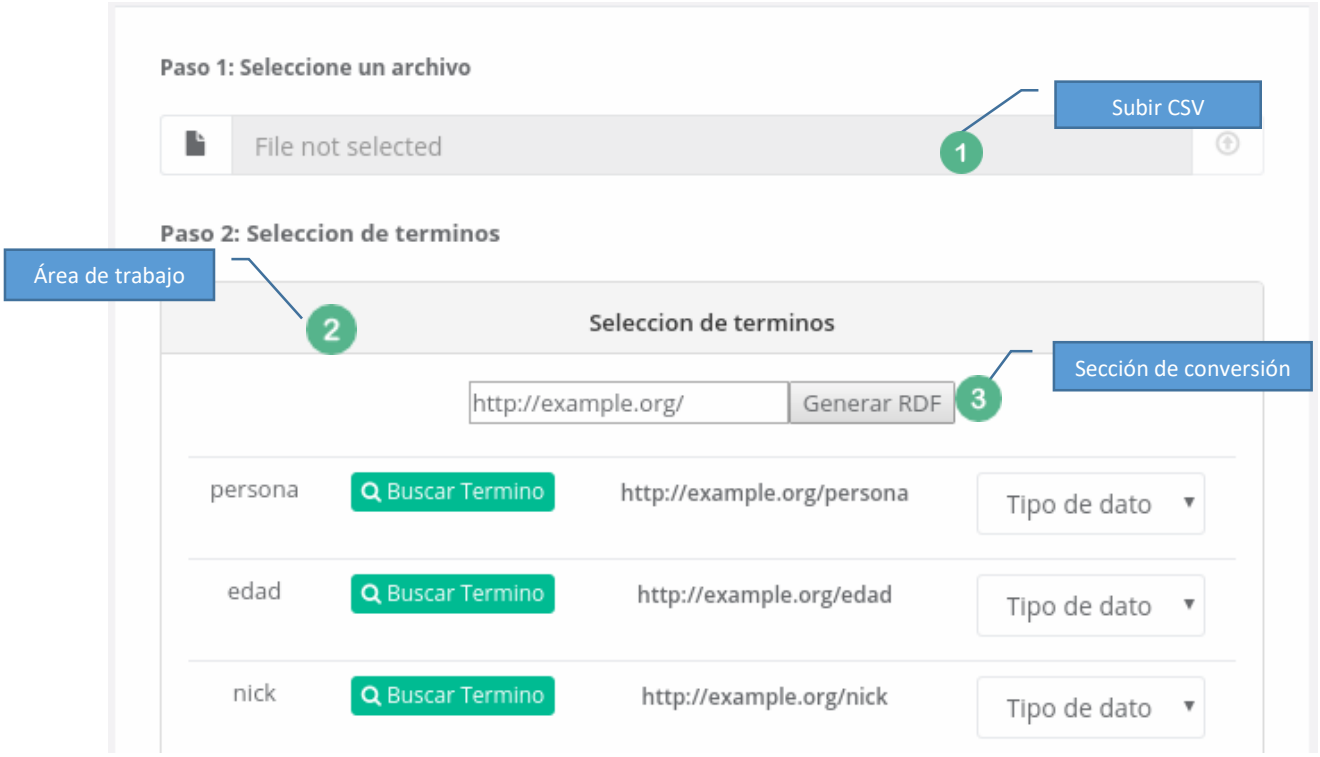
Desarrollo de interfaz resultado

Se realiza una interfaz web para mostrar el contenido recolectado, el cual se hace uso de tablas para proyectar la información, teniendo en cuanta que se puede realizar filtros de búsqueda y exportaciones a diferentes formatos.



Diseño de interfaz Transformación de datos

La interfaz web del proceso de transformación de datos está compuesta por cuatro secciones: cada una de ellas sigue un flujo para realizar la transformación de datos.



Diseño de interfaz resultado de transformación

Se compone de una serie de botones en la cual el usuario pueda realizar la transformación a una socialización RDF, además de contar con la opción de descarga de cualquier tipo de socialización.

