

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE
MÁSTER EN ESTADÍSTICA COMPUTACIONAL Y CIENCIA DE
DATOS PARA LA TOMA DE DECISIONES



O-Metrics: Herramienta Web
para la Visualización de la
Evolución de Repositorios de
Ontologías

Autor:
Miguel Antonio Linares Santana

Tutor:
Manuel Quesada Martínez

TRABAJO FIN DE MÁSTER

Septiembre - 2023

Agradecimientos

En primer lugar, quiero expresar mi más profundo agradecimiento a Dios, quien ha sido mi guía constante, proporcionándome la fortaleza y la perseverancia para superar cada obstáculo en el camino hacia la culminación de este Máster. Ha sido mi roca y mi refugio, y su amor y misericordia han sido el cimiento sobre el cual he construido cada éxito.

En segundo lugar, no tengo palabras suficientes para agradecer a mis queridos padres, quienes han sido una fuente inagotable de apoyo incondicional. Su fe inquebrantable en mis habilidades me ha brindado el aliento necesario para perseguir mis sueños y alcanzar metas significativas.

Además, quiero reconocer con especial gratitud a mi gran amigo Gregory Durán. Tu presencia en mi vida ha sido una bendición; siempre estuviste allí para ofrecer tu apoyo, alentar mis esfuerzos y creer en mis capacidades, incluso en los momentos en que dudaba de mí mismo. Tu amistad y tu fe inquebrantable en mí han sido una fuente de inspiración y fortaleza que ha trascendido más allá de lo imaginable.

Por último, quisiera extender mi gratitud también a cada uno de los profesores y compañeros que han formado parte de este camino. Sus consejos, conocimientos y experiencias compartidas han enriquecido este período de aprendizaje, brindándome herramientas valiosas para enfrentar los desafíos profesionales que vendrán.

Resumen

Las ontologías han emergido como una herramienta esencial en diversas disciplinas. Las ontologías son almacenadas en repositorios accesibles a través de Internet, por lo que su estudio y el aseguramiento de su calidad se ha convertido en un aspecto interesante; y para el que la Ciencia de Datos tiene el potencial de revolucionar la forma de mantener este tipo de recursos de conocimiento.

En este trabajo se abordará el desarrollo de una metodología y su implementación como una plataforma web de visualización, O-Metrics, y que usando herramientas de Ciencia de Datos proporcionará una solución automatizada y eficiente para rastrear y visualizar la evolución de los repositorios de ontologías. El lenguaje de desarrollo será Python. Sobre él se integrarán tecnologías de procesamiento y visualización de datos como: el API de GitHub para inferir información de la evolución ontologías, los Servicios Web de OQuaRE para el procesamiento y obtención de métricas, o librerías como Flask, HTML5/CSS y Plotly para el desarrollo del *dashboard* como herramienta web.

Una vez implementada, se ha aplicado O-Metrics sobre el repositorio OBO Foundry, que en septiembre de 2023, agrupa a 257 ontologías. A través de un riguroso proceso de filtrado, nos centramos en un grupo seleccionado de 170 ontologías respaldadas por archivos OWL y mantenidas activamente en GitHub. Este enfoque ha permitido una categorización en quince dominios distintos, brindando una visión tanto cuantitativa como cualitativa del campo, y poniendo al descubridor áreas prometedoras para futuras investigaciones.

Palabras clave: ontologías, ciencia de datos, O-Metrics, OBO Foundry.

Índice

1. Introducción	1
1.1. Justificación y antecedentes	1
1.1.1. Ciencia de Datos para analizar ontologías	4
1.2. Objetivos del estudio	6
1.2.1. Objetivo General	6
1.2.2. Objetivos Específicos	7
1.3. Hipótesis	7
1.4. Metodología	8
2. Materiales y Métodos	10
2.1. Estado del arte	10
2.1.1. Definiciones de Ontología	10
2.1.2. Evolución de ontologías en el tiempo	13
2.1.3. Librerías y Servicios Web para analizar ontologías	15
2.1.4. Acceso a ontologías y sus versiones	16
2.1.5. Caracterización de una ontología mediante métricas	21
2.1.6. Herramientas para la visualización de datos	28
2.2. Selección de lenguajes de programación para O-Metrics	31
2.2.1. Librerías para el procesamiento y visualización de datos . .	34
2.2.2. Procesamiento de las ontologías y extracción de métricas .	37
2.3. Arquitectura de la solución a desarrollar	38

3. Resultados	40
3.1. Planificación del proyecto	40
3.2. Diseño del software e implementación de O-Metrics	43
3.2.1. Requisitos y casos de uso	43
3.2.2. Estructura de la Base de Datos	45
3.2.3. Integración y Extensibilidad del Sistema	47
3.3. Descripción del repositorio analizado: OBOFoundry	48
3.4. Ejemplo de uso de O-Metrics	51
3.4.1. Inferfaz para la recolección de Datos	51
3.4.2. Dashboards principales: actividad y métricas	53
3.5. Código fuente	65
3.6. Comparación con los métodos tradicionales	66
4. Conclusiones y Trabajo Futuro	68
4.1. Discusión y limitaciones	70
4.2. Trabajo futuro	72

Índice de figuras

1.	Captura de pantalla de BioPortal	2
2.	Captura de pantalla de OBO Foundry	3
3.	Ejemplo de la ontología GALEN (extraída de [59])	12
4.	Captura de pantalla de la información resumen de la ontología sobre Consentimiento Informado en BioPortal	13
5.	Ejemplo de Ontologías en archivo JSON	18
6.	Ilustración de los “commits” en GitHub	20
7.	Principios fundamentales de OBO Foundry	23
8.	Clasificación de “pitfall” (extraída de [46])	24
9.	Solicitud HTTP POST desde Postman	26
10.	Archivos recibidos por la solicitud	27
11.	Dashboard sobre la tasa RSMR de pacientes con sepsis (I)	29
12.	Dashboard sobre la tasa RSMR de pacientes con sepsis (II)	30
13.	Dashboard de VISUEL (extraída de [3])	31
14.	Dashboard GNPS (extraída de [44])	32
15.	Arquitectura de la Solución Propuesta	38
16.	Diagrama Metodología Scrum	40
17.	Cronograma de las distintas fases	43
18.	Estructura de Base de Datos	46
19.	Interfaz de Usuario para la Gestión de Base de Datos de Repositorios	52
20.	Interfaz para la Verificación de Datos de Repositorios	53

21.	Interfaz para la Incorporación de Métricas a través del OQuaRE Metrics Web Service	54
22.	Paneles de Visualización para Actividad del Repositorio (izquierda) y Ontologías (derecha)	55
23.	Informaciones generales sobre los repositorios	56
24.	Leyenda de Gráfico de Dona	57
25.	Actividad General de Repositorios	58
26.	Gráfico de Barras con filtros	59
27.	Filtros de Leyenda Aplicados	60
28.	Actividad de archivos OWL	60
29.	Métricas Básicas de Ontologías	61
30.	Métricas Básicas de Ontologías	62
31.	Métricas OQuaRE por versión	62
32.	Salud de la Ontología	63
33.	Métricas OQuaRE con las Últimas Versiones	63
34.	Métricas OQuaRE por Dominio	64
35.	Captura del repositorio de Bitbucket de O-Metrics	65

Índice de tablas

1.	Listado de Dominios	50
----	-------------------------------	----

1. Introducción

Las ontologías han emergido como una herramienta esencial en diversas disciplinas, especialmente en áreas como la Inteligencia Artificial (IA), la Ciencia de Datos, y la Informática Biomédica [56]. Existen ontologías para cubrir diferentes ámbitos y creadas por expertos de dominios concretos. Aunque profundizaremos en su definición posteriormente, las ontologías proporcionan una base para la representación de conocimientos en la Web, permitiendo la integración de información y la interoperabilidad entre sistemas y aplicaciones. En la actualidad, su uso se ha extendido a diferentes ámbitos, por lo que su estudio y el aseguramiento de su calidad se ha convertido en un aspecto interesante para investigadores e investigadoras.

1.1. Justificación y antecedentes

Las ontologías pueden tener múltiples usos como la representación estructurada de conocimiento, la integración de Información, la Web Semántica, la especificación y la comunicación genérica de conocimientos de un dominio o la organización de los significados de los términos [37, 6, 4]. En el ámbito de la Bioinformática, las ontologías juegan un papel crucial para representar y organizar el conocimiento biológico y médico [5] y para la integración y análisis de datos a gran escala en otras disciplinas. Por ejemplo, las ontologías son fundamentales para el Procesamiento del Lenguaje Natural y la Extracción de Información [14], permitiendo a las máquinas entender y responder a consultas en lenguaje natural

de forma más precisa y coherente. Las ontologías han pasado a ser una herramienta clave en la Ciencia de Datos y en otras disciplinas, aunque aún quedan desafíos significativos en su desarrollo y gestión que se deberán abordar en los próximos años.

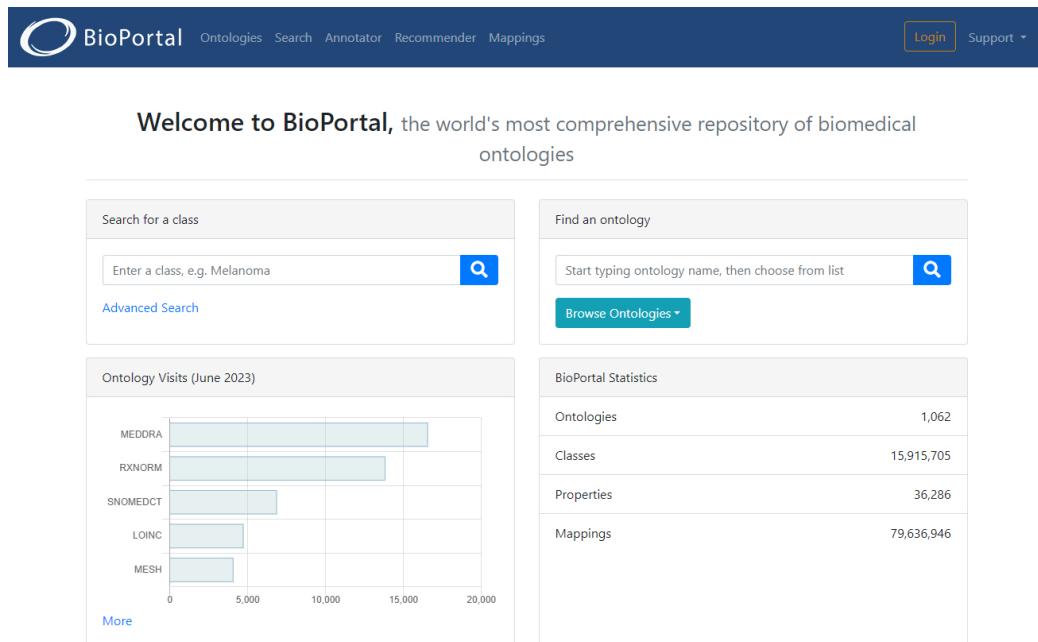


Figura 1: Captura de pantalla de BioPortal

Como artefactos software, las ontologías pasaron de ser diagramas de conceptos físicos a ser complejos archivos digitales los cuales han pasado a almacenarse en repositorios en linea a través de Internet. Por ejemplo, BioPortal [38] se define como el repositorio de ontologías biomédicas más completo del mundo, y provee de acceso a cientos de ontologías que abarcan una amplia gama de temas. En la Figura 1 se muestra la página principal de BioPortal (en julio de 2023) en la que

se aprecian algunas cifras relacionadas con su actividad y relevancia. En dicho instante, este repositorio contenía 1062 ontologías compuestas aproximadamente por 16 millones de clases, 36 mil propiedades y 80 millones de mappings.

Existen otras alternativas a BioPortal, entre las que destacamos el OBO Foundry [56]. La comunidad del OBO (*Open Biological and Biomedical Ontology*) busca el desarrollo comunitario de ontologías interoperables para las Ciencias Biológicas. Para ello, promueven el seguimiento de principios de buenas prácticas que se pueden consultar en su web, y que deberían ser utilizados por todos los creadores de ontologías que quieran pertenecer a esta comunidad. En la Figura 2 se muestra la captura de pantalla de una parte de las ontologías disponibles en el OBO Foundry. En agosto de 2023, el repositorio contenía 181 ontologías distribuidas bajo licencia Creative Commons y agrupadas en diferentes dominios.

OBO Library: find, use, and contribute to community ontologies						
Download table as: [YAML JSON-LD RDF/Turtle]						
Search Table						
<input type="text"/> Search table ...						
Ontology Domains:	<input checked="" type="checkbox"/> Group By Domain	<input checked="" type="checkbox"/> Hide Inactive	<input checked="" type="checkbox"/> Hide Obsolete			
Upper						
ID ^	Title ^	Description	Quick Access	Re-Use ^	Social	
bfo	Basic Formal Ontology	The upper level ontology upon which OBO Foundry ontologies are built.				Stars 223
cob	Core Ontology for Biology and Biomedicine	COB brings together key terms from a wide range of OBO projects to improve interoperability.				Stars 30
ro	Relation Ontology	Relationship types shared across multiple ontologies				Stars 77
Anatomy and development						
ID ^	Title ^	Description	Quick Access	Re-Use ^	Social	
aism	Ontology for the Anatomy of the Insect SkeletoMuscular system (AISM)	The AISM contains terms used in insect biodiversity research for describing structures of the exoskeleton and the skeletomuscular system. It...				Stars 10
amphx	The Amphioxus Development and Anatomy Ontology	An ontology for the development and anatomy of Amphioxus (<i>Branchiostoma lanceolatum</i>).				Stars 3
bspo	Biological Spatial Ontology	An ontology for representing spatial concepts, anatomical axes, gradients, regions, planes, sides, and surfaces				Stars 11

Figura 2: Captura de pantalla de OBO Foundry

A medida que las ontologías crecen en tamaño y complejidad, mantenerlas y actualizarlas se vuelve cada vez más desafiante [35]. A pesar de los avances en las herramientas y metodologías para el desarrollo de ontologías, aún persisten dificultades en la representación precisa de conceptos complejos y en la resolución de problemas de homogeneidad y coherencia [17]. El uso de sistemas de versiones como GitHub permiten analizar estos artefactos desde el punto de vista de su evolución en el tiempo, sin embargo no encontramos soluciones completas que exploten o visualicen información relativa a su evolución. Por ejemplo, repositorios como los dos anteriores mencionados, únicamente proveen de reportes básicos cada cierto tiempo.

1.1.1. Ciencia de Datos para analizar ontologías

La Ciencia de Datos es una disciplina que utiliza métodos estadísticos, algoritmos y modelos para extraer conocimiento de grandes conjuntos de datos. Esta disciplina ha tenido un impacto significativo en una amplia gama de industrias, desde la Salud hasta la Banca, y tiene el potencial de revolucionar la manera en que mantenemos y actualizamos ontologías complejas. Un ejemplo de esto lo podemos ver en [48].

En la actualidad podemos ver ejemplos de trabajos que han utilizado la Ciencia de Datos para el mantenimiento de ontologías concretas. Por ejemplo:

- En 2015, un equipo de investigadores de la Universidad de Stanford [34] utilizó técnicas de aprendizaje automático para identificar términos que se utilizan con frecuencia en la ontología de la medicina clínica (SNOMED

CT). Esta información se utilizó para mejorar la precisión y la coherencia de la ontología.

- En 2016, un equipo de investigadores de la Universidad de Oxford [18] utilizó técnicas de visualización para crear un mapa de la ontología de los datos de la investigación biomédica (Bio2RDF). Este mapa ayudó a identificar áreas de la ontología que necesitaban ser mejoradas.

Estos son sólo dos ejemplos de los muchos trabajos que han utilizado la Ciencia de Datos para el mantenimiento de ontologías. Esta disciplina tiene el potencial de revolucionar la manera en que mantenemos y actualizamos ontologías complejas, y conducir a ontologías que son más precisas, coherentes y fáciles de usar. Por tanto, creemos que la Ciencia de Datos tiene el potencial de revolucionar la manera en que mantenemos y actualizamos ontologías complejas. Esta disciplina puede ayudarnos a identificar patrones y tendencias que no serían visibles a simple vista, y también puede ayudarnos a visualizar los datos de una manera que sea fácil de entender. Esto puede conducir a ontologías que son más precisas, coherentes y fáciles de usar.

Desde el punto de vista de los repositorios de ontologías, creemos que es interesante trabajar sobre la idea de crear un proceso automatizado que nos permita obtener métricas de los repositorios y las ontologías que son parte de ellos, y que estas métricas puedan ser analizadas con un eje temporal a modo de evolución. Por ello, se tratará en este Trabajo Fin de Máster el desarrollo de una herramienta que permita procesar de forma semiautomática diferentes aspectos relacionados con

ontologías públicamente accesibles a través de Internet. Este trabajo permitiría cubrir la necesidad de una herramienta que se ocupe de este vacío, proporcionando una solución automatizada y eficiente para rastrear y visualizar la evolución de los repositorios de ontologías, las ontologías individuales y sus características asociadas. Tal herramienta no solo simplificaría la tarea de rastrear la evolución de las ontologías, sino que también abriría nuevas oportunidades para el análisis y el estudio de las ontologías y los campos de conocimiento que representan.

1.2. Objetivos del estudio

1.2.1. Objetivo General

Nuestro objetivo es diseñar y desarrollar un proceso semiautomático que se encargue de proporcionar al usuario información relevante acerca de las tendencias y los patrones de comportamiento de las ontologías públicamente accesibles en repositorios. Asimismo, aspiramos a que dicho análisis ofrezca una herramienta valiosa para analizar la evolución de las ontologías a lo largo del tiempo, así como información sobre los principios de construcción de ontologías que son seguidos por sus diferentes autores. Este trabajo se enmarca en la búsqueda de proporcionar una herramienta web avanzada que permita automatizar la gestión y análisis de ontologías, facilitando así la labor de los usuarios y promoviendo la adhesión a las mejores prácticas de construcción de ontologías.

1.2.2. Objetivos Específicos

- Concebir y desarrollar un método de trabajo semiautomático con el propósito de trazar y describir de forma meticulosa las modificaciones entre distintas versiones de una ontología a partir del análisis y representación de elementos como los ‘commits’ realizados sobre sistemas de versionado.
- Concebir y desarrollar un método de trabajo semiautomático con el propósito de describir las ontologías a partir de diversos tipos de métricas seleccionadas a partir del análisis del estado del arte.
- Implementar y una herramienta en línea que pueda ser ofrecida a los usuarios para ejecutar los métodos desarrollados de forma sistemática y mantenida en el tiempo, permitiendo su uso como herramienta de aseguramiento de la calidad de este tipo de recursos.

1.3. Hipótesis

Basándonos en el análisis de la situación actual de las ontologías y las limitaciones existentes en las herramientas de estudio y visualización de las mismas, formulamos la siguiente hipótesis central para nuestra investigación:

“La implementación de una herramienta de procesamiento y visualización interactiva que utilice ontologías provenientes de repositorios como BioPortal o el OBO Foundry, permitirá una mejor comprensión y seguimiento de la evolución de las ontologías y sus características a lo largo del tiempo. Esta herramienta

semiautomática proporcionará insights valiosos y mejorará la eficiencia en los estudios de ontología, superando las limitaciones de los métodos manuales existentes”.

Esta hipótesis se establece con el objetivo de abordar una brecha significativa en el campo del estudio de las ontologías, donde a pesar de la abundancia de datos y la importancia del seguimiento de la evolución de las ontologías, no existen herramientas adecuadas para hacerlo. En este estudio, intentaremos explorar y validar esta hipótesis a través del diseño y desarrollo de una solución innovadora que facilite la gestión y análisis de las ontologías y los repositorios donde son alojadas.

1.4. Metodología

Este estudio propone un enfoque sistemático y detallado para analizar la evolución de las ontologías, utilizando una metodología que integra herramientas semiautomáticas y técnicas de visualización de datos. Nuestro enfoque no solo se centrará en el desarrollo de esta metodología, sino también en el desarrollo de una herramienta que la implemente y su aplicación con datos de repositorios reales, buscando así demostrar su aplicabilidad en el mundo real. La robustez de la metodología se medirá en función de su capacidad para detectar y visualizar tendencias y patrones en la evolución de las ontologías de una manera que sea simple, resumida y útil para los usuarios.

Nuestro proyecto abarca una extensa metodología orientada a proporcionar una visión comprensiva y detallada de la evolución de las ontologías en el tiem-

po. Este proceso consta de una serie de pasos meticulosamente diseñados para garantizar un análisis profundo y preciso, cada uno de los cuales contribuye a la creación de una herramienta robusta y eficiente.

- **Análisis del estado del arte:** análisis del estado del arte relacionado con los fundamentos de las ontologías y herramientas de Ciencia de Datos para el procesamiento y visualización de ontologías.
- **Selección de repositorios a explotar:** análisis de repositorios de ontología y su viabilidad para ser explotados. Las ontologías deben ser accesibles y con su código fuente disponible para obtener y procesar sus datos.
- **Diseño de metodología para caracterizar ontologías:** a partir del análisis del estado del arte y de la selección del repositorio se diseñará una estrategia para caracterizar las ontologías desde su análisis y extracción de métricas.
- **Selección de lenguajes de programación para la Ciencia de Datos:** seleccionar los lenguajes de programación y las librerías adecuadas que permitan procesar y obtener de forma automática datos de las ontologías que se desean caracterizar.
- **Diseño e implementación de herramienta:** se implementará una Herramienta Web que permite la ejecución de la metodología desarrollada de forma sistemática. La herramienta deberá ser dinámica y permitir su ejecutada en un futuro actualizando dinámicamente sus datos. La herramienta permitirá tanto el procesamiento como la visualización de los datos.

2. Materiales y Métodos

2.1. Estado del arte

En este capítulo describiremos la solución de los distintos métodos y trabajos relacionados con nuestra propuesta, contextualizando nuestra solución y integrando en ella aspectos relevantes para la comunidad.

2.1.1. Definiciones de Ontología

Las ontologías, pese a no ser un término muy conocido por el público general, es un término que ha adquirido cierta relevancia dentro del ámbito científico durante las últimas décadas. Su origen proviene de la filosofía, donde distintos autores ofrecen una definición ligeramente diferente de este término [27]. Para algunos las ontologías son “*...la ciencia de lo que es (α), para otros de lo que hay (β), y para otros es un sinónimo de «metafísica» (γ)*”. Una definición que resume y sintetiza estas tres líneas de pensamientos es la que propone [55], la cual dice que las ontologías son “*la ciencia de lo que es, de los tipos y estructuras de objetos, propiedades, eventos, procesos, y relaciones en cada área de la realidad*”.

En 1991, Neches et al. explicaron la necesidad de permitir compartir y reutilizar cuerpos de conocimiento de forma computacional para desarrollar sistemas grandes basados en el conocimiento y más complejos [36]. Según la definición de Neches, “*una ontología define los términos y relaciones básicas que componen el*

vocabulario de un área temática, así como las reglas de combinación de términos y relaciones para definir extensiones del vocabulario”. Este fue el primer intento de utilizar ontologías en el campo de la Inteligencia Artificial (IA). Poco a poco estos conceptos fueron ganando popularidad en el ámbito de la Informática y la Ciencia de Datos.

Una de las definiciones de ontología más referenciadas en IA, habiendo recibido 23337 citas en Google Scholar en julio de 2023 es la de Gruber [15]: “*Una ontología es una especificación explícita de una conceptualización*”, siendo una conceptualización una abstracción o vista simplificada del mundo que queremos representar. Sin embargo, las ontologías como artefactos de ingeniería en IA requieren una definición más extensa que la proporcionada por Gruber, por lo que en 1998, Guarino [16] define una ontología como:

“En sentido filosófico, podemos referirnos a una ontología como un sistema particular de categorías que dan cuenta de una determinada visión del mundo. Como tal, este sistema no depende de un lenguaje concreto: la ontología de Aristóteles es siempre la misma, independientemente del lenguaje utilizado para describirla. Por otra parte, en su uso más extendido en IA, una ontología se refiere a un artefacto de ingeniería, constituido por un vocabulario específico utilizado para describir una determinada realidad, más un conjunto de suposiciones explícitas sobre el significado previsto de las palabras del vocabulario. Este conjunto de suposiciones suele tener la forma de una teoría lógica de primer orden, en la que las palabras del vocabulario aparecen como nombres de predicados unarios o binarios, denominados respectivamente conceptos y relaciones. En el

caso más sencillo, una ontología describe una jerarquía de conceptos relacionados por relaciones de subsunción; en casos más sofisticados, se añaden axiomas adecuados para expresar otras relaciones entre conceptos y restringir su interpretación prevista.”

A modo de ejemplo, en la Figura 3 se muestra un extracto de la ontología Galen. En la parte superior izquierda se aprecia la jerarquía de conceptos. En la parte derecha la definición de la clase *Surgeon* a partir de otros conceptos y relaciones. Para más información, se recomienda al lector el trabajo [60].

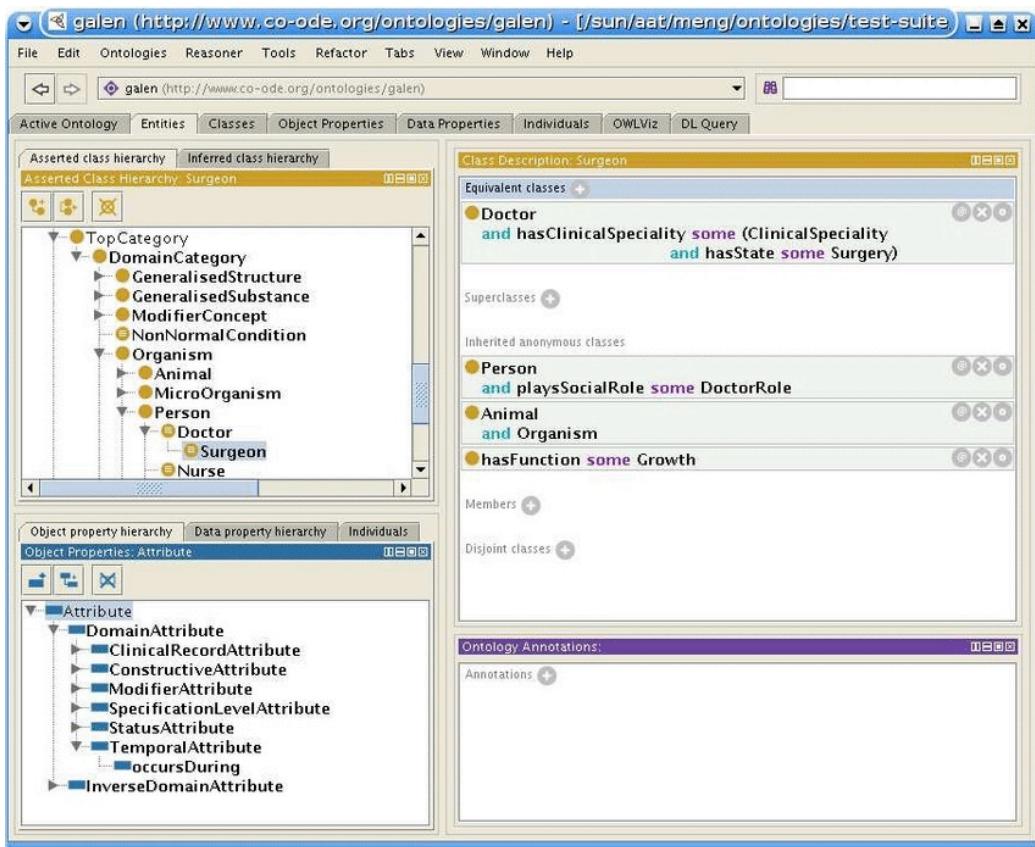


Figura 3: Ejemplo de la ontología GALEN (extraída de [59])

2.1.2. Evolución de ontologías en el tiempo

Con el transcurso del tiempo, las ontologías han experimentado evoluciones significativas debido a múltiples factores, entre ellos el surgimiento de nuevos conocimientos científicos, las necesidades cambiantes de los usuarios y la continua refinación y estandarización de las prácticas de construcción y mantenimiento de ontologías.

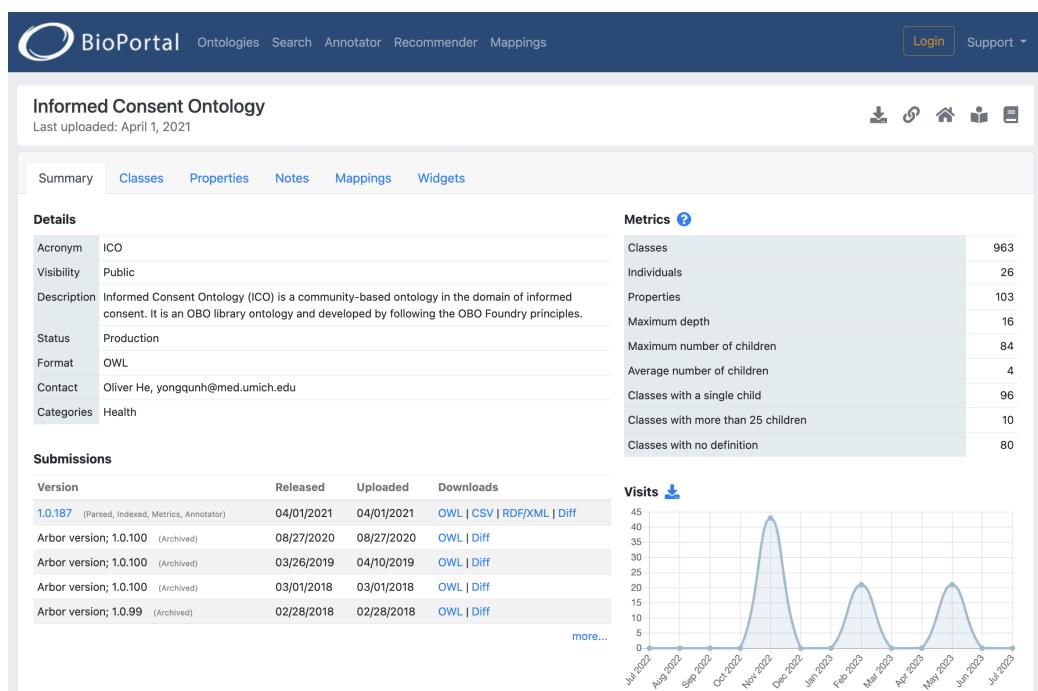


Figura 4: Captura de pantalla de la información resumen de la ontology sobre Consentimiento Informado en BioPortal

En la Figura 4 se muestran, como ejemplo, las métricas y estadísticas ofrecidas por BioPortal sobre una de sus ontologías: *Informed Consent Ontology*. En la parte inferior izquierda ofrece un registro de todas las versiones almacenadas. La

única información que ofrece sobre la diferencia entre las versiones es un pequeño resumen (accesible desde el botón *Diff*), entre versiones consecutivas, y realizado con la herramienta Bubastis [29]. En la parte superior derecha podemos ver una tabla de métricas descriptivas, pero únicamente aplicada a la última versión de la ontología. Por último, en la parte inferior derecha se ofrecen estadísticas relacionadas con las veces que esta ontología ha sido visitada.

El seguimiento de la evolución de las ontologías no solo aporta información valiosa sobre la dinámica y los patrones de cambio de las ontologías en sí, sino que también proporciona una visión más amplia de cómo evolucionan los campos de conocimiento representados por estas ontologías. Es importante entender esta evolución ya que ayuda a comprender cómo los sistemas de conocimiento pueden cambiar y adaptarse con el tiempo. Este entendimiento puede requerir del desarrollo o ampliación de herramientas y métodos para el análisis de ontologías y los repositorios que las contienen.

En nuestro caso, centraremos la atención en trazar y describir de forma meticulosa las modificaciones entre distintas versiones de una ontología. Explorar la evolución de las ontologías se sitúa en el corazón de nuestro estudio y proporciona el marco para entender cómo las ontologías cambian en un entorno en constante evolución. En resumen, en este trabajo analizaremos los cambios que sufren las ontologías de un determinado repositorio, a lo largo del tiempo, desde dos puntos de vista:

- **Actividad de las ontologías:** se analizarán información relacionada con las veces que una ontología es actualizada en un repositorio en el tiempo.

- **Cambios estructurales de las ontologías:** se analizarán métricas específicas de las ontologías que permitan conocer como ha cambiado su contenido.

2.1.3. Librerías y Servicios Web para analizar ontologías

A pesar de las propuestas anteriores, en el panorama tecnológico actual, no existen herramientas especializadas que se dediquen a visualizar y rastrear la evolución de repositorios de ontologías, las ontologías en sí mismas y las características inherentes a ellas. Esta brecha en el mercado de herramientas tecnológicas es intrigante, dado que un recurso de este tipo podría aportar un valor inmenso a aquellos que buscan llevar a cabo un estudio general y sistemático de las ontologías y observar cómo han evolucionado y cambiado a lo largo de los años.

El universo de herramientas y plataformas destinadas a facilitar la extracción y el análisis de datos relacionados con la evolución de las ontologías es amplio y diversificado. A pesar de esta riqueza, la implementación eficaz de dichas herramientas a menudo presupone una extensa familiaridad técnica y un compromiso considerable en términos de inversión temporal. Por ejemplo, la biblioteca OwlReady2 [41] escrita en Python permite el procesamiento y manipulación de ontologías y su integración con scripts de Ciencia de Datos. Sin embargo, dicha biblioteca tiene limitaciones ya que no permite la explotación de las ontologías de forma tan avanzada como OWLAPI [40]; OWLAPI es compatible con el lenguaje compilado Java, lo que limita su integración con Python.

Un desafío persistente radica en la integración y cohesión de distintas herramientas en una solución común. Aunque cada herramienta puede ser poderosa en

su dominio específico, la creación de una solución comprensiva y coherente exige la superación de barreras técnicas y conceptuales. Para evitar estos problemas de compatibilidad, la digitalización y el surgimiento de Servicios Web ha abierto nuevas puertas para la obtención e integración de datos de datos. Según [20], un Servicio Web es un software que puede mantener una comunicación máquina a máquina a través de la red, en nuestro caso esta red es el Internet. En el ámbito de las ontologías merece la pena destacar el trabajo [2] desarrollado por investigadores de la Universidad de Murcia, que ofrecen un servicio disponible en línea para calcular métricas detalladas de ontologías. La convergencia de estas distintas tecnologías, cuando se maneja adecuadamente, tiene el potencial de revolucionar la forma en que abordamos y entendemos la evolución y la gestión de las ontologías.

2.1.4. Acceso a ontologías y sus versiones

La tarea de rastrear manualmente la evolución de las ontologías a través de diversas versiones y actualizaciones puede ser extremadamente tediosa y laboriosa. Incluso con las herramientas existentes, la extracción manual de datos a gran escala no es práctica ni eficiente. Además, las herramientas actuales pueden no ser capaces de proporcionar la profundidad y el detalle requeridos para un análisis realmente significativo de la evolución de las ontologías. Como hemos comentado previamente, la principal fuente de datos de nuestro proyecto serán las ontologías públicamente accesible en repositorios de ontologías. En este proyecto decidimos centrarnos en repositorios que permitan ser analizados de forma sistemática, por eso dirigimos nuestra visión al OBO Foundry.