Anexo 6 Script para construir XPATH

```
<script type="text/javascript">
        //Inicializacion del documento
        $(document).ready(function () {
           //DECLARACION DE VARIABLES
           var aux=0
           var class_valido=[]
           //Llamado a iframe
           var inframeDoc = document.getElementById('myframeuj').contentWindow;
           //Iteration sobre elementos DOM
           $(inframeDoc).mouseover(function (event) {
              class_valido[0]=$(event.target).attr('class');
              if(class_valido!=""){
                 longi=class_valido[0].length;
                 var ps=class_valido[0].slice(-1)
              }
              if(ps==" "){
                 $(event.target).addClass('clase_espacio');
              }

              $(event.target).addClass('outline-element');
           }).mouseout(function (event) {
              $(event.target).removeClass('outline-element');
           }).click(function (event) {

             //Proceso de XPATH
              var aux_clase = "";

              console.log(this)
              var tagNames = this.nodeName;
              var tel=[]
              //Negar enlaces
              var nFilas = $("#mi-tabla tr").length;
              If (tagNames != "A") {
                 event.preventDefault();
                 $(event.target).removeClass('outline-element');
```

```
            ps=getElementXPath (event.target,nFilas);
        }
        //Llamar a método getElementXPath
        ps = getElementXPath(event.target,nFilas);

        $(event.target).toggleClass('outline-element-clicked');

    });
});

 //Función ayuda a recuperar los hijos de un elemento seleccionado
function childs(element){
    var parent_elemet =$(element).clone();
    child_elements=parent_elemet.children()
    return child_elements
}
//Método para encontrar padres
 function find_text_in_parent(parent){
    var parent_elemet =$(parent).clone();
    child_elements=parent_elemet.children()
    log=child_elements.length
    papas=child_elements.remove()
    parent_elemet=parent_elemet.text()
    return [parent_elemet,log,child_elements];
 }
 //Método para obtener un xpath de un elemento seleccionado
 function getElementXPath(element,aux) {

            //Declaration de variables
             var aux_clase = "";
    i=0
    var j=0
    var parent_celement=[];
    var clas_cotar=" ";
    return $(element).addBack().map(function () {
    var $this = $(this);
```

```
var tagNames = this.nodeName;
var p = $(element).attr('class');
tagNames=tagNames.toLowerCase();
var banderas
var texto =$(element).text();
var name_clase_hijo=""
var parent_felement =$(element).parent();
var auxiliar_exp=""
var get_parent=parent_felement[0]
var exp_first="";
var name_clase_padre1=""
var parent_telement=get_parent.tagName.toLowerCase();
parent_celement[0]=$(get_parent).attr('class');
aux_padre1='class="'+parent_celement[0]+"";
s =$(parent_felement).parent();
var get_parents=s[0]
        var parent_telements=get_parents.tagName.toLowerCase();
   parent_c=$(s).attr('class');

        //Llamada a metodos recorrer DOM

   aux_padre2='class="'+parent_c+"";
   hijo_clase=p.split(' outline-element-clicked').length - 1;
   padre1_clase=aux_padre1.split(' outline-element-clicked').length - 1;
   padre2_clase=aux_padre2.split (' outline-element-clicked').length - 1;
   if(padre1_clase>=1){
      aux_padre1=aux_padre1.replace(" outline-element-clicked","")
   }
   if(padre2_clase>=1){
      aux_padre2=aux_padre2.replace(" outline-element-clicked","")
   }
   t=$('#myframeuj .quotesss');
   ' outline-element-clicked'

   if (aux==1){
```

```
            if(p!=null){
               valido_exp='./'+tagNames+'[@class="'+p+'"]';
            }else{
               valido_exp='. /'+tagNames;
            }

            valido_exp=valido_exp.replace(" outline-element-clicked","")

            variable_estatica=valido_exp;

         }else if (aux>1){
            container = document.getElementById('myframeuj');
         varis=$(container).children();

         tg=$('#myframeuj').contents().find("body").html();
         primer_padre=tg.split(aux_padre1).length - 1;
         segundo_padre=tg.split(aux_padre2).length - 1;
```

//Lógica de construccion XPATH teniendo en cuenta padres, hijos y hermanos

```
         if(p!=""){
            name_clase_hijo='[contains(concat(" ", normalize-space(@class), " "),'+"' '+p +' ")]'
         }if(parent_celement[0]!=""){
            name_clase_padre1='[contains(concat(" ", normalize-space(@class), " "),'+"' '+parent_celement[0] +' ")]'
         }if(parent_c!=""){
            name_clase_padre2='[contains(concat(" ", normalize-space(@class), " "),'+"' '+parent_c +' ")]'
         }

         if(primer_padre==segundo_padre){
```

```
            exp_first = "./" + parent_telements
+name_clase_padre2+"/"+parent_telement+name_clase_padre1+"/"+tagNames+name_clas
e_hijo;


          }else if (primer_padre>segundo_padre){
           exp_first = "./"
+parent_telement+name_clase_padre1+"/"+tagNames+name_clase_hijo;
          }else if (segundo_padre>primer_padre){
           exp_first = "./"+tagNames+name_clase_hijo;
        }
        cls1=name_clase_padre1.split(" ").length - 1;
           cls2=name_clase_padre2.split(" ").length - 1;
        if(cls1>1 || cls2>1){
           exp_first = "./"+tagNames+name_clase_hijo;
        }
        clss=exp_first.split('[@class="outline-element-clicked"]').length - 1;
        out=exp_first.split(" outline-element-clicked").length - 1;
        valido_exp=exp_first;
        if(clss>=1){
           valido_exp=valido_exp.replace('[@class="outline-element-clicked"]',"")
        }
        if(out>=1){
           valido_exp=valido_exp.replace(" outline-element-clicked","")
        }
        padres = find_text_in_parent(element);
           hijos_extarct=padres[2]
           t=0
           z=3
           var arreglos=['principio']
           if (padres[1]>1){
              bandera=1
                 do{
                 ps=Metodo_texto_hijo(hijos_extarct,texto)
                 hijos_extarct=ps[0]
                 expresion=ps[2]
```

```
        anterior=expresion;
        bandera=ps[1]

        if(arreglos[t]==expresion){
           break
        }
        auxiliar_exp=auxiliar_exp+expresion
        t=t+1;
        arreglos.push(expresion)
        longitud=arreglos.legth
        }while(z==3)
}


}


sizes=valido_exp.substr(0, variable_estatica.length)

if(sizes==variable_estatica & aux>1){
   valido_exp=valido_exp.replace(variable_estatica,"")
   valido_exp="."+valido_exp

}
if(auxiliar_exp!=""){
   valido_exp=valido_exp+"/"+auxiliar_exp
}
contien_undefined=valido_exp.split('[@class="undefined"]').length - 1;
contien_out=valido_exp.split(' outline-element-clicked').length - 1;
if(contien_out>=1){
   valido_exp=valido_exp.replace(" outline-element-clicked","")
}
if(contien_undefined>=1){
   do{
      valido_exp=valido_exp.replace('[@class="undefined"]',"")
      v=valido_exp.split('[@class="undefined"]').length - 1;
```

```
            }while(v==1)

        }
        contien_vacio=valido_exp.split('[@class=""]').length - 1;
        if(contien_vacio){
            valido_exp=valido_exp.replace('[@class=""]',"")
        }
        valido_exp=valido_exp.replace('clase_espacio',"")
        return valido_exp+'/text()';

    }).get().join("/");

}
//Metodo longitud hijo
function longchild(element){
    varis=$(element).children();
    longitud=varis.length;
    return longitud;
}
// Método que retorna igualdad de texto
function Metodo_texto_hijo(hijos,txt,tag){
    long=hijos.length
    var j=0
    while(j<long){
        texto_hijo=hijos[j]
        tvt=$(texto_hijo).text()

        indice=txt.indexOf(tvt)
        if(indice>=0){
            element=hijos[j];
            tags=element.tagName;
            ftagName=tags.toLowerCase();
            var pa = $(texto_hijo).attr('class');
            if(typeof pa==="undefined"){
                exps = ftagName ;
            }else{
```

```
                exps = "/"+ftagName + "[@class=" + "'" + pa + "'" + "]/";

            }
        }
        child_nuev=childs(hijos[j])
        hij=longchild(hijos[j])
        if(hij>=1){
            bandera=0
            child_nuev=childs(hijos[j])
        }
            j=j+1
    }
    return [child_nuev,bandera,and exps]
  }

</script>
```