La estructura fundamental de nuestros datos se origina en la plataforma del OBO Foundry, de donde se extraen en formato JSON mediante un archivo específico proporcionado en su sitio web. Este archivo consiste en una serie de objetos que siguen una estructura particular, una representación que se puede apreciar en la Figura 5. Este archivo contiene información crucial sobre cada ontología, incluyendo su dominio, su abreviatura y el enlace al repositorio de GitHub donde se almacena. Este archivo JSON es la piedra angular de nuestro proyecto, ya que nos proporciona el punto de partida para identificar y localizar las ontologías relevantes para nuestro análisis.

Antes de proceder al análisis de cada ontología, realizamos una verificación para asegurar que la ontología sigue estando activa. Este proceso se realiza a través del atributo “activity_status” del objeto JSON correspondiente. Solo las ontologías activas son consideradas para el análisis posterior. Iniciamos la verificación del estado de actividad de cada objeto, a través de dicho atributo. Este paso es crucial, ya que nos permite filtrar y concentrarnos en aquellos elementos que están en uso activo, lo que optimiza la eficiencia de nuestro análisis y el valor de los datos obtenidos. Una vez confirmada la actividad de los objetos, procedemos a recopilar información específica de cada uno de ellos. Esta se basa en los siguientes atributos:

- “domain”: representa la clasificación general de la ontología, permitiéndonos entender su aplicación y contexto en un sentido amplio.
- “id”: es la abreviatura o el identificador único de la ontología, esencial para

```
{
  "activity_status": "active",
  "build": {
    "checkout": "git clone https://github.com/EBISPOT/amphx_ontology.git",
    "path": ".",
    "system": "git"
  },
  "contact": {
    "email": "hescriva@obs-banyuls.fr",
    "github": "hescriva",
    "label": "Hector Escriva",
    "orcid": "0000-0001-7577-5028"
  },
  "dependencies": [
    {
      "id": "uberon"
    }
  ],
  "description": "An ontology for the development and anatomy of Amphioxus (Branchiostoma lanceolatum).",
  "domain": "anatomy and development",
  "homepage": "https://github.com/EBISPOT/amphx_ontology",
  "id": "amphx",
  "layout": "ontology_detail",
  "license": {
    "label": "CC BY 3.0",
    "logo": "http://mirrors.creativecommons.org/presskit/buttons/80x15/png/by.png",
    "url": "http://creativecommons.org/licenses/by/3.0/"
  },
  "ontology_purl": "http://purl.obolibrary.org/obo/amphx.owl",
  "preferredPrefix": "AMPHX",
  "products": [
    {
      "id": "amphx.owl",
      "ontology_purl": "http://purl.obolibrary.org/obo/amphx.owl"
    },
    {
      "id": "amphx.obo",
      "ontology_purl": "http://purl.obolibrary.org/obo/amphx.obo"
    }
  ],
  "repository": "https://github.com/EBISPOT/amphx_ontology",
  "title": "The Amphioxus Development and Anatomy Ontology",
  "tracker": "https://github.com/EBISPOT/amphx_ontology/issues"
},
}
```

Figura 5: Ejemplo de Ontologías en archivo JSON

mantener un registro organizado y preciso.

- “repository”: se trata del enlace directo al repositorio de GitHub donde reside la ontología. Esto facilita el acceso directo a los datos y su posterior manipulación.

Con estos datos esenciales en nuestras manos, nos embarcamos en el proceso de extracción de los datos contenidos en estos repositorios. Nuestra primera meta

en este proceso es ubicar los archivos de las ontologías dentro de sus respectivos repositorios de GitHub [13]. Para esto vamos a utilizar las APIs de GitHub destinadas específicamente para la extracción de datos de sus repositorios [12].

Siguiendo con nuestra investigación inicial, observamos que los archivos de las ontologías tienden a seguir ciertos patrones de ubicación al explorar los repositorios. Aprovechando este hallazgo, hemos compilado una lista de posibles enlaces que podrían conducir a los archivos en cuestión. Con esta lista a nuestra disposición, procedemos a inspeccionar cada repositorio. Esta táctica nos permite descubrir con eficacia las ubicaciones de los archivos OWL y OBO. La identificación de estos archivos es un paso crítico ya que son ellos los que contienen los datos semánticos que constituyen las ontologías, permitiéndonos así avanzar en nuestro análisis y en el logro de nuestros objetivos.

Una vez localizados los archivos de ontología, podemos proceder a rastrear su evolución a lo largo del tiempo. Utilizamos la funcionalidad de “commits” proporcionada por GitHub para este propósito. Avanzamos por tanto a la siguiente etapa de nuestro proceso de análisis: la identificación y registro de los cambios que se han efectuado en estos archivos a lo largo del tiempo.

La plataforma de GitHub ofrece una característica muy útil en este aspecto: los “commits”. Los “commits” son, esencialmente, puntos de guardado o instantáneas de los archivos en un momento particular del tiempo, que registran cualquier cambio que se haya realizado. Cada “commit” representa un conjunto de cambios en el repositorio, que pueden incluir la creación, modificación o eliminación de archivos. Además, este registro de cambios es exhaustivo e incluye

detalles como el autor del cambio, una descripción del cambio, y la fecha y hora exactas de cuando se realizó el cambio. En la Figura 6 se puede observar estos detalles.

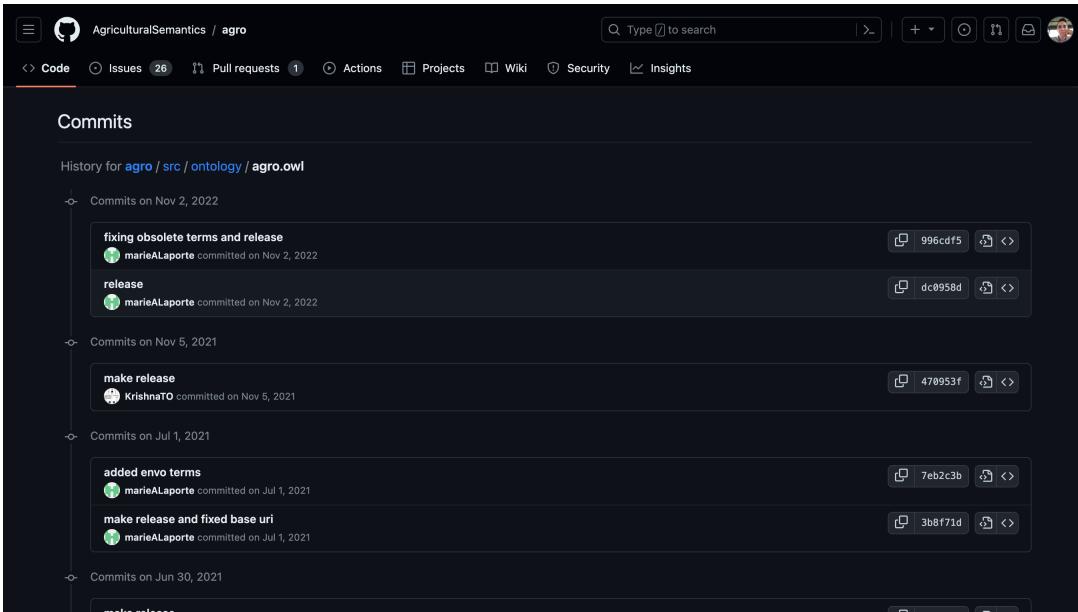


Figura 6: Ilustración de los “commits” en GitHub

En nuestra investigación, aprovechamos esta característica para rastrear la evolución de las ontologías. Al analizar la serie de “commits” asociados con los archivos OWL y OBO de cada ontología, podemos identificar cuándo y cómo se han modificado estas ontologías a lo largo del tiempo. Además, también nos permite detectar patrones o tendencias en la evolución de las ontologías, lo que podría proporcionar una valiosa percepción sobre cómo y por qué estas ontologías cambian. Cada “commit” identificado se almacenará cuidadosamente en nuestro conjunto de datos para cada ontología, construyendo de este modo un registro histórico

detallado de las modificaciones efectuadas. Esta información se mantendrá organizada y accesible para su uso en análisis y visualizaciones posteriores.

Además de analizar los “commits” de los archivos de ontología individuales, se puede aplicar un enfoque similar para rastrear los cambios en los repositorios en general. Al utilizar los enlaces a los repositorios recopilados previamente, se puede acceder y analizar los “commits” de todo el repositorio, proporcionándonos una vista aún más completa de la evolución de los datos. Este proceso integral permitirá no sólo identificar los cambios individuales, sino también comprender el panorama general de cómo estas ontologías, y los datos que contienen, han evolucionado y se han adaptado a lo largo del tiempo. En última instancia, este análisis exhaustivo nos acerca a nuestro objetivo final: la creación de una herramienta efectiva y precisa para el estudio de la evolución de las ontologías a lo largo del tiempo.

2.1.5. Caracterización de una ontología mediante métricas

En el apartado anterior hemos analizado la actividad de una ontología. Sin embargo, dicho análisis no incluye información sobre qué tipo de cambios ha sufrido la ontología entre sus distintas versiones. Los cambios de una ontología se pueden deber a diferentes aspectos como la corrección de su contenido o la generación de nuevo conocimiento. Un ejemplo concreto de estas evoluciones es la Ontología de Enfermedades Humanas (Human Disease Ontology), que ha experimentado numerosas actualizaciones a lo largo de los años para reflejar la aparición de nuevos conocimientos y comprensiones en el campo de la medicina

[49]. En un informe reciente, los desarrolladores de esta ontología detallan cómo han agregado datos específicos sobre COVID-19 para reflejar la emergencia de la pandemia [50]. Esta es solo una muestra del dinamismo y la adaptabilidad de las ontologías ante los cambios en el entorno de la información.

Además de estos cambios específicos relacionados con el contenido de la ontología, existen principios generales de buenas prácticas para su construcción. Como se comentó en la introducción, en esta línea el referente es el OBO Foundry, que nació a principios del nuevo milenio con el objetivo de organizar y fomentar el desarrollo de ontologías basadas en estándares y principios comunes [56]. Se trata de un consorcio de ontologías basadas en principios comunes y mecanismos de gobernanza que todas las ontologías deben cumplir para ser incluidas en el repositorio. En la Figura 7 se muestran el listado de estos principios, que se pueden consultar con más detalle en su página web.

Como se puede en la figura anterior, los principios del OBO abarcan una amplia gama de aspectos, desde la definición de la ontología, hasta el uso de relaciones estándar, y aunque son considerados como recomendaciones o sugerencias, su adhesión es esencial para garantizar la coherencia y la interoperabilidad entre las diferentes ontologías alojadas por el OBO Foundry. La gestión de OBO Foundry es comunitaria, con un equipo de voluntarios dedicados a su mantenimiento y a la toma de decisiones sobre su futuro [21]. Este enfoque colaborativo ha facilitado el intercambio de conocimientos y experiencia entre diferentes dominios, contribuyendo a la evolución y mejora continua de las ontologías. Creemos que la herramienta propuesta en este TFM supondría un elemento de interés al ofrecer

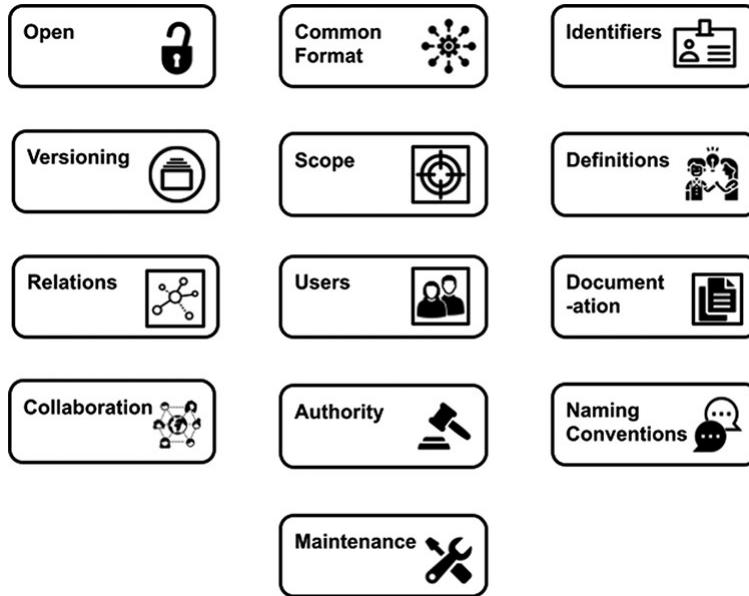


Figura 7: Principios fundamentales de OBO Foundry

datos adicionales sobre estas ontologías. Por ejemplo, uno de los principios del OBO es *Versioning* y que las ontologías mantengan un registro de sus diferentes versiones a lo largo el tiempo. Aquí queremos explotar esto para visualizar los cambios y la actividad de las distintas ontologías del repositorios.

Además de estos principios generales, la comunidad ha trabajado activamente en la necesidad de definir métricas que permitan medir y valorar el seguimiento de ciertos estándares los cuales pueden ayudar a definir la calidad de una ontología como tal. Como ejemplos representativos se podrían mencionar OOPS [46], OQuaRE [10] y OntoQA [61].

La herramienta OOPS [61] ofrece tanto una interfaz Web como un Servicio Web (API) que se basa en revisar una lista de errores típicos o “*pitfalls*” que los desarrolladores de estas ontologías cometan para así evaluar la calidad de la

misma [46]. En la Figura 8 podemos ver todos estos errores típicos clasificados por sus diferentes dimensiones.

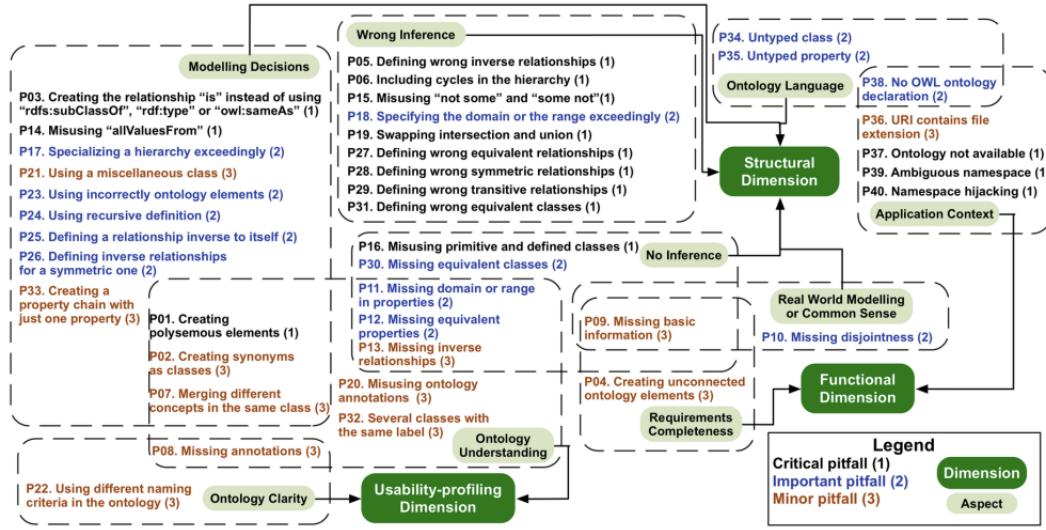


Figura 8: Clasificación de “pitfall” (extraída de [46])

El proyecto de OntoQA [61] es también una Herramienta Web, aunque no está en línea sino que su última versión debe ser instalada y mantenida por el usuario en su propio equipo. A diferencia de OPPS no utiliza una lista de errores comunes, sino que a través de cálculos puede hacer un análisis de la calidad de las ontologías. A continuación se detallan algunas de las métricas que OntoQA es capaz de calcular:

- **Métricas del Esquema:** apuntan al diseño de las ontologías. Es una realidad que el diseño no nos puede dar una garantía del conocimiento de la misma pero si la misma sigue un buen diseño nos podría dar una pista del estado del mismo. Algunas de las métricas definidas son “*Relationship Richness*”

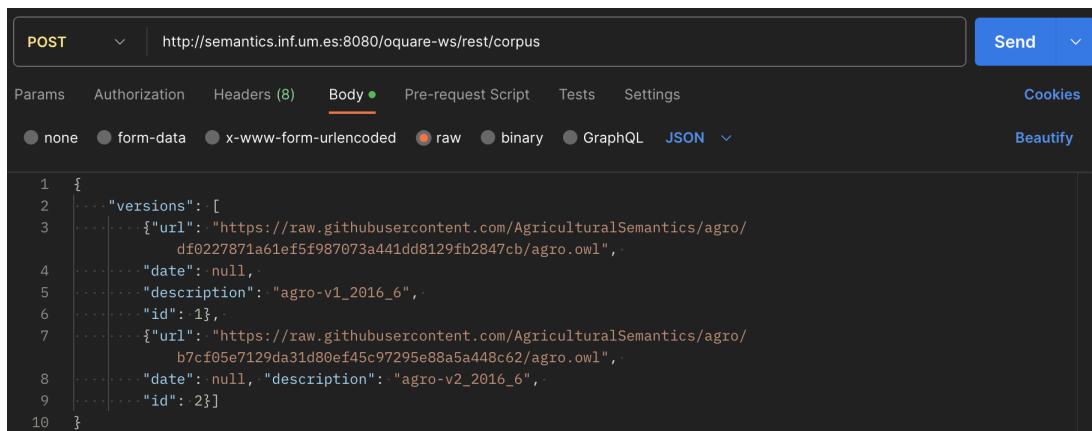
(Riqueza de la Relación), “*Attribute Richness*” (Riqueza de los Atributos) y “*Inheritance Richness*” (Riqueza de la Herencia).

- **Métricas de Instancias:** [46] nos indica que “*la ubicación de los datos de instancia y la distribución de los datos pueden indicar la efectividad del diseño de la ontología y la cantidad de conocimiento representado por la ontología.*” Básicamente estas métricas se dividen en dos que son “*Knowledgebase Metrics*” (Métricas de Base de Conocimiento) y “*Class Metrics*” (Métricas de Clase).

Por último tenemos las métricas de OQuaRE [10], las más relevantes a nuestro trabajo. Estas nacen de la necesidad de encontrar un punto de encuentro entre la Ingeniería de Software y la Ingeniería Ontológica. El objetivo fue tratar de aplicar los mismos estándares de evaluación de la calidad del software del reconocido ISO/IEC 25000:2005, comúnmente conocido como SQuaRE [1], a las ontologías debido a que se quiso considerar una ontología como un artefacto de software [11]. OQuaRE se construye sobre SQuaRE buscando definir todos los elementos requeridos para la evaluación de las ontologías dando “*soporte de evaluación, proceso de evaluación y métricas*”[11]. Algunas de estas métricas definidas e implementadas por OQuaRE son: *Annotation Richness (ANOnto)* (media del número de anotaciones por clase), *Attribute Richness (AROnto)* (media del número de atributos por clase), *Coupling between Objects (CBOOnto)* (número de clases relacionadas), *Class Reachness (CROnto)* (media del número de instancias por clase), *Properties Richness (RROnto)* (número de propiedades definidas

en la ontología dividida por el número de propiedades y relaciones), o *Number of Properties (POnto)* (número de propiedades por clase).

Estas solo son unas cuantas propiedades de las proponen [11] en su documento de investigación, que además se ofrece a la comunidad a través del Servicio Web (gratuito) comentado en las secciones anteriores. El servicio permite obtener distintos distintos tipos de métricas útiles en OQuaRE: las básicas y las calculadas. Para acceder a este servicio solo se necesita hacer una solicitud HTTP de tipo POST utilizando una URL siguiendo los pasos indicados en [2].



```
POST http://semantics.inf.um.es:8080/oquare-ws/rest/corpus
Body (JSON)
1 {
2   "versions": [
3     {
4       "url": "https://raw.githubusercontent.com/AgriculturalSemantics/agro/
5           df0227871a61ef5f987073a441dd8129fb2847cb/agro.owl",
6       "date": null,
7       "description": "agro-v1_2016_6",
8       "id": 1
9     },
10    {
11      "url": "https://raw.githubusercontent.com/AgriculturalSemantics/agro/
12          b7cf05e7129da31d80ef45c97295e88a5a448c62/agro.owl",
13      "date": null,
14      "description": "agro-v2_2016_6",
15      "id": 2
16    }
17  ]
18 }
```

Figura 9: Solicitud HTTP POST desde Postman

En la Figura 9 podemos ver un ejemplo de como se podría hacer esta solicitud utilizando una aplicación dedicada para hacer solicitudes web llamada Postman. La solicitud web puede devolver dos tipos de respuesta:

- **Un correo electrónico con todos los datos:** esta opción es la más sencilla de implementar, pero puede ser ineficiente para grandes conjuntos de datos.
- **Un archivo comprimido ZIP con un conjunto de archivos CSV y XML:**

esta opción es más compleja de implementar, pero ofrece una serie de ventajas sobre la primera.

En este trabajo utilizaremos la segunda opción, ya que es la mejor manera de procesar los datos. El archivo comprimido ZIP nos permite obtener todos los datos de una vez, lo que facilita el procesamiento. Además, los archivos CSV y XML son formatos estándar, por lo que son fáciles de leer y analizar. Además, la elección de esta opción nos permitirá realizar las operaciones de lugar, lo que es esencial para el éxito del proyecto. Estas se realizan en el mismo lugar donde se encuentran los datos, lo que reduce el tiempo de procesamiento y la necesidad de transferir datos a otro servidor. En la Figura 10 podemos ver un ejemplo de los archivos que retorna la solicitud una vez descomprimido el archivo.

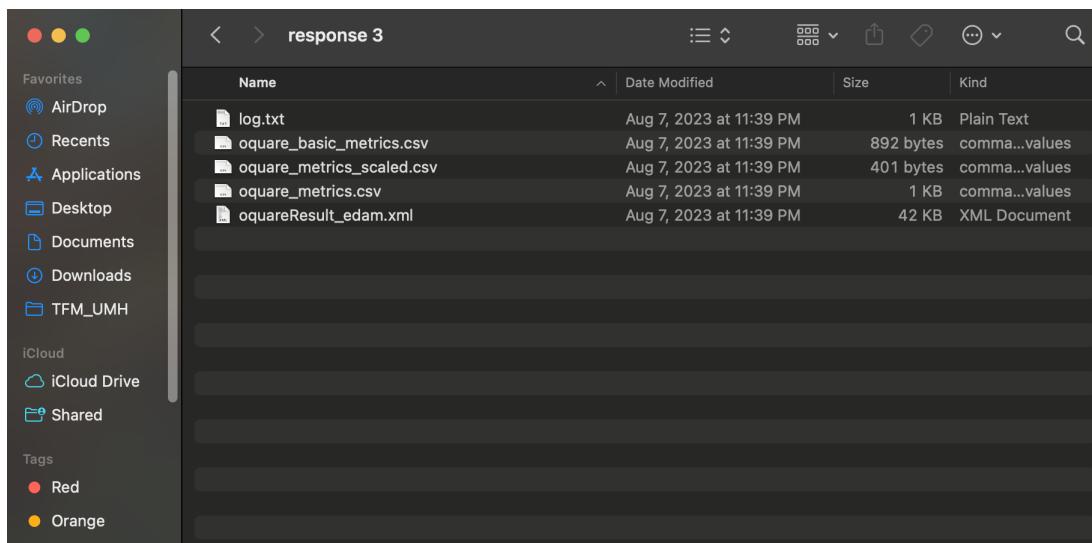


Figura 10: Archivos recibidos por la solicitud

2.1.6. Herramientas para la visualización de datos

Aunque no hemos identificado herramientas específicas que se centren en la visualización de datos para ontologías, durante nuestra revisión bibliográfica nos encontramos con trabajos que hacen uso de *Dashboards* para representar información. Este tipo de soluciones creemos que son apropiadas para mejorar la visualización de las diferentes métricas ofrecidas por las soluciones mencionadas en el apartado anterior. En este contexto, nos gustaría destacar tres herramientas que llamaron nuestra atención por su interesante forma de representar los datos y, en alguno de los casos, su forma de filtrar.

En primer lugar tenemos un *dashboard* dedicado a comparar la tasa de mortalidad estandarizada por riesgo (RSMR) de pacientes con sepsis [26]. El equipo de investigadores ha desarrollado una Herramienta Interactiva Web que permite comparar las tasas de mortalidad estandarizadas por riesgo (RSMR) de pacientes con sepsis en distintos estados de EEUU, facilitando así el análisis profundo y la comprensión de las estadísticas relacionadas con la sepsis. Para su desarrollo se ha utilizando el *framework* Shiny [54] de R que permite integrar cálculos estadísticos basados en R con la generación de gráficos. Este *dashboard* procesa datos del “*National Inpatient Sample Database*” analizando cerca de 35 millones de hospitalizaciones por año. La herramienta, que resume ocho años de datos, está disponible públicamente y se puede acceder desde cualquier plataforma a través del sitio web [25].

Esta herramienta visual ofrece una serie de funciones interactivas, incluyen-

do un mapa de calor y gráficos de series temporales, que permiten a los usuarios explorar las variaciones sustanciales en las RSMR en toda la EEUU. Con su enfoque en la facilidad de uso y accesibilidad, la herramienta se erige como un recurso valioso para individuos y entidades interesadas en las desigualdades de salud, facilitando la toma de decisiones informadas en el sector salud y promoviendo la formulación de políticas públicas basadas en datos concretos. En las Figuras 11 y 12 podemos ver algunas visualizaciones de este *dashboard*.

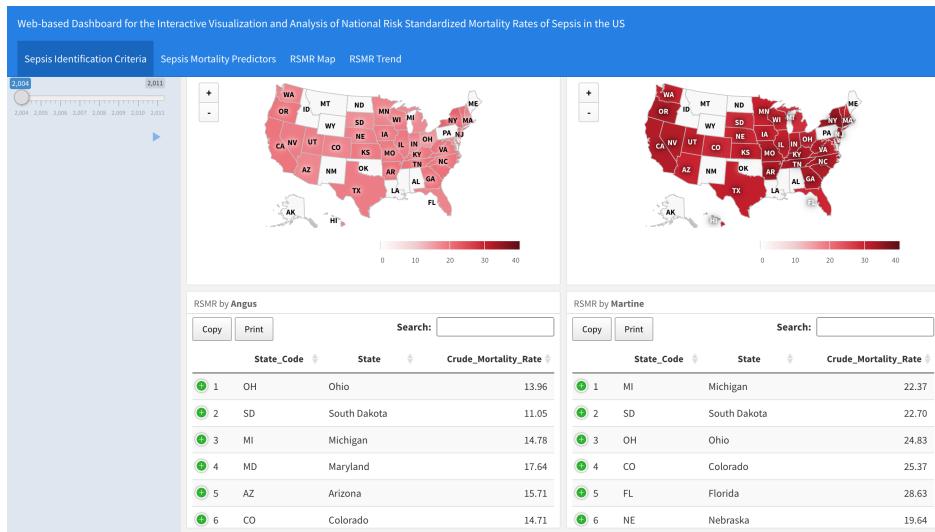


Figura 11: Dashboard sobre la tasa RSMR de pacientes con sepsis (I)

Por otro lado tenemos a VISUEL [3], la cual surge como una herramienta clave para el análisis y visualización de datos, facilitando la exploración rápida e intuitiva de grandes volúmenes de información mediante representaciones visuales interactivas. A través de diversas técnicas, como diagramas de dispersión y coordenadas paralelas, ofrece una perspectiva detallada y personalizable de los datos disponibles.

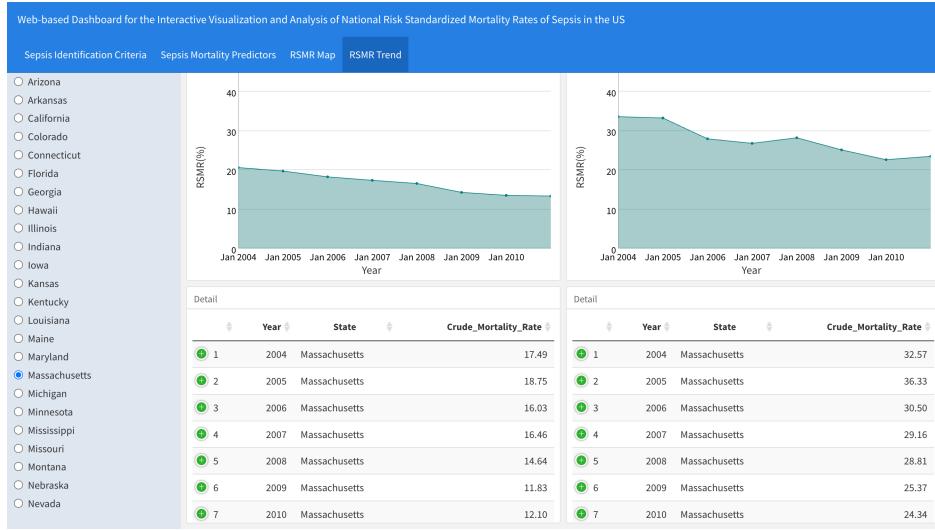


Figura 12: Dashboard sobre la tasa RSMR de pacientes con sepsis (II)

La plataforma VISUEL se ha revelado especialmente útil en la industria vinícola argentina [3], donde ha permitido descubrir aspectos significativos y relevantes del sector. Mediante la creación y análisis de múltiples vistas, VISUEL facilita una comprensión profunda de los datos, promoviendo un análisis más informado y una toma de decisiones basada en información detallada y veraz. Aunque este *dashboard* no está accesible, en la Figura 13 adjuntamos una captura extraída de su publicación.

Por último está GNPS Dashboard [44], el cual facilita la exploración y análisis detallado de los datos obtenidos a través de espectrometría de masas (MS), proporcionando visualizaciones de alta calidad para publicaciones. Es una herramienta esencial para revisar y validar datos de MS publicados. Una función destacada es su capacidad para compartir visualizaciones a través de URLs o códigos QR, promoviendo una colaboración efectiva y transparente entre los usuarios. Esto no



Figura 13: Dashboard de VISUEL (extraída de [3])

solo elimina malentendidos durante las reuniones remotas, sino que fomenta la transparencia y la reproducibilidad en el análisis de datos de MS, facilitando la entrada a este tipo de análisis tanto en entornos de investigación como educativos. Podemos acceder a este tablero a través de su página web [43]. En la Figura 14 podemos apreciar como se ve este dashboard.

2.2. Selección de lenguajes de programación para O-Metrics

Para la implementación de la herramienta que estamos diseñando, la elección recayó sobre el lenguaje de programación Python. Este lenguaje destacó como nuestra opción preferida por varias razones fundamentales. En primer lugar, la simplicidad y legibilidad de Python son bien conocidas, lo que se traduce en un

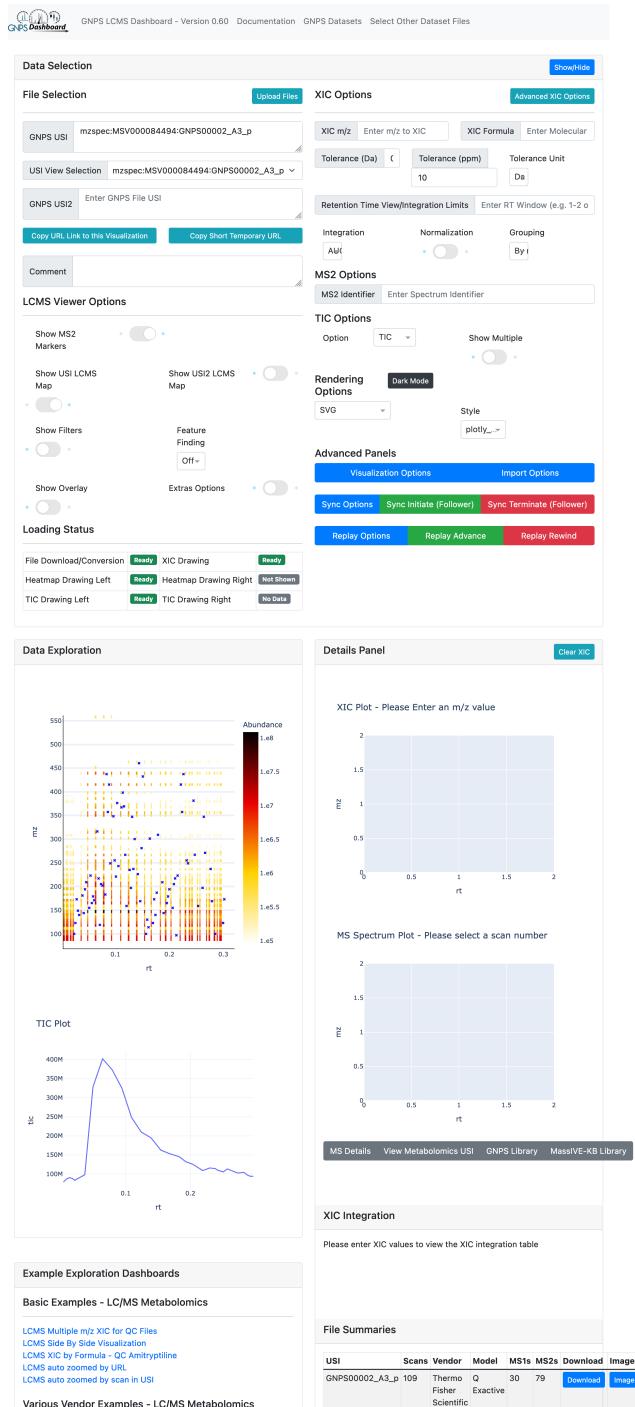


Figura 14: Dashboard GNPS (extraída de [44])

código de fácil comprensión y modificación, incluso para aquellos con habilidades de programación básicas. En segundo lugar, la popularidad de Python es innegable, lo que se traduce en una extensa y activa comunidad de usuarios y una gran diversidad de recursos disponibles para solucionar problemas o potenciar las funcionalidades. Además, dispone de una amplia gama de bibliotecas y capacidad para manejar datos complejos.

Python, un lenguaje de programación concebido por Guido van Rossum en los últimos años de la década de 1980 y lanzado oficialmente en 1989 [62]; desde entonces ha demostrado su valor y resiliencia a lo largo del tiempo. Mientras sus aspectos funcionales y técnicos han experimentado transformaciones, su filosofía central, que enfatiza la legibilidad del código y la expresión concisa de ideas, ha permanecido intacta. La popularidad de Python ha experimentado un auge considerable en los últimos años, en parte gracias a su amplia aplicación en campos como la Ciencia de Datos y la Inteligencia Artificial, lo que ha llevado a algunos a denominarlo un “*lenguaje de programación de crecimiento rápido*” [57].

Las características fundamentales de Python que han cimentado su estima entre los desarrolladores son diversas [53]:

- **Simplicidad:** es un lenguaje de alto nivel, se distingue por su sintaxis limpia y sencilla en comparación con otros lenguajes como C#, Java o C++. Esto se traduce en una mayor concentración en la solución de problemas y menos en la gestión de la complejidad del lenguaje, convirtiendo a Python en una excelente opción para los recién iniciados en la programación.

- **Portabilidad:** es un lenguaje multiplataforma, capaz de operar en diversos sistemas operativos sin necesidad de modificaciones en el código.
- **Código abierto:** su naturaleza de código abierto facilita su distribución gratuita y su desarrollo comunitario. Actualmente, una comunidad sólida y comprometida de desarrolladores voluntarios respalda su mantenimiento y mejora continua.
- **Extensa biblioteca estándar y comunitaria:** ofrece una amplia biblioteca estándar, que ahorra tiempo en la codificación y acelera el desarrollo. Además, las bibliotecas desarrolladas por la comunidad y las corporaciones, como Google y Facebook, aportan una variedad de funciones útiles para diversas aplicaciones.