Anexo 7 Archivo generador RDF.

```
<?php
ob_start();
session_start();
//LLamada de libreris EasyRDF
set_include_path(get_include_path() . PATH_SEPARATOR .
'/opt/lampp/htdocs/arch/ld/easyrdf/lib/');
require_once "EasyRdf.php";
//Llamada al archivo subido por el usuario
$csv=$_SESSION['csv'];
$operation=$_SESSION['operation'];
$arreglo_prefix=$_SESSION['arreglo_prefix'];
//variables que almacenan datos del csv
$long_alt=count($csv);
$long_anch=count($csv[1]);
$long_anch=$long_anch-1;
$long_alt=$long_alt-1;
//Creacion de un grafo para los datos CSV
$graph = new EasyRdf_Graph();
//Mapeo del csv haciendo fusion con EasyRdf
for ($i = 1; $i <= $long_alt; $i++) {
    $me = $graph->resource('http://www.example.com/row/'.$csv[$i][0]);
    $pf=$csv[$i];
    for ($j = 0; $j <= $long_anch; $j++) {
        $aux=$operation[$j];
        if($aux==0){
            $me->set($arreglo_prefix[$j], $pf[$j]);
        }else if($aux==1){
            $me->set($arreglo_prefix[$j],$graph->resource("http://example.com/".$pf[$j]));
        }else if($aux==2){
            $me->set($arreglo_prefix[$j],new EasyRdf_Literal_Date($pf[$j]));

        }else if($aux==3){
            $me->set($arreglo_prefix[$j],new EasyRdf_Literal_INteger($pf[$j]));
        }
```

```
    }
}
//Serializaciones en diferentes formatos
if (isset($_REQUEST['format'])) {
    $format = preg_replace("/[^\w\-]+/", '', strtolower($_REQUEST['format']));
} else {
    //Formato predeterminado
    $format = 'ntriples';
}

?>
```

Anexo 9 Pruebas de funcionalidad

Dentro delas pruebas realizadas a las diferentes URLs ingresadas al sistema se obtuvo los siguientes resultados:

Prueba 001 página Quotes:

Indicador	Resultado	Observaciones
Numero de ítems	3	
Soporta paginación	Valido	
Soporta Crawler	Valido	
Numero de paginas	10	
Numero de filas recolectadas	100	
Tiempo de recolección	13.25seg	

En la Figura siguiente se muestra el resultado en pantalla que el usuario recibe en la recolección de datos

item2	item3	item1
Albert Einstein	change,deep-thoughts,thinking,world	“The world as we have created it is a process of our thinking. It cannot be changed without changing our thinking.”
Albert Einstein	inspirational,life,live,miracle,miracles	“There are only two ways to live your life. One is as though nothing is a miracle. The other is as though everything is a miracle.”

Prueba 002 página IEEE.

Indicador	Resultado	Observaciones
Numero de ítems	3	La página IEEE carga el contenido dinámicamente es por ello el tiempo de recolección aumento.
Soporta paginación	Valido	
Soporta Crawler	Valido	
Numero de paginas	5	
Numero de filas recolectadas	125	
Tiempo de recolección	19.25seg	

En la Figura siguiente se muestra el resultado en pantalla que el usuario recibe en la recolección de datos

Tamanio	Titulo	autores
(756 Kb)	Omics ,: From Scattered Individual Software Tools to Integrated Workflow Management Systems,IEEE/ACM Transactions on Computational Biology and Bioinformatics	Tianle Ma,Aidong Zhang
(682 Kb)	Contributing to the OER movement: A practical experience: The case of the , School, UNA,2017 XLIII Latin American Computer Conference (CLEI)	Sonia Mora Rivera,Mayela Coto Chotto
(663 Kb)	From information to reflection-design strategies for personal ,2017 IEEE Life Sciences Conference (LSC)	Naseem Ahmadpour,Karen Anne Cochrane

Prueba 004 página Hoteles

Indicador	Resultado	Observaciones
Numero de ítems	7	La página no cuenta con paginación soportada en la aplicación ya que se realiza un scroll para poder generar más datos
Soporta paginación	No soporta	
Soporta Crawler	Valido	
Numero de paginas	1	
Numero de filas recolectadas	25	
Tiempo de recolección	11.01seg	

En la Figura siguiente se muestra el resultado en pantalla que el usuario recibe en la recolección de datos

calles	ciudad	Caalificacion	nombre	Pais	Visitas	Telefono
2km North of the Town Square	, Loja	10.0	Madre Tierra Hotel Spa,Loja	Ecuador	280 reviews	0203 450 6788
Jose Antonio Eguiguren y 18	, Loja	10.0	Romar Royal Hotel,Loja	Ecuador	36 reviews	0203 450 6788
Av Zoilo Rodriguez	, Loja	9.2	Howard Johnson Hotel Loja,Loja	Ecuador	155 reviews	0203 450 6788
Colón 14-30 y Bolívar	, Loja	9.0	Hotel Libertador,Loja	Ecuador	33 reviews	0203 450 6788
Miguel Riofrio 14-62	, Loja	8.8	Zamorano Real Hotel,Loja	Ecuador	52 reviews	0203 450 6788