2.2.1. Librerías para el procesamiento y visualización de datos

Una de las características más distintivas de Python, en particular en el contexto de la Ciencia de Datos, es su rica colección de librerías dedicadas al procesamiento y análisis de datos. Librerías como Pandas [42], NumPy [39] y SciPy [51], entre otras, proporcionan un conjunto de herramientas poderoso y diverso para la manipulación, análisis y visualización de datos [57].

En cuanto a la visualización, en el ecosistema de Python, existen numerosas librerías de visualización de alta calidad, entre las cuales Matplotlib [30], Seaborn [52], Plotly [45] y Bokeh [7] son particularmente reconocidas por su utilidad y popularidad en la comunidad de análisis de datos y ciencias [58].

En nuestro proyecto, para construir nuestro tablero interactivo, o *dashboard* utilizaremos un stack de tecnologías web muy robusto y en particular las siguientes herramientas:

- **Flask [47]:** es una librería de Python que se encarga de crear aplicaciones web de una manera sencilla intentando usar el menor número de líneas de código posibles.
- **HTML/CSS y Bootstrap [33, 32, 8]:** nos permitirá crear todas nuestras visualizaciones a partir de librerías web con una mayor integración y atractivo para el usuario.

Dado que hemos decidido tomar una implementación web, la elección de librerías de visualización se restringe a aquellas que sean compatibles con estas tecnologías. La elección de la librería óptima puede ser un desafío dada la gran cantidad de opciones con características similares. Algunas librerías que podemos seleccionar son D3.js [31], Highcharts [23], Chart.js [9] y la implementación web de Plotly [24]. A continuación realizamos un pequeño análisis de cada una de ellas:

- **D3.js (Data-Driven Documents) [31]:** es una biblioteca de JavaScript muy potente y flexible que permite crear visualizaciones de datos interactivas directamente en el navegador web. Su capacidad para manipular documentos basados en datos la convierte en una de las herramientas más versátiles. No obstante, esta flexibilidad viene con una curva de aprendizaje más empinada, que podría ralentizar el desarrollo.

- **Highcharts [23]:** es otra biblioteca de JavaScript que facilita la creación de gráficos interactivos para sitios web. Es muy configurable y fácil de integrar en proyectos web, pero su modelo de licencia comercial puede ser un inconveniente para proyectos con restricciones presupuestarias.
- **Chart.js [9]:** ofrece simplicidad y ligereza en una biblioteca para visualización de datos. Proporciona una manera fácil de crear gráficos en HTML5 con la ayuda de lienzo (canvas) de HTML. Su enfoque en gráficos más simples y convencionales facilita la implementación rápida, pero podría ser limitante para visualizaciones más complejas .
- **Plotly [24]:** es una biblioteca que no solo ofrece una amplia variedad de tipos de gráficos sino que también ofrece potentes funcionalidades interactivas. A diferencia de D3.js, tiene una curva de aprendizaje más amigable, y es altamente customizable.

Después de considerar cuidadosamente estas alternativas, hemos seleccionado Plotly como nuestra biblioteca de visualización principal para el proyecto. Esta decisión está fundamentada en la capacidad de Plotly para integrarse de manera fluida en nuestros flujos de trabajo y en la facilidad con la que permite construir *dashboards* interactivos. Además, esta naturaleza interactiva permite una exploración de datos más dinámica e intuitiva. Otro aspecto destacable es que se integra bien con bibliotecas de análisis de datos como Pandas, lo cual es una ventaja en proyectos de Ciencia de Datos.

En el contexto de nuestro herramienta, y de forma resumida, los datos recogi-

dos y analizados serán transformados en visualizaciones interactivas con la ayuda de la librería Plotly, que se integrarán en una herramienta web con la ayuda de Flask. Estas visualizaciones facilitarán la interpretación de los resultados y proporcionarán una representación gráfica de la evolución de las ontologías. Además, gracias a su carácter interactivo, las visualizaciones permitirán una exploración más dinámica y atractiva de los datos.

2.2.2. Procesamiento de las ontologías y extracción de métricas

Merece la pena destacar que inicialmente exploramos el uso de una librería especializada para el manejo de datos de ontologías llamada Owlready2 [22]. Con la ayuda de Owlready2 se puede extraer información valiosa sobre la evolución de las ontologías y sus diferentes versiones. Sin embargo, esta solución fue descargada al considerar el tiempo de procesamiento y la carga en el servidor que alojaría nuestra aplicación. En su lugar, decidimos utilizar el servicios web externos y basado en OQuaRE mencionado anteriormente.

El uso de este servicio de terceros ofrecen una serie de ventajas sobre el uso de una biblioteca local. En primer lugar, son más escalables, lo que significa que pueden manejar más tráfico sin afectar al rendimiento de la aplicación. En segundo lugar, son más fáciles de mantener, ya que esta tarea queda delegada en un tercero que se encarga de actualizar y parchear los servicios. En tercer lugar, son más seguros, ya que los administradores del sistema donde están alojados utilizan las últimas medidas de seguridad para proteger los datos.

2.3. Arquitectura de la solución a desarrollar

En la Figura 15 se ilustra el esquema arquitectónico integral de la solución que proponemos. Este esquema está compuesto por tres componentes críticos: el “*Data Mining Engine*”, la “*O-Metrics API*”; y el “*OQuaRE Metrics Web Service*”. La “*O-Metrics API*” actúa como el núcleo operativo que interconecta todas las funcionalidades y procesos.

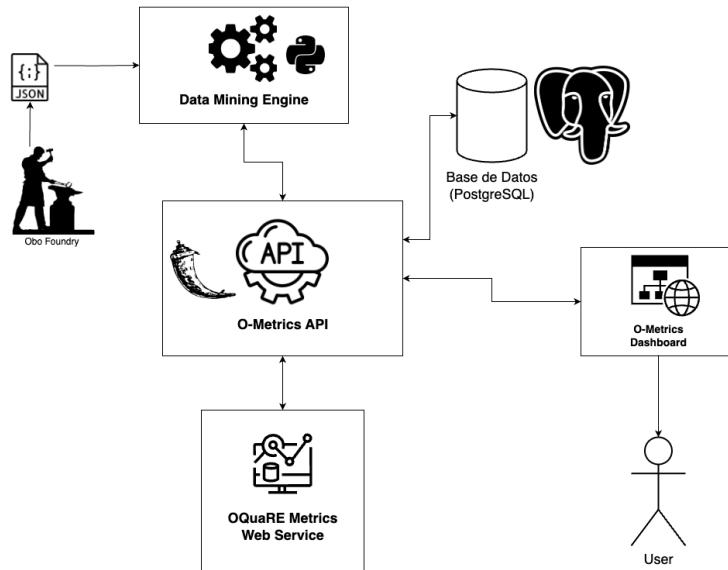


Figura 15: Arquitectura de la Solución Propuesta

De forma resumida, el “*Data Mining Engine*” tiene la responsabilidad de extraer y procesar información de los repositorios de GitHub donde se almacenan las ontologías. Este motor obtiene la ubicación de estos repositorios de un archivo JSON que se genera a partir de datos disponibles en el sitio web del OBO Foundry, como se indica en la figura asociada. Adicionalmente, se hace uso del “*OQuaRE*

Metrics Web Service” para adquirir métricas relevantes relacionadas con las ontologías en estudio. Por último, es vital señalar que todos los datos procesados son almacenados en una base de datos centralizada desarrollada en PostgreSQL. Este repositorio de datos está diseñado para ser accesible exclusivamente a través de la “*O-Metrics API*”, garantizando así la seguridad e integridad de la información almacenada.

Además, en el diseño de nuestra solución una de las consideraciones más importantes ha sido la accesibilidad para el usuario final. Por ello, hemos desarrollando una interfaz gráfica de usuario (GUI) a través de la cual los usuarios pueden interactuar fácilmente con las diversas funcionalidades, que se describirán con más detalle en el siguiente capítulo.

3. Resultados

En este capítulo, se van a tratar aspectos relacionados con la implementación de la herramienta y cómo se ha aplicado a un repositorio de ontología relevante para la comunidad.

3.1. Planificación del proyecto

Para el desarrollo de este proyecto decidimos utilizar la metodología ágil Scrum [19]. Por experiencias anteriores puedo decir que esta metodología es excelente cuando se trata de proyectos web debido a su manera de adaptarse a la volatilidad de los cambios que se producen en estos tipos de proyectos. El énfasis de esta metodología en la comunicación entre los diferentes miembros del equipo garantiza al final un mejor producto.

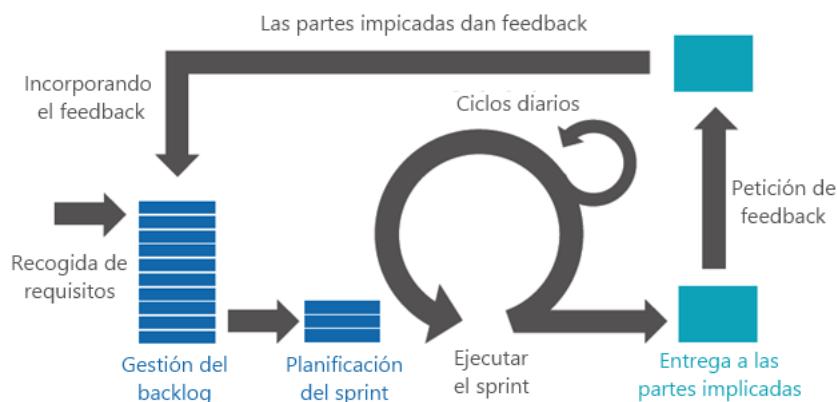


Figura 16: Diagrama Metodología Scrum

En la Figura 16 podemos ver representado como funciona esta metodología a

grande rasgos. Scrum utiliza unos elementos llamados *sprints* o iteraciones que no deben ser mayores a cuatro semanas [19]. Un *sprint* equivale a un conjunto de tareas asignadas a un equipo. Pasemos a dar un poco más de detalle de los elementos que podemos ver en la figura anterior:

1. **Recogida de requisitos:** se realiza una reunión para recopilar las características requeridas para la aplicación en cuestión.
2. **Planificación del sprint:** una vez bien definidos los requisitos, se planifica el *sprint* de modo que se pueda cumplir de la manera más eficiente posible.
3. **Ciclos diarios:** esta parte trata de reuniones frecuentes que tienen los integrantes del equipo para informar en qué van a estar trabajando y comunicar dudas o dificultades que están afrontando en el momento.
4. **Ejecución del sprint:** ya para este paso se le da inicio a la iteración y se asegura su propia elaboración a través de los ciclos diarios.
5. **Entrega de las partes implicadas:** una vez finalizado el *sprint* se realiza una reunión para hacer entrega de los avances.
6. **Retroalimentación:** se recibe una retroalimentación de la entrega y se procede a repetir todo el proceso.

En este proyecto no se llevó esto de una manera estricta ya que por ejemplo, no se mantuvo un ciclo diario a base de reuniones. Lo que si procuramos fue mantener una comunicación frecuente a través de correo electrónico o aplicaciones de

mensajería entre estudiante y tutor de modo que ambos estuviéramos al tanto de que estaba sucediendo en cada momento.

En la Figura 17 podemos ver un cronograma de las tareas realizadas en cada *sprint* durante el desarrollo de O-Metrics, donde cada columna representa el periodo de una semana. A continuación, se describen brevemente los diferentes bloques y tareas incluidas:

■ **Bloque 1:** Planificación

- 1.1. Búsqueda de Fuente de Datos.
- 1.2. Búsqueda de APIs o Servicios Web disponibles.
- 1.3. Elección de librerías y stack de tecnologías a utilizar.

■ **Bloque 2:** Análisis

- 2.1. Análisis del OBO Foundry como fuente de datos.
- 2.2. Análisis de la API de GitHub y los Servicios Web de OQuaRE.
- 2.3. Análisis de plantillas web para la interfaz gráfica.

■ **Bloque 3:** Diseño de la herramienta

- 3.1. Diseño de la Base de Datos.
- 3.2. Esquematización de la interfaz gráfica.
- 3.3. Diseño de los componentes del software.

■ **Bloque 4:** Desarrollo de la herramienta

- 4.1. Implementación de los Modelos.
- 4.2. Realización de algoritmos para la extracción de datos.
- 4.3. Implementación de los Controladores.
- 4.4. Realización de la conexión del *backend* y *frontend*.

Tareas	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20	S21	S22
Bloque 1																						
1.1.		■																				
1.2.			■																			
1.3.				■																		
Bloque 2																						
2.1.	■				■																	
2.2.		■				■																
2.3.						■																
Bloque 3									■	■												
3.1.									■	■												
3.2.							■	■														
3.3.									■	■												
Bloque 4																						
4.1.											■	■	■									
4.2.											■	■	■									
4.3.												■	■	■	■	■	■	■	■	■	■	■
4.4.												■										

Figura 17: Cronograma de las distintas fases

3.2. Diseño del software e implementación de O-Metrics

3.2.1. Requisitos y casos de uso

Para ilustrar la utilidad práctica y la eficacia analítica de O-Metrics, presentamos varios escenarios de uso que destacan cómo la herramienta puede ser empleada para abordar cuestiones críticas en la gestión y evaluación de ontologías.

- **Evolución temporal de repositorios:** en el primer escenario, un investigador interesado en el desarrollo de ontologías específicas podría utilizar el

panel de visualización para estudiar la evolución temporal de los repositorios. Utilizando los filtros disponibles, el usuario puede enfocarse en métricas particulares como la frecuencia de los cambios, la adición de nuevos conceptos, o la modificación de relaciones existentes. Estos datos, visualizados a lo largo del tiempo, ofrecen *insights* significativos sobre la actividad de desarrollo y pueden ser indicativos de la madurez o estabilidad del repositorio.

- **Comparativa de métricas ontológicas:** en el segundo escenario, un equipo de ontólogos podría emplear O-Metrics para comparar múltiples ontologías basándose en diferentes métricas ofrecidas a través del servicio web de OQuaRE. Esta funcionalidad permitiría, por ejemplo, evaluar la coherencia, la granularidad y la complejidad de diversas ontologías para identificar las más aptas para un proyecto específico.
- **Auditoría de consistencia y calidad:** en un tercer caso, los administradores de un proyecto podrían usar la herramienta para realizar auditorías regulares de la consistencia y calidad de las ontologías involucradas. Al acceder a una vista consolidada que presenta indicadores clave, los administradores pueden tomar decisiones informadas sobre áreas de mejora y posibles refactores.
- **Alertas y notificaciones:** un valor añadido de O-Metrics es la capacidad de configurar alertas y notificaciones. Por ejemplo, un usuario podría establecer una alerta para ser notificado cuando una ontología en particular se

actualiza con nuevos conceptos, lo que podría tener un impacto directo en el rendimiento de un sistema que la utilice.

Estos ejemplos ilustran la versatilidad y profundidad analítica de O-Metrics, lo que lo convierte en una herramienta útil para múltiples actores involucrados en el desarrollo y gestión de ontologías.

3.2.2. Estructura de la Base de Datos

La estructura de la base de datos juega un papel fundamental en la plataforma O-Metrics, sirviendo como el eje central donde se consolidan y organizan todos los datos que alimentarán las diversas funcionalidades de la herramienta. Esta base de datos está segmentada en tres esquemas diferenciados: “*General*”, “*Ontology*”, y “*Repository*”. En la Figura 18 podemos ver un diagrama de nuestra base de datos y sus tablas.

El esquema “*General*” opera como el epicentro para la administración de usuarios en la plataforma, albergando las tablas “*Persona*” y “*Tipo Persona*”. La tabla “*Persona*” está diseñada para contener detalles esenciales de los usuarios, tales como el nombre, la dirección de correo electrónico y cualquier otra información de contacto relevante. Por otro lado, la tabla “*Tipo Persona*” juega un papel crucial en la gestión de los permisos de los usuarios dentro de la plataforma, definiendo los niveles de acceso y las operaciones permitidas para cada usuario, garantizando así una segmentación clara y una estructura de permisos robusta.

Trasladándonos al esquema “*Repository*”, encontramos un diseño meticuloso que facilita el almacenamiento de datos diversificados relativos a los repositorios.

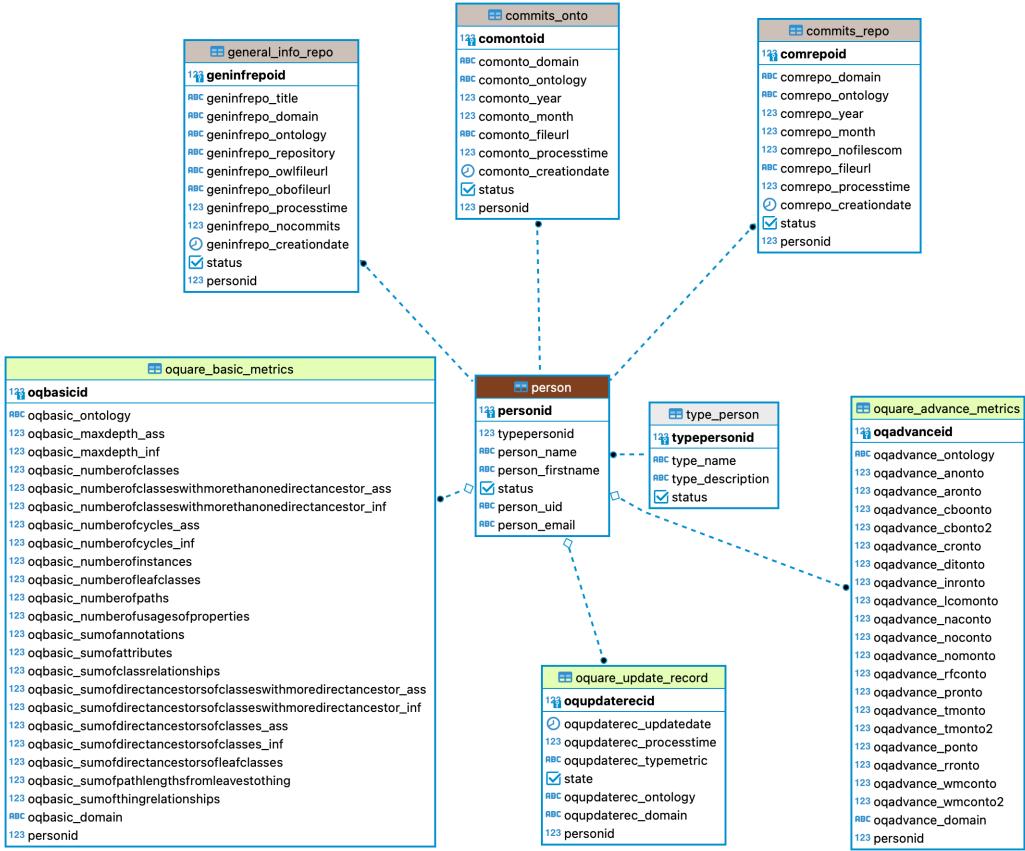


Figura 18: Estructura de Base de Datos

La primera tabla de este esquema se destina a acumular información general sobre los repositorios, incluyendo detalles como el nombre del repositorio, su descripción y metadatos asociados. Las otras dos tablas presentes en este esquema están dedicadas a retener datos más específicos, potencialmente hospedando detalles más granulares sobre las actividades, las interacciones y las métricas clave de los repositorios en cuestión.

Finalmente, nos encontramos con el esquema “*Ontology*”, que se constituye

ye como la base para la acumulación de toda la información relacionada con las métricas OQuaRE. Este esquema no solo encapsula datos directamente derivados de las evaluaciones de OQuaRE, sino que también puede albergar detalles adicionales que ayuden a enriquecer el análisis y la interpretación de estas métricas, como por ejemplo, las anotaciones y las referencias relacionadas.

La estructura de la base de datos está diseñada no solo para facilitar una organización lógica y sistemática de los datos, sino también para permitir una recuperación eficiente de la información, facilitando así operaciones de análisis complejas y garantizando una experiencia de usuario fluida y productiva. Es imperativo que esta estructura se mantenga flexible para acomodar posibles ajustes y expansiones futuras, asegurando que O-Metrics pueda evolucionar y adaptarse a las necesidades cambiantes del paisaje de desarrollo de ontologías.

3.2.3. Integración y Extensibilidad del Sistema

Uno de los aspectos cruciales del diseño de O-Metrics es la alta cohesión e integración entre sus componentes principales: la “Gestión de la Base de Datos” y los “Paneles de Visualización”. Este diseño modular permite que cada componente pueda ser actualizado o extendido de forma independiente sin afectar la funcionalidad del otro.

En el ámbito de la seguridad, se ha tomado en cuenta la integridad de los datos y la confidencialidad de la información almacenada. La base de datos en PostgreSQL, por ejemplo, está configurada para ser accesible exclusivamente a través de la API de O-Metrics, proporcionando un nivel adicional de seguridad.

Todas las transmisiones de datos entre los componentes del sistema se realizan utilizando protocolos seguros.

En cuanto a la escalabilidad, O-Metrics está diseñado para adaptarse al crecimiento tanto en el número de ontologías analizadas como en la cantidad de usuarios. La arquitectura del sistema se ha optimizado para permitir la fácil adición de nuevos módulos o características, asegurando que la plataforma pueda adaptarse a futuras necesidades y tecnologías emergentes.

Este enfoque integrado y escalable no sólo hace que O-Metrics sea una herramienta robusta para el análisis y monitoreo de ontologías sino también una plataforma flexible y segura, capaz de evolucionar junto con las necesidades cambiantes en el ámbito de la gestión de ontologías.

3.3. Descripción del repositorio analizado: OBOFoundry

Se ha aplicado la metodología usando la herramienta sobre el repositorio OBO Foundry ya que sus ontologías se encuentran públicamente accesibles y promueven el uso de un sistema de versionado que permite trazarlas en el tiempo. En septiembre de 2023, el repositorio del OBO Foundry ofrecía a disposición de la comunidad un total de 257 ontologías, según verificamos a través del análisis del archivo JSON que usamos para extraer datos cruciales, tal como ilustramos en la Figura 15. Este proceso de selección y filtrado riguroso garantiza que nuestro estudio esté basado en datos sólidos y confiables. Es esencial destacar que no todas estas ontologías se encuentran en estado activo o disponen de repositorios en GitHub. Por esta razón, hemos decidido implementar un filtro inicial que nos

permitiera enfocarnos solo en aquellas que cumplen con estas condiciones, reduciendo así la cifra a 181. La selección no termina aquí. Anteriormente, señalamos que nuestra metodología incluye el uso de una lista de enlaces para acceder a los archivos de cada ontología. Al aplicar este criterio, descubrimos que solo 170 de ellas cuentan con los archivos OWL, que son vitales para llevar a cabo nuestros análisis detallados. El restante grupo de ontologías, que forma parte de las 181 seleccionadas en el paso anterior, podrán ser analizadas desde el punto de vista de su actividad en GitHub, pero no a través de métricas de OQuaRE que describan su contenido. El listado completo de las ontologías seleccionadas se puede consultar desplegando la herramienta, cuyo código está disponible (ver sección 3.5).

Además, dentro de nuestro procedimiento de recolección y análisis de datos, se categorizarán las ontologías según los diferentes dominios a los que pertenecen. Este enfoque nos permite tener una visión más amplia y detallada del entorno de las ontologías, ofreciendo una perspectiva que va más allá de la simple acumulación de datos, y facilitando la comprensión de las particularidades y aplicaciones específicas de cada una. Hemos identificado un total de quince dominios distintos. Cada uno de estos dominios representa un área específica de conocimiento.

La Tabla 1 muestra los diferentes dominios identificados y el número de ontologías que se han agrupado bajo cada uno de ellos, proporcionando así una visión cuantitativa y cualitativa del universo de ontologías que estamos manejando. En ella se puede apreciar que algunos dominios agrupan a una mayor cantidad de ontologías que otros, lo que señala la diversidad y riqueza del campo de estudio de las ontologías. Con una visión clara de los dominios existentes y el número de on-

tologías que los conforman, los investigadores y profesionales pueden enfocarse en áreas específicas con mayor precisión.

Dominio	Número de Ontologías
<i>health</i>	37
<i>anatomy and development</i>	36
<i>phenotype</i>	21
<i>chemistry and biochemistry</i>	16
<i>organisms</i>	15
<i>investigations</i>	15
<i>information</i>	11
<i>environment</i>	7
<i>biological systems</i>	6
<i>diet, metabolomics, and nutrition</i>	5
<i>information technology</i>	4
<i>upper</i>	3
<i>agriculture</i>	2
<i>microbiology</i>	2
<i>simulation</i>	1

Tabla 1: Listado de Dominios

En cuanto a la extracción y análisis de datos de la actividad de las ontologías, se han recolectado miles de *commits* que serán visualizados para mostrar la actividad de las ontologías a lo largo del tiempo. Aunque lo ideal sería procesar cada uno de esos *commits* con OQuaRe para obtener sus métricas y visualizarlos, hemos decidido restringir el número de versiones a procesar en esta segunda fase.

3.4. Ejemplo de uso de O-Metrics

O-Metrics se configura como una plataforma en línea especializada en el monitoreo y análisis de la evolución temporal y métricas pertinentes de ontologías alojadas en el OBO Foundry. La arquitectura de la solución se bifurca en dos componentes principales:

- **Recolección de Datos y gestión de la BBDD:** este componente se encarga de la extracción de datos desde los repositorios de GitHub y el servicio web de OQuaRE, para posteriormente archivarlos en la base de datos del sistema.
- **Paneles de Visualización (Dashboards):** este segmento se dedica a la representación gráfica de los datos extraídos, proporcionando múltiples tipos de visualizaciones analíticas.

3.4.1. Inferfaz para la recolección de Datos

La recolección de datos se ha automatizado de manera que el usuario pueda realizarla o actualizarla con nuevos datos en cualquier momento. En la Figura 19,

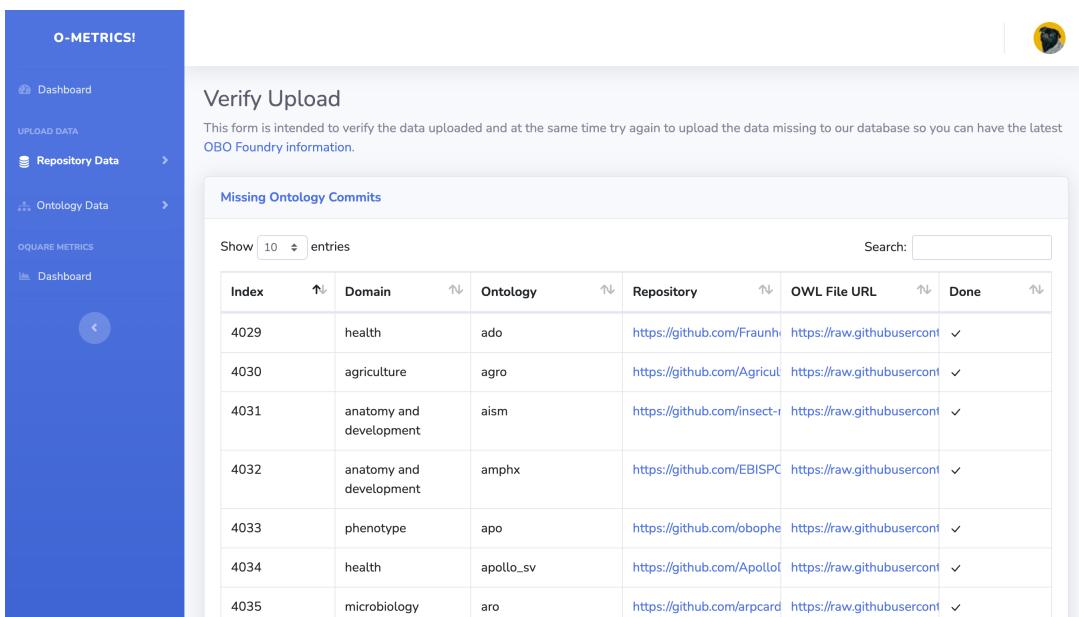
se muestra la interfaz de usuario para esta funcionalidad, en la que se destacan botones específicos para la actualización y carga de datos desde el archivo JSON proveniente del OBO Foundry.

Index	Title	Domain	Ontology	Repository	Ontology URL
0	Alzheimer's Disease Ontology	health	ado	https://github.com/Fraunhofer/AlzDO	http://purl.obolibrary.org/obo/alz.owl
1	Agronomy Ontology	agriculture	agro	https://github.com/Agricultral/ontology	http://purl.obolibrary.org/obo/agro.owl
2	Ontology for the Anatomy of the Insect SkeletoMuscular system (AISM)	anatomy and development	aism	https://github.com/insect-ontology/aism	http://purl.obolibrary.org/obo/aism.owl
3	The Amphiopus Development	anatomy and development	amphx	https://github.com/EBISPC/AmphiopusDevelopment	http://purl.obolibrary.org/obo/amphx.owl

Figura 19: Interfaz de Usuario para la Gestión de Base de Datos de Repositorios

Una vez realizada la ingestión inicial de datos, la plataforma entra en una fase de verificación en la que la herramienta trabajaría en segundo plano para evitar largos tiempos de espera por parte del usuario. En la Figura 20 se presenta un ejemplo de la interfaz de verificación, desde la que los usuarios pueden inspeccionar las tablas que muestran el estado de procesamiento de cada ontología. Una vez terminado el análisis y recolección de los *commits* la herramienta podrá continuar con el cálculo de métricas de OQuaRE a través del “*OQuaRE Metrics Web Service*”. Este proceso se realiza de forma individual para cada ontología, dado que el

servicio web está diseñado para tratar cada entidad de manera singular. Un ejemplo de esta interfaz se presenta en la Figura 21. Al igual que en el caso anterior, procesar cada versión en segundo plano y de forma independiente permite a nuestro sistema ser más robusto y poder recuperarse de posibles errores producidos durante el proceso.



Index	Domain	Ontology	Repository	OWL File URL	Done
4029	health	ado	https://github.com/FraunhoferIIS/ontology	https://raw.githubusercontent.com/FraunhoferIIS/ontology/main.owl	✓
4030	agriculture	agro	https://github.com/AgriculturalGenomics/ontology	https://raw.githubusercontent.com/AgriculturalGenomics/ontology/main.owl	✓
4031	anatomy and development	aism	https://github.com/insect-iis/ontology	https://raw.githubusercontent.com/insect-iis/ontology/main.owl	✓
4032	anatomy and development	amphx	https://github.com/EBISPC/ontology	https://raw.githubusercontent.com/EBISPC/ontology/main.owl	✓
4033	phenotype	apo	https://github.com/obophe/ontology	https://raw.githubusercontent.com/obophe/ontology/main.owl	✓
4034	health	apollo_sv	https://github.com/ApolloLabs/ontology	https://raw.githubusercontent.com/ApolloLabs/ontology/main.owl	✓
4035	microbiology	aro	https://github.com/arpa-card/ontology	https://raw.githubusercontent.com/arpa-card/ontology/main.owl	✓

Figura 20: Interfaz para la Verificación de Datos de Repositorios

3.4.2. Dashboards principales: actividad y métricas

En cuanto al segundo componente principal, los “Paneles de Visualización” ofrecen dos *dashboards* diferenciados. El primero se centra en el análisis de la evolución temporal de los repositorios, y el segundo se concentra en las métricas y estadísticas de las ontologías. Ambos paneles están dotados de filtros que per-

Index	Domain	Ontology	Repository	OWL File URL	Done
4029	health	ado	https://github.com/FraunhoferGesellschaft/QuaRE-Metrics	https://raw.githubusercontent.com/FraunhoferGesellschaft/QuaRE-Metrics/main/QuaRE-Metrics.owl	✓
4030	agriculture	agro	https://github.com/AgriSPO/agriSPO	https://raw.githubusercontent.com/AgriSPO/agriSPO/main/agriSPO.owl	✓
4031	anatomy and development	aism	https://github.com/insecta-informatics/aism	https://raw.githubusercontent.com/insecta-informatics/aism/main/aism.owl	✓
4032	anatomy and development	amphx	https://github.com/EBI-SPARQL/amphx	https://raw.githubusercontent.com/EBI-SPARQL/amphx/main/amphx.owl	✓
4033	phenotype	apo	https://github.com/obophe/obophe	https://raw.githubusercontent.com/obophe/obophe/main/obophe.owl	✓
4034	health	apollo_sv	https://github.com/ApolloProject/apollo	https://raw.githubusercontent.com/ApolloProject/apollo/main/apollo.owl	✓
4035	microbiology	aro	https://github.com/arpa-care/arpa-care	https://raw.githubusercontent.com/arpa-care/arpa-care/main/arpa-care.owl	✓

Figura 21: Interfaz para la Incorporación de Métricas a través del OQuaRE Metrics Web Service

miten al usuario acceder a análisis más específicos o generalizados. La Figura 22 ofrece una captura de pantalla global con el objetivo de ilustrar estas interfaces de usuario. Entremos en detalle en estos paneles de visualización y los diferentes filtros que se pueden aplicar en cada uno de ellos.