Prueba 005 página Ebay

Indicador	Resultado	Observaciones
Numero de ítems	3	

Soporta paginación	Valido	
Soporta Crawler	Valido	
Numero de paginas	5	
Numero de filas recolectadas	200	
Tiempo de recolección	18.01seg	

En la Figura siguiente se muestra el resultado en pantalla que el usuario recibe en la recolección de datos

Pais	Precio	Nombre_Libro
, , , , , , , **From Venezuela**	$4.75	Pack 12 libros Digitales en Pdf Fiodor Dostoevskii Literatura Disfrutar Leer
, , , , , , , **From United States, ,**	$495.00	PRAELECTIONES IN DUODECIM LIBROS CODICIS JUSTINIANI IMP
, , , , , , , **From United States, , , ,**	$273.00	Amigos / Friends (Libros Gigantes) (Spanish Edition)-ExLib
, , , , , , , **From United States, , , ,**	$152.39	Tres libros : ensayos y poemas. De fusilamientos. Prosas dispersas (Letras Mexic
, , , , , , , **From United States, , , ,**	$273.00	Los 100 Mejores Libros Del Siglo XX

Prueba 006 página CNN

Indicador	Resultado	Observaciones
Numero de ítems	4	
Soporta paginación	Valido	
Soporta Crawler	Valido	
Numero de paginas	3	
Numero de filas recolectadas	50	
Tiempo de recolección	16.01seg	

item2	item3	item1	item4
Ana María Cañizares, ,CNN en Español	, (16:29 GMT) 27 enero, 2018	Explosión cerca de comando policial de Ecuador deja 13 agentes heridos, , ,	Por ,, , , , ,El estallido de un artefacto explosivo en la parte posterior de un comando policial en Ecuador la madrugada de este sábado dejó al menos 13 heridos, informaron autoridades ecuatorianas. El presidente Lenín Moreno dijo en Twitter que el ataque es "un acto terrorista ligado a bandas de narcotraficantes" y decretó el estado de excepción en dos ciudades.
Ana María Cañizares, ,CNN en Español	, (02:06 GMT) 24 enero, 2018	Lanzan huevos a Rafael Correa en un acto de campaña, , ,	Por ,, , , , , Durante un recorrido en el sector de La Maná, a tres horas de Quito, en la campaña por el "no" en la consulta popular, el expresidente de Ecuador, Rafael Correa, pasó un mal momento luego de que un grupo de personas lanzaran huevos y frutas al camión que lo transportaba a él y a miembros de su equipo de seguridad, que lo protegieron con escudos y parasoles.
CNN Español	, (18:02 GMT) 1 febrero, 2018	El ELN recluta venezolanos, dice comandante de las Fuerzas Militares de Colombia, , ,	Por , , , ,El máximo comandante de las Fuerzas Militares de Colombia, general general Alberto José Mejía, dijo que los cabecillas de esa guerrilla se esconden en Venezuela.

Anexo 10 Manual de usuario

Manual de usuario

Proyecto DATAIMPORT

Autor: Ana Cristina Cardenas

Versión: 001

Fecha: 05/02/2018

1. Introducción

El presente documento muestra al usuario el funcionamiento de la aplicación denominada DATAIMPORT la cual ayuda a la recolección de datos de internet y la transformación de datos estructurados a serializaciones RDF.

2. Objetivos

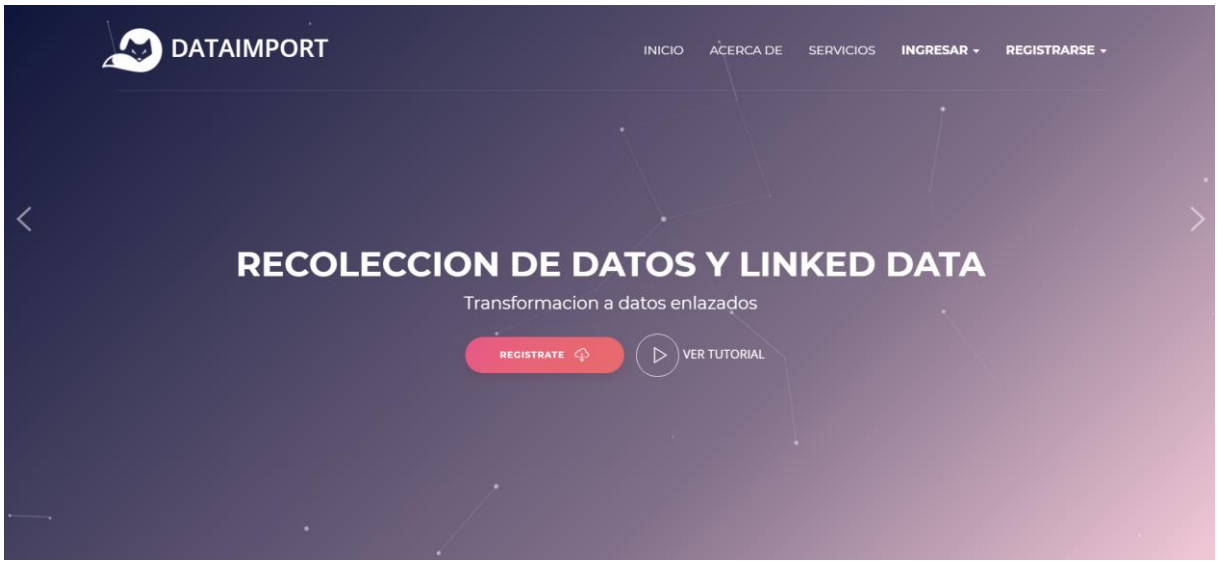
Se pretende mostrar de manera clara el funcionamiento de la aplicación y tener constancia de un requerimiento establecido en el alcance del proyecto.

3. Manual de usuario

Para el ingreso de la aplicación el usuario debe ingresar la dirección http://46.101.8.12 en el navegador posteriormente se despliega la pantalla principal.

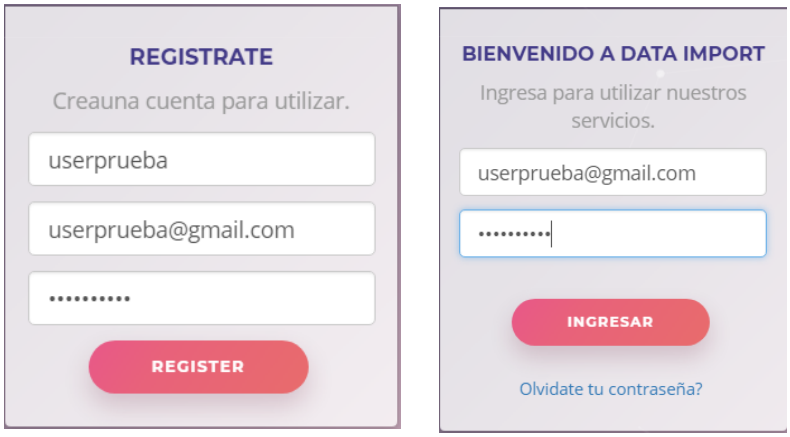
3.1. Pantalla de inicio

La pantalla de inicio da al usuario información de lo que ofrece la aplicación, además se encuentra un video tutorial en la que se realiza todo el proceso de recolección y transformación de datos y las secciones de registro e ingreso a la aplicación.



3.1.1. Registro e ingreso

Dentro de esta sección el usuario requiere un registro previo para utilizar las funcionalidades de la aplicación, para realizar el registro es necesario que el usuario llene todos los campos



3.1.2. Área de trabajo

Una vez iniciada sesión el usuario puede acceder al área de trabajo donde está compuesta por tres secciones

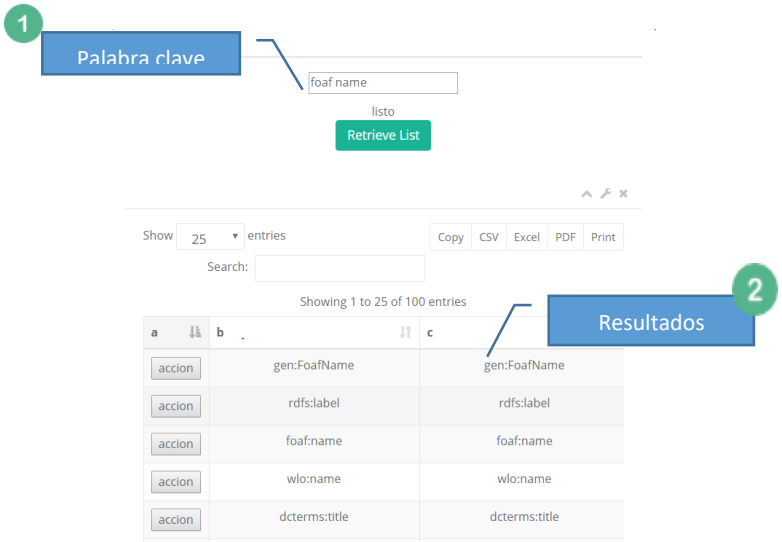
- **Sección 1:** Muestra información del usuario activo dentro de la aplicación, cada usuario tiene un límite de tiempo dentro de la aplicación por inactividad en el sitio de 5 minutos.
- **Sección 2:** Muestra las diferentes funcionalidades de la aplicación.
- **Sección 3:** Se lista los proyectos generados por el usuario, dentro de esta sección se puede ver y eliminar un determinado proyecto.

3.1.2.1. Herramienta transformación de datos

- **Selección de archivo :** El usuario debe ingresar un archivo CSV delimitado por comas para poder seguir con las funcionalidades que brinda la aplicación

- **Selección de términos:** Al cargar el archivo CSV se muestra al usuario la siguiente interfaz la cual está compuesta por tres áreas importantes:
    - **Sección 1:** Se establece una URI válida para cada proyecto, esta URI puede ser editada por el usuario de acuerdo a sus necesidades.

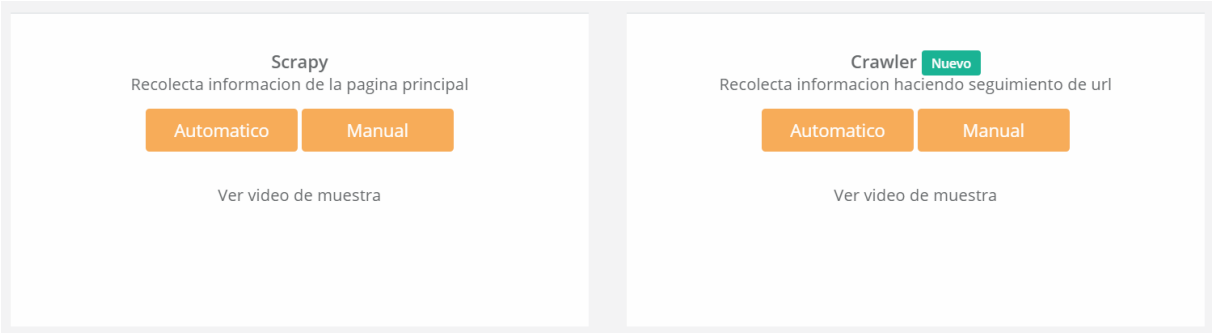
    - **Sección 2:** Cada columna representa un tipo de dato dentro del lenguaje RDF se establecen cuatro tipos “texto, entero, fecha y recurso”; el usuario puede seleccionar un tipo de dato por cada columna.
    - **Sección 3**: En esta sección el usuario puede buscar vocabularios existentes al hacer clic en “buscar termino aparece la siguiente interfaz.