Primero comenzemos con el dedicado a la actividad de los repositorios. Para proporcionar un análisis detallado y comprensible de los datos, este panel ofrece diversas métricas y gráficos interactivos (ver Figura 23). Inicialmente, se muestra una serie de tarjetas informativas que incluyen métricas clave como el número de ontologías recolectadas, el número de dominios, el número de ontologías que contienen archivos con extensión OBO, y el número de ontologías con archivos de extensión OWL. Adicionalmente, el panel incluye dos tipos de gráficos. El

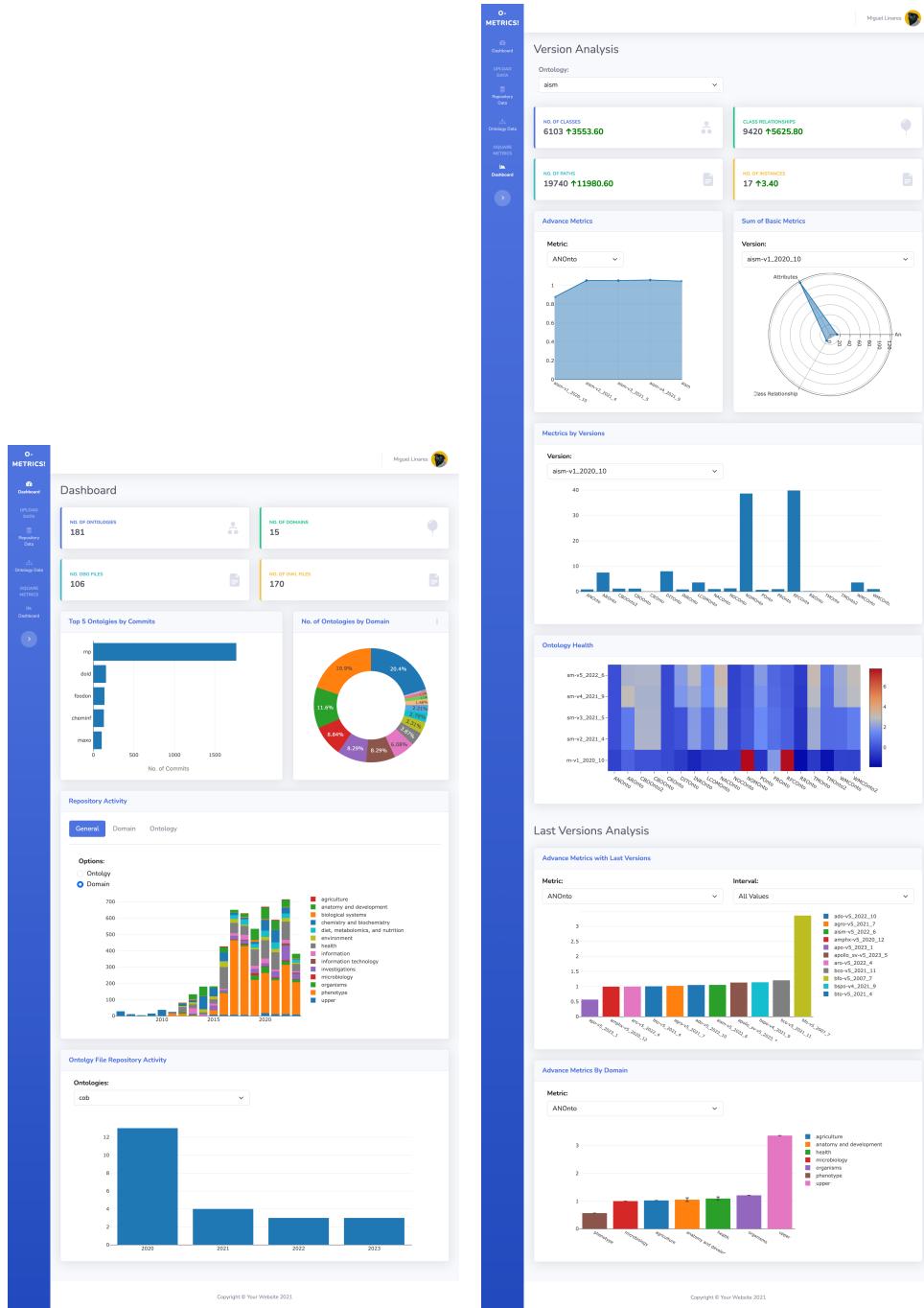


Figura 22: Paneles de Visualización para Actividad del Repositorio (izquierda) y Ontologías (derecha)

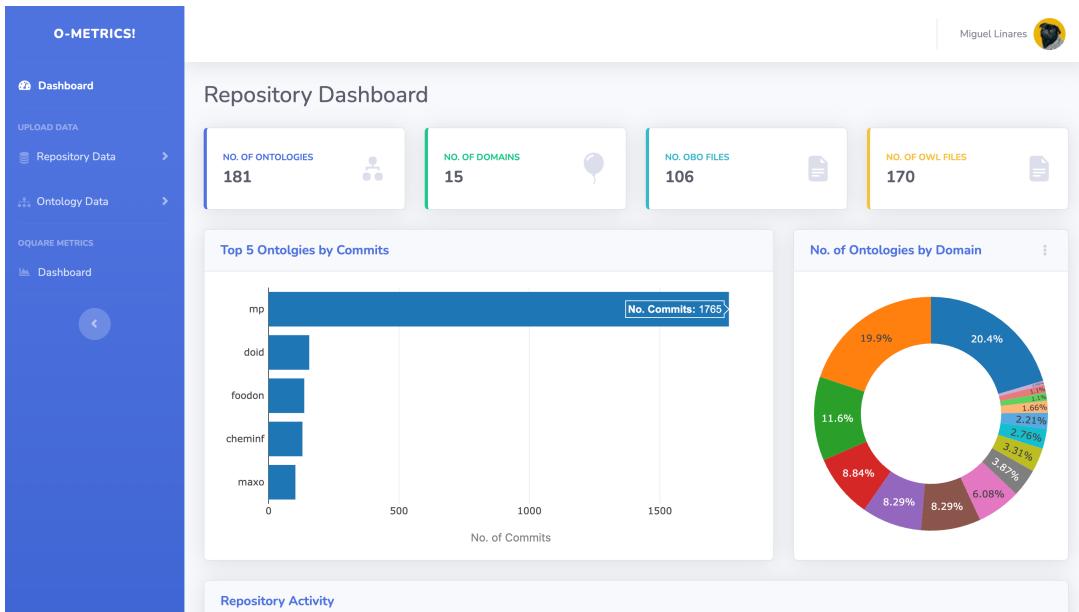


Figura 23: Informaciones generales sobre los repositorios

primero es un gráfico de barras que representa las cinco ontologías con el mayor número de *commits*. El segundo es un gráfico de donas que muestra la distribución de las ontologías por dominio y el porcentaje que cada dominio contribuye al total. Ambos gráficos ofrecen información más detallada al pasar el cursor sobre sus elementos, a través de tarjetas flotantes. Por ejemplo, en la Figura 23 se puede apreciar esta funcionalidad con el gráfico de barras. Además, un menú oculto permite activar la leyenda en el gráfico de dona al presionar los tres puntos, como se muestra en la Figura 24.

En cuanto a la actividad del repositorio, se dispone de una serie de gráficos agrupados por pestañas. El gráfico principal muestra la actividad del repositorio en general, clasificada por año. Este gráfico también permite la aplicación de filtros

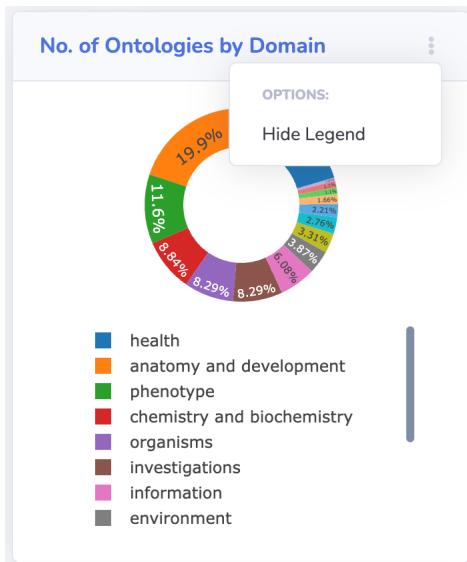


Figura 24: Leyenda de Gráfico de Dona

para visualizar la actividad por ontología o por dominio, como se muestra en la Figura 25.

Los gráficos adicionales, accesibles a través de pestañas, permiten un filtrado más específico por ontologías o dominios, utilizando listas desplegables para la selección. Estos gráficos presentan la misma información que el gráfico general pero permiten una inspección más detallada al limitar la visualización a un solo elemento seleccionado. En la Figura 26 se presentan estos gráficos especializados.

Es importante señalar que la leyenda en los gráficos no solo cumple una función informativa, sino que también actúa como un filtro interactivo. Esto permite a los usuarios seleccionar específicamente las ontologías o dominios que deseen visualizar. Un ejemplo de esta funcionalidad se muestra en la Figura 27, donde se han filtrado varios dominios utilizando la leyenda como filtro.

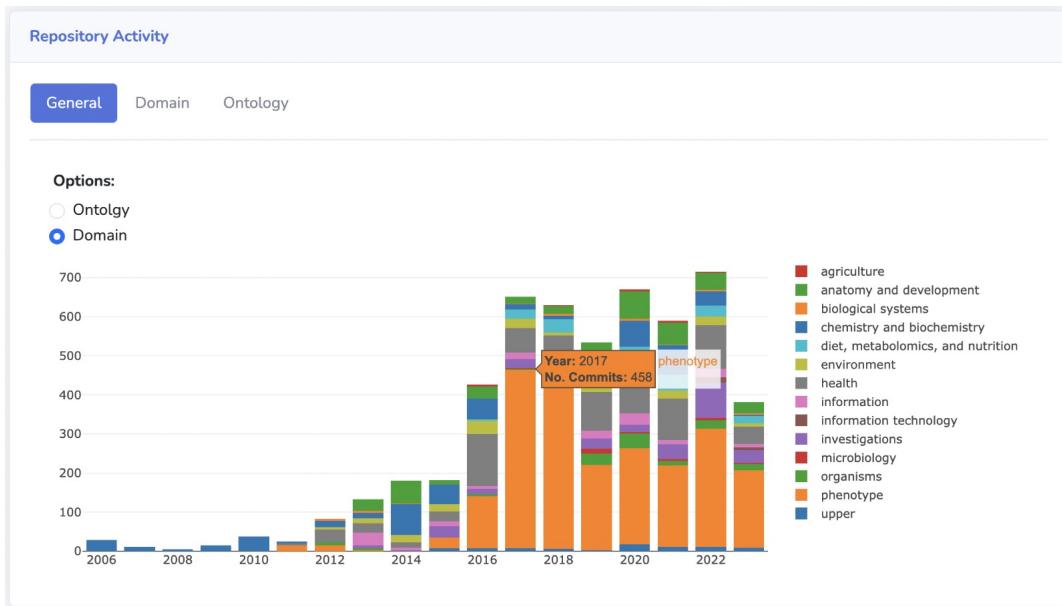


Figura 25: Actividad General de Repositorios

Finalmente, el panel ofrece una visualización detallada de la actividad relacionada con archivos de extensión OWL en los repositorios. En este caso, el gráfico de barras incluye un filtro que permite la selección de una única ontología para su visualización. Esta funcionalidad y cómo se presenta visualmente se puede apreciar en la Figura 28.

El segundo tablero de visualización está dedicado a las métricas básicas y de OQuaRE de las ontologías en su totalidad. El mismo está dividido en dos partes:

- **Análisis por versiones:** todos los gráficos realizados son centrados en todas las versiones o con filtros para seleccionar una en específico.
- **Análisis de las últimas versiones:** todos sus gráficos son para las últimas versiones de cada ontología analizada.

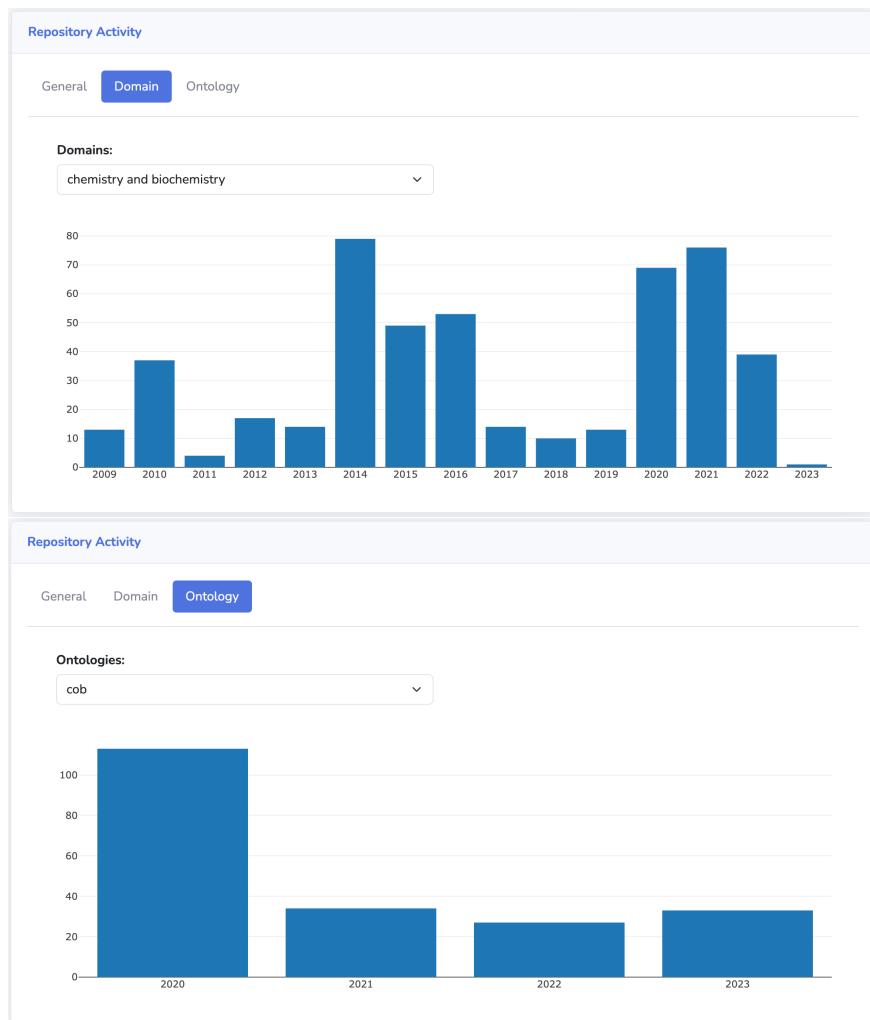


Figura 26: Gráfico de Barras con filtros

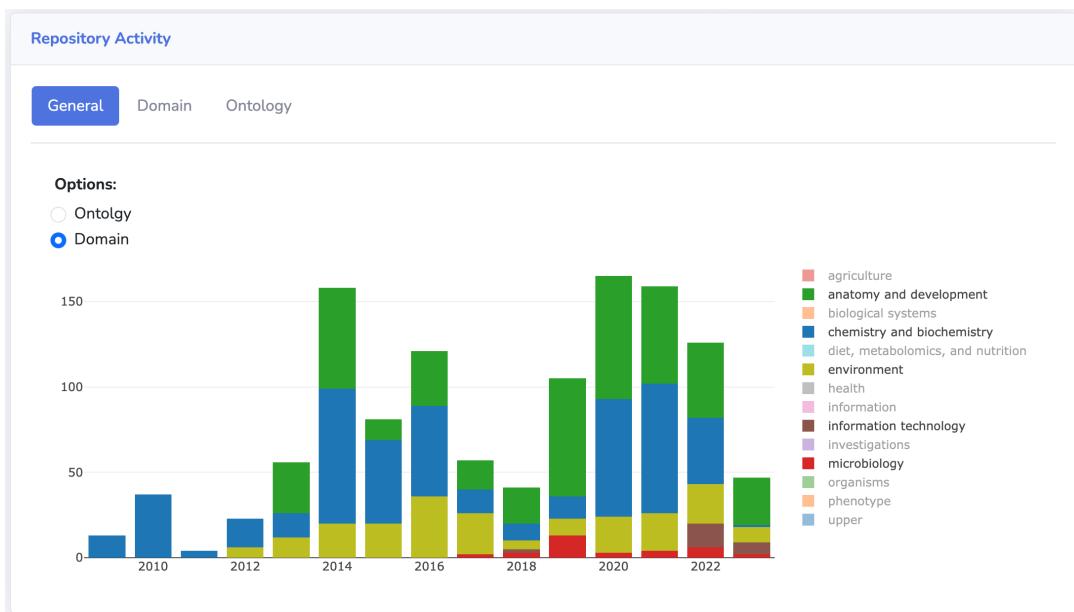


Figura 27: Filtros de Leyenda Aplicados

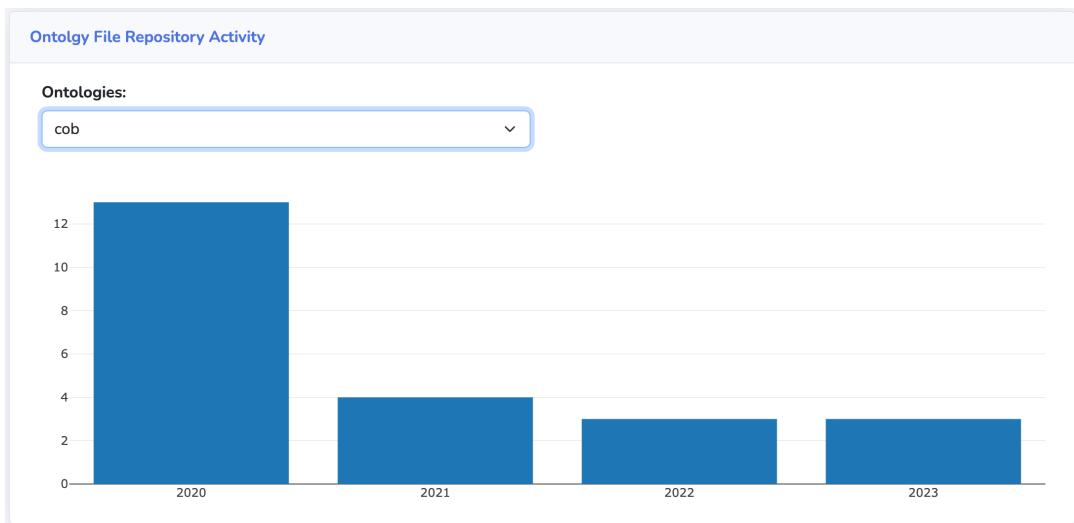


Figura 28: Actividad de archivos OWL

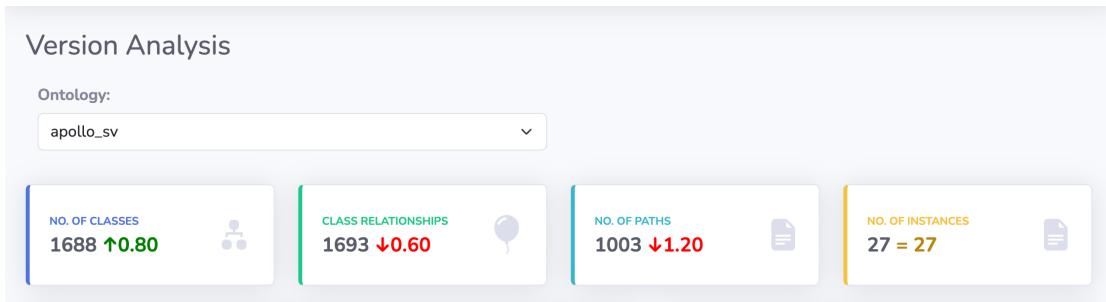


Figura 29: Métricas Básicas de Ontologías

Como se puede observar en la Figura 29, en la primera parte tenemos un filtro global que se utiliza para seleccionar la ontología concreta que queremos analizar. En línea con el *dashboard* anterior, lo primero vemos en tarjetas informativas generales de la última versión de la ontología con la diferencia de sus medias. Si el número está en verde el valor está por encima de la media, si es rojo está por debajo de la media y si es amarillo pues el valor se quedó igual.