Generar RDF: Para generar las diferentes serializaciones RDF debe hacer clic en “generar RDF” dentro de las serializaciones se establece 7 formatos existentes como se muestra en la siguiente figura.

3.1.2.2. Herramienta recolección de datos

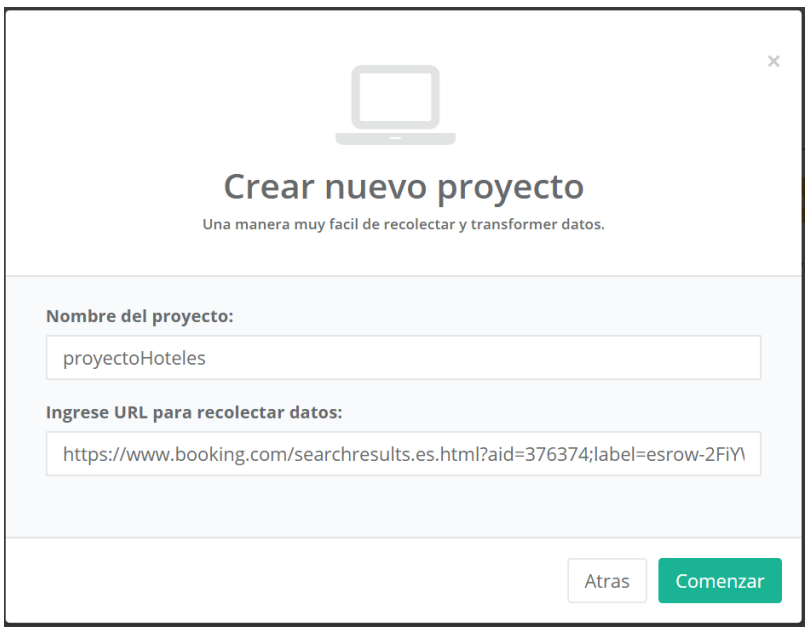
Dentro de este apartado el usuario puede seleccionar las herramientas que mejor se acoplen a sus necesidades; actualmente existen dos tipos de extracción de datos categorizados de la siguiente manera:



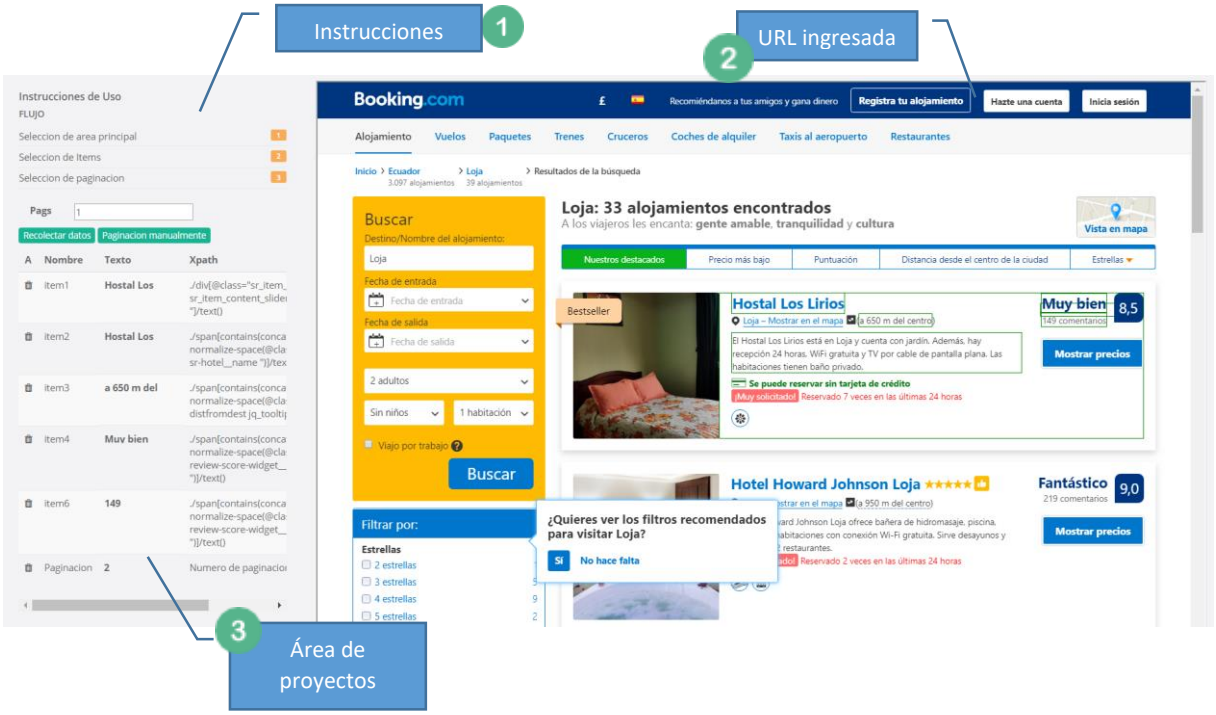
- **Scrapy:** Herramienta capaz de extraer la información de la página principal teniendo en cuenta la paginación de cada sitio.
  Se puede realizar el proceso de dos maneras, dependiendo del sitio que se requiere extraer la información.

Proceso de recolección Automático

Creación del proyecto: En la primera forma “Automático” el usuario tiene que ingresar el nombre del proyecto y la URL del sitio que requiere recolectar datos, para este manual se trabaja con datos de hoteles que brinda booking y clic en comenzar.

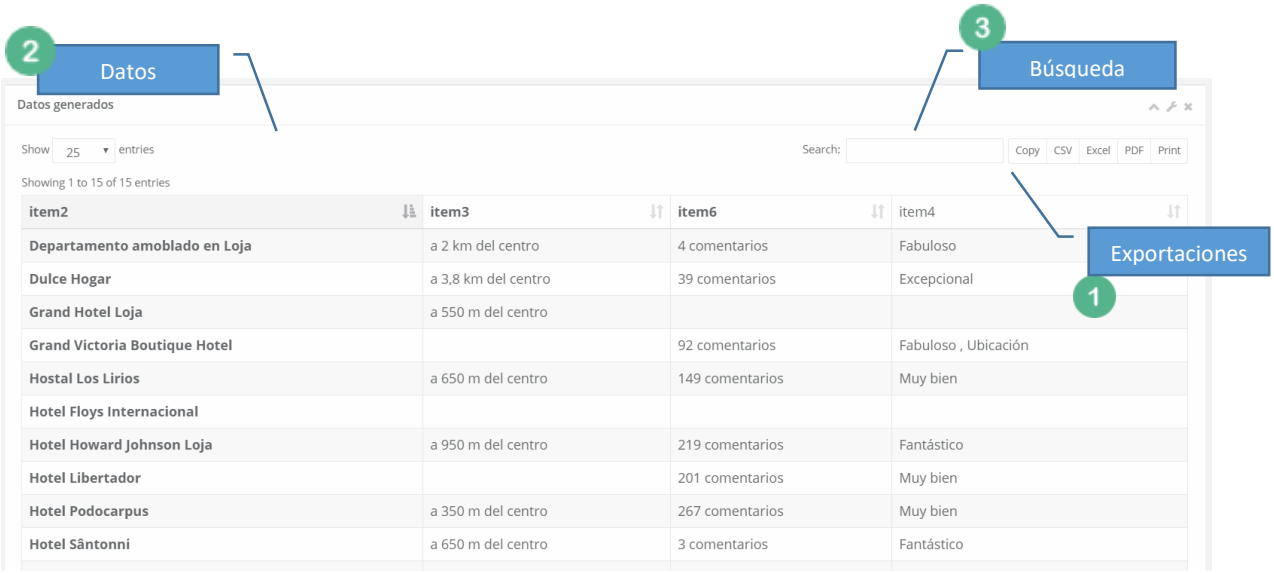


Recolección de datos: Aparece en pantalla el área de trabajo de la recolección de datos, esta sección está dividida en tres partes importantes que ayudan a la mejora de la extracción de datos.



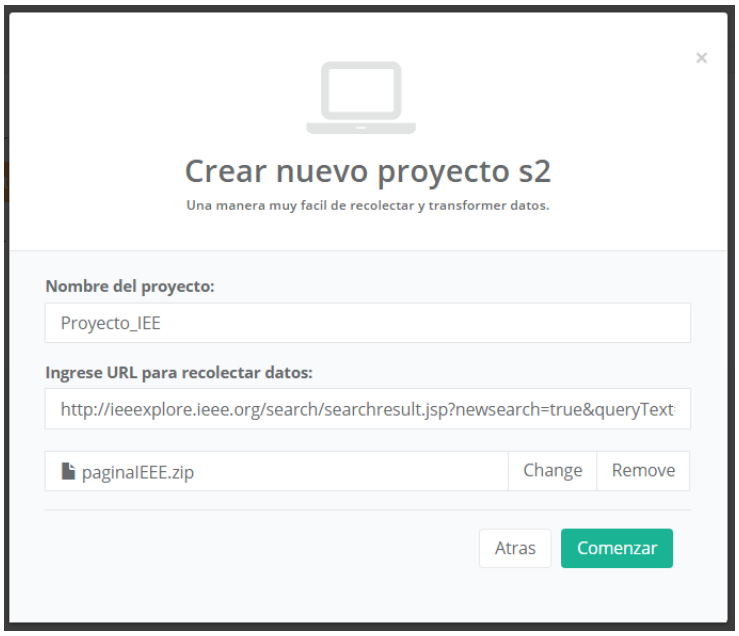
- **Sección 1:** Se describen las instrucciones de uso para tener una recolección de datos exitosa
  - **Selección de área principal:** El usuario tiene que tener en cuenta el área que contiene la mayor información haciendo clic dentro de la sección.
  - **Selección de ítems:** El usuario debe seleccionar los ítems que requiere extraer, además puede modificar el nombre de la columna en la sección 3.
  - **Selección de paginación:** El usuario identifica la paginación para recolectar datos, se tiene en consideración el límite de páginas que el usuario ingrese.
- **Sección 2:** Se refleja la URL ingresada para que el usuario seleccione los ítems que requiere extraer teniendo en cuenta el proceso de selección.
- **Sección 3:** En esta sección se verifica el texto que se extraerá dentro de la recolección de datos.

Una vez seleccionados todos los campos en usuario puede recolectar los datos haciendo clic en “recolectar datos” como resultado se tiene los datos estructurados de la página web.



Proceso de recolección Manual

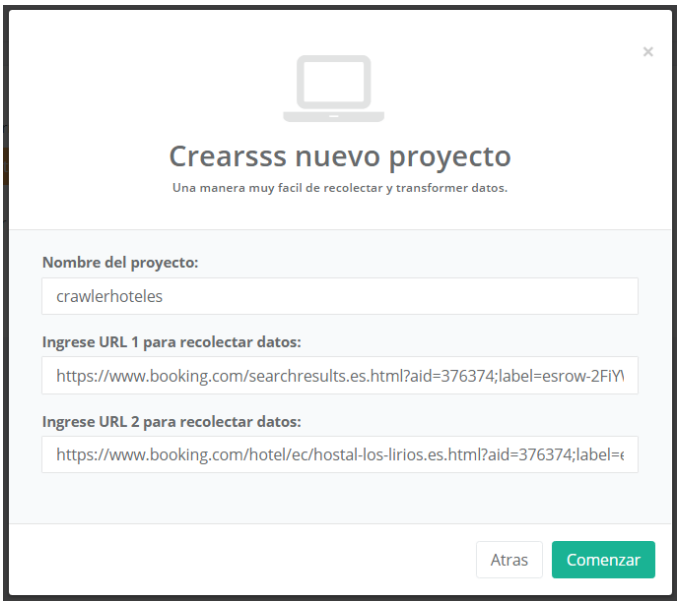
Esta sesión admite subir archivos .zip en la que contiene la página web y los recursos necesarios para poder cargar la página, el proceso a continuación es igual al proceso automático ver sección anterior.