El siguiente gráfico (ver Figura 30) tiene un filtro para seleccionar una métrica de OQuaRE concreta y dinámicamente visualizará sus valores para las diferentes versiones de la ontología analizadas. Esto permitirá generar un gráfico de evolución sobre el tiempo. El gráfico de radar de la derecha permite visualizar la relación entre las métricas “Attributes”, “Class Relationship” y “Annot” para las distintas versiones de la ontología seleccionada. Se han seleccionado estas tres métricas porque su relación es de utilidad para los desarrolladores de ontologías.

Continuando con este *dashboard*, tenemos una representación de todas las métricas OQuaRE al mismo tiempo por versión. Podemos ver un ejemplo de esto en la Figura 31. Para finalizar esta primera parte tenemos un mapa de calor (ver

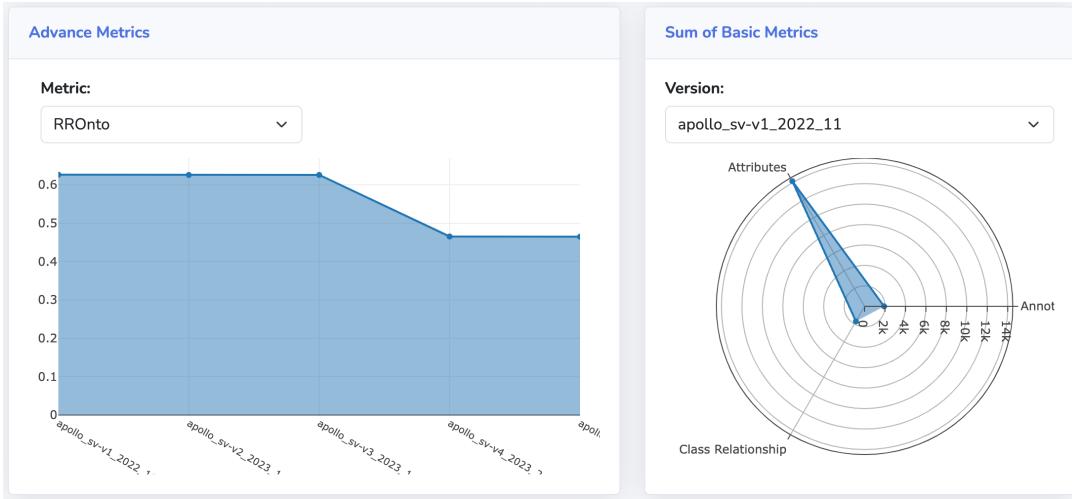


Figura 30: Métricas Básicas de Ontologías

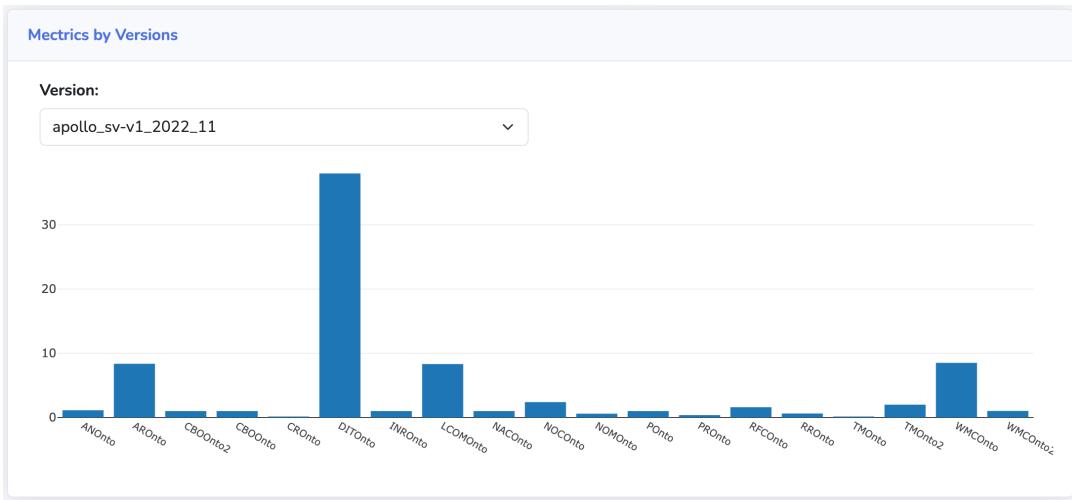


Figura 31: Métricas OQuaRE por versión

Figura 32) que utilizamos calcular la “salud” de cada versión de la ontología seleccionada arriba. Para generar estos resultados calculamos los valores z, a través de la media y la desviación típica, lo que nos deja ver cuales valores son estables y cuales no son constantes.

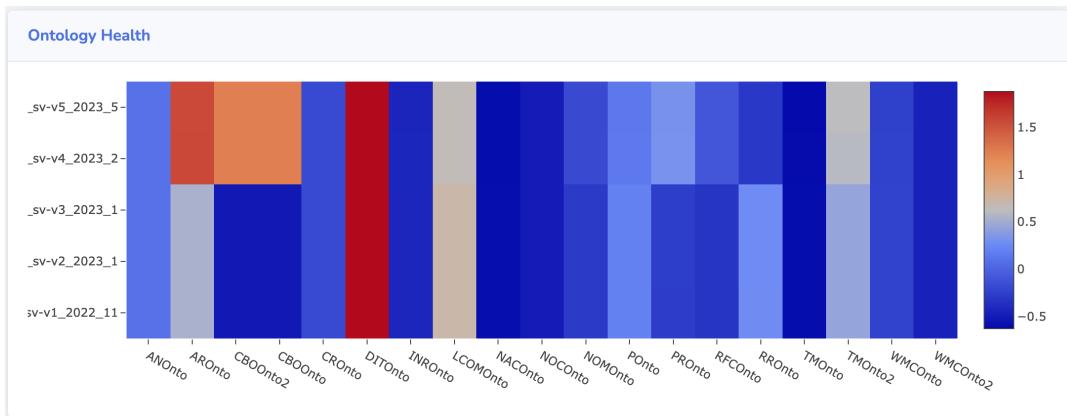


Figura 32: Salud de la Ontología



Figura 33: Métricas OQuaRE con las Últimas Versiones

En la segunda parte del tablero tenemos las dos últimas visualizaciones que no dependen de la ontología seleccionada arriba (ver Figura 33). El primero se utiliza para visualizar una analítica para todas las últimas versiones de las ontologías. Como ya mencioné, se puede seleccionar una métrica pero también se

puede seleccionar rangos de valores para filtrar aún más los valores. Así podemos comparar la distribución una métrica a través de las diferentes ontologías. Esto genera un gráfico que consideramos bastante informativo. La leyenda, aunque parezca reiterativa, hemos decidido preservarla debido a que hace la misma función que las anteriores que ya he mencionado.

El segundo y último gráfico (ver Figura 34) deja ver como se distribuyen los valores de una métrica seleccionada entre los diferentes dominios. Para lograr esto, calculamos la media y la desviación típica (esta se va representada por las líneas negras). Al igual que en gráficos anteriores tenemos tarjetas flotante en los que podemos ver los valores específicos de cada barra y líneas.



Figura 34: Métricas OQuaRE por Dominio

3.5. Código fuente

Aunque no hemos desplegado la herramienta en un servidor público debido a los costes de mantenimiento que esto supondría, el código fuente de O-metrics y los datos recolectados en este trabajo están disponibles en un repositorio privado de Bitbucket [28]. En la Figura 35 se muestra la captura de pantalla de la página principal en la que se puede ver la estructura del código fuente del proyecto desarrollado.

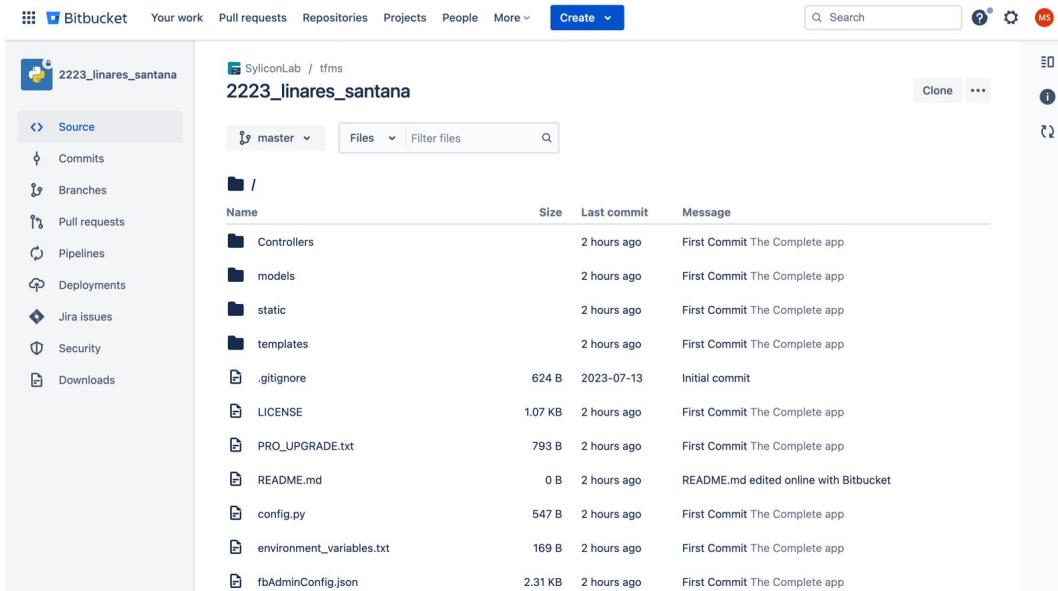


Figura 35: Captura del repositorio de Bitbucket de O-Metrics

Publicar estos resultados en Bitbucket permitiría a aquellos miembros de la comunidad interesados en ella poder solicitar acceso y desplegarla de forma privada o pública en equipos gestionados por ellos. También podrían contribuir a la mejora y actualización de este proyecto como colaboradores.

3.6. Comparación con los métodos tradicionales

La evaluación y gestión de ontologías ha sido tradicionalmente una tarea laboriosa que requería una profunda pericia técnica en las áreas de la Web Semántica, Ciencias de la Información, o Programación, entre otras. En el paradigma tradicional, las métricas se obtienen a menudo de forma manual o mediante scripts *ad-hoc*, lo cual puede resultar en un proceso de análisis discontinuo y en la falta de un seguimiento a lo largo del tiempo. O-Metrics contribuye a mejorar estos procesos desde los siguiente puntos de vista:

- **Automatización y escalabilidad:** una de las principales ventajas de O-Metrics sobre los enfoques tradicionales es su capacidad para automatizar tareas que antes requerían intervención manual. Esto no solo reduce la posibilidad de error humano sino que también escala eficientemente para manejar un gran número de ontologías y repositorios, algo particularmente útil en ambientes donde se manejan múltiples proyectos simultáneamente.
- **Interfaz de usuario amigable:** otro punto fuerte es la interfaz de usuario intuitiva que permite a los usuarios, incluso aquellos con conocimientos técnicos limitados, realizar tareas complejas con unos pocos clics. En contraposición, los métodos tradicionales a menudo requieren que el usuario tenga conocimientos de programación y ontológicos avanzados.
- **Visualización dinámica de datos:** O-Metrics ofrece capacidades avanzadas de visualización que van más allá de lo que se podría lograr con herramientas

tas más rudimentarias. Estas visualizaciones no solo son informativas sino también interactivas, permitiendo a los usuarios explorar datos complejos de una manera más accesible.

- **Integración de múltiples fuentes de datos:** mientras que los métodos convencionales a menudo trabajan en silos, O-Metrics integra múltiples fuentes de datos, como el servicio web de OQuaRE y el API de GitHub en una interfaz unificada. Esto facilita una visión más completa y matizada de la evolución de las ontologías en cuestión.
- **Consistencia y replicabilidad:** finalmente, el enfoque automatizado y estandarizado de O-Metrics asegura una mayor consistencia en la recopilación y análisis de datos, lo que facilita la replicabilidad de los estudios y el *benchmarking* contra métricas predefinidas.

En resumen, O-Metrics representa un avance significativo en la forma en que se pueden evaluar y gestionar las ontologías, ofreciendo un conjunto de herramientas que son al mismo tiempo más potentes y más accesibles que las disponibles en métodos más tradicionales.

4. Conclusiones y Trabajo Futuro

La popularidad de las ontologías y su acceso a través de repositorios públicos a través de Internet ha despertado un interés creciente en el desarrollo de metodologías que permitan el estudio y el aseguramiento de su calidad. El paradigma tradicional de este tipo de análisis se basa en la generación de métricas que se obtienen esporádicamente o de forma manual. En este proyecto hemos diseñado una metodología, que ha sido implementado en la herramienta O-Metrics, que permite automatizar este proceso y adaptarlo a repositorios concretos.

O-Metrics es una herramienta práctica que ayuda a simplificar la gestión y el análisis de ontologías y sus respectivos repositorios. O-Metrics ofrece implementa en primer lugar un proceso de recolección de datos y cálculo de métricas, que posteriormente pueden ser dinámicamente consultadas a través de diferentes *dashboards* que incluyen una variedad de gráficos y filtros. Los usuarios pueden obtener una vista general o detallada de sus repositorios, ontologías y dominios. Las funcionalidades como tarjetas flotantes y leyendas interactivas añaden un nivel adicional de utilidad a la plataforma. Además, su diseño orientado a la usabilidad hace que sea accesible para una amplia gama de usuarios, proporcionando una forma más eficiente de interactuar con estos tipos de datos.

Hemos aplicado la herramienta para recolectar datos del repositorio OBO Foundry, lo que nos ha permitido analizar un total de 181 ontologías categorizadas en 15 dominios diferentes. Se ha analizado la evolución en el tiempo de estas ontologías explotando la información almacenada en el sistema de control de

versiones GitHub. Todo esto ha permitido confirmar la validez de la hipótesis originalmente planteada: *“La implementación de una herramienta de procesamiento y visualización interactiva que utilice ontologías provenientes de repositorios como BioPortal o el OBO Foundry permitirá una mejor comprensión y seguimiento de la evolución de las ontologías y sus características a lo largo del tiempo. Esta herramienta semiautomática proporcionará insights valiosos y mejorará la eficiencia en los estudios de ontología, superando las limitaciones de los métodos manuales existentes”*.

La implementación de esta herramienta ha requerido integrar tecnologías y bibliotecas útiles que giran entorno a su lenguaje principal Python. Entre las tecnologías integradas destacamos el papel del API de GitHub, el Servicio Web de OQuaRE, el *framework* Flask y Plotly para creación de sitios web y su integración con el *stack* de desarrollo web: HTML5, CSS y JS.

Aunque O-Metrics ha sido diseñada para integrar y analizar datos de ontologías desde múltiples fuentes, solo hemos registrado la información de un repositorio (el OBO Foundry). Sin embargo, al automatizar la recopilación y procesamiento de datos, no solo hemos optimizado el tiempo de análisis, sino que también hemos mejorado la calidad del mismo al eliminar errores manuales inherentes a los métodos tradicionales. Además, la capacidad de la herramienta para ofrecer visualizaciones interactivas ha añadido un nuevo nivel de profundidad al análisis. Los usuarios pueden explorar los datos de forma más intuitiva, lo que fomenta una mejor comprensión de las complejidades y las relaciones intrínsecas entre las diferentes métricas y características de las ontologías. Esto se traduce en una

capacidad mejorada para tomar decisiones informadas y realizar investigaciones más robustas en el campo de la ontología.

También vale la pena destacar la flexibilidad y escalabilidad de nuestra solución. El diseño modular y la selección cuidadosa de tecnologías y bibliotecas permiten futuras extensiones y mejoras, lo cual es especialmente relevante dado el rápido avance del campo de las ontologías y la Ciencia de Datos en general.

En resumen, nuestra herramienta la mejora de los métodos existentes para el estudio de las ontologías almacenadas en repositorios. Esperamos que este trabajo abra nuevas vías para una investigación más avanzada y completa, siendo una herramienta de relevancia en este ámbito académico y práctico.

4.1. Discusión y limitaciones

O-Metrics se erige como una plataforma modular que integra tanto desarrollos propios como herramientas previamente existentes para optimizar su funcionalidad. Un componente clave en este ecosistema es el servicio web de OQuaRE de la Universidad de Murcia, que ha sido muy útil para acelerar los tiempos de procesamiento y en minimizar la carga sobre los servidores de usuario. Esto se ha demostrado particularmente útil como una alternativa más eficiente a la librería de Python, OWLReady2, que, aunque robusta, demanda una significativa capacidad de procesamiento. Además, las características y métricas calculadas por OQuaRE son más extensas que las que podemos calcular con OWLReady2. Por estos motivos descartamos el uso de OWLReady2 y externalizamos el cálculo de métricas en un tercero.

Adicionalmente, la API de GitHub desempeña un papel crucial, ya que facilita la extracción de una parte sustancial de nuestros datos. Su presencia ha sido vital para el proyecto, evitando la necesidad de recurrir a técnicas de *Web Scraping* más arduas e inciertas. En el aspecto de visualización, la librería Plotly.js contribuye con gráficos altamente interactivos y estéticamente atractivos. En resumen, O-Metrics es el resultado de una colaboración sinérgica entre diversas tecnologías y recursos, convergiendo para formar una plataforma integral y altamente funcional.

Aunque este estudio es profundo y de gran alcance, no está exento de ciertas limitaciones que deben considerarse cuidadosamente para una interpretación precisa y la correcta aplicación de los resultados obtenidos. Las limitaciones más significativas son:

- **Restricciones en la fuente de datos:** el análisis en este estudio se limita a las ontologías contenidas en el repositorio seleccionado, el OBO Foundry. Aunque este repositorio es ampliamente reconocido y utilizado en la comunidad científica, no abarca la totalidad de las ontologías