- **Crawler:** Esta herramienta realiza el seguimiento de enlaces es decir ingresa a una URL específica y extrae la información de la página de un segundo nivel.

Proceso de recolección Automático

El usuario debe ingresar la URL de página 1 en la cual contienen el área principal de las URLs a ingresar y la URL 2 para la selección de ítems.



A continuación se despliega la interfaz de trabajo para la recolección de datos a diferencia de scrapy se muestra dos pestañas en al que el usuario requiere seguir una serie de instrucciones

Instrucciones:

- Selección de enlace
- Selección de paginación
- Selección de ítems
- Clic en recolectar información

Anexo 11 Manual de programador

Manual de programador

Proyecto DATAIMPORT

Autor: Ana Cristina Cardenas

Versión: 001

Fecha: 05/02/2018

1. Introducción

El propósito de este manual es dar continuidad al proyecto, expandiendo el alcance hasta llegar a un prototipo estable dentro de la investigación.

2. Aspectos generales

Actualmente el código fuente de la aplicación se encuentra alojado dentro del repositorio de la UTPL “git taw” en la dirección: *https://git.taw.utpl.edu.ec/accardenas4/proyecto_tt.git*

3. Componentes

Las configuraciones generales como urls estáticas y configuración de base de datos del proyecto se encuentran en los archivo “***config.php y manejar_conexion.php” .***

Dentro del proyecto se enceuntran archivos de configuración necesarios, a continuación se realiza una breve descripción de los archivos del proyecto

Capas		
Capa de presentación	**Capa de lógica utilizada**	**Descripción**
Index.php	-guardar_registro.php -ingresar_sistema.php -salir.php	Contiene el código fuente de la pantalla principal de la aplicación incluye registro y logueo.
Proyectos_usuario.php	-class_mysql.php -salir.php	Contiene el código fuente del área de trabajo que lista los proyectos por cada usuario
Herramientas_datos.php	-geston_url.php -gestion_url_manual.php -gestion_url_crawler.php -gestion_url_crawler_manual.php -salir.php	Contiene el código fuente de la pantalla para seleccionar las herramientas establecidas dentro de la aplicación
Mailbox.php	-scraper.php -salir.php	Contiene el código fuente de la pantalla de selección de configuración.

Mailbox_crawler.php	-scraper.php -salir.php	Contiene el código fuente de la pantalla de selección de configuración acoplando a necesidades crawler.
Data_user.php	-class_mysql.php -salir.php	Contiene código fuente de la aplicación que lista la data recolectada por el usuario
Herramientas_ld.php	-rdf.php	Contiene el código fuente para realizar el mapeo del CSV a las diferentes serializaciones RDF

En la capa de presentación de los archivos mailbox.php y mailbox_crawler.php se realiza el script para generar las expresiones XPATH seleccionadas por el usuario.

Detalle de archivos de la capa lógica

- **Guardar_registro.php:** Se establece una conexión al archivo manejar_conexion.php para insertar en la base de datos el usuario que se registra en la aplicación.
- **Ingresar_sistema.php:** Se establece una conexión al archivo manejar_conexion.php para consultar en la tabla User y validar las credenciales del usuario, además se levantan variables de sesión del usuario.
- **Class_mysql.php:** Contiene las consultas hacia la base de datos.
- **Salir.php:** Se destruyen las variables de sesión activas por el usuario.
- **Gestión_url.php:** Contiene la gestión de archivos del lado del servidor y el uso de exec para ejecutar “wget” para bajar el contenido HTML con los siguientes parámetros.

Comando	Descripción
--no-directories	No crea jerarquía de directorios.
--recursive	Descarga recursivamente los archivos.
--page-requisites	Descarga archivos necesarios para que la página funcione correctamente.
--html-extension	Retorna el archivo como un .HTML

--convert-links	Convierte enlaces locales en hipervínculos visibles.
--user-agent	Envía encabezados a las solicitudes HTTP.

- **Sraper.php:** Contiene código para establecer comunicación con la terminal ejecutando un archivo .py mediante exec (), se establece un entorno virtual y se aloja las diferentes herramientas necesarias para realizar la extracción de datos.

Tecnología utilizada

Se utilizó un entorno virtual para alojar selenium y scrapy como requerimientos dentro del servidor se tiene:

Componente	Versión
asn1crypto	**0.24.0**
attrs	**17.4.0**
Automat	**0.6.0**
cffi	**1.11.4**
Constantly	**15.1.0**
Cryptography	**2.1.4**
Cssselect	**1.0.3**
enum34	**1.1.6**
Hyperlink	**17.3.1**
Idna	**2.6**
incremental	**17.5.0**
Ipaddress	**1.0.19**
Lxml	**4.1.1**
Parsel	**1.3.1**
pyasn1	**0.4.2**
pyasn1-modules	**0.2.1**
Pycparser	**2.18**
PyDispatcher	**2.0.5**
pyOpenSSL	**17.5.0**
Queuelib	**1.4.2**
Scrapy	**1.5.0**
Selenium	**3.8.1**
service-identity	**17.0.0**

Six	**1.11.0**
w3lib	**1.18.0**
zope.interface	**4.4.3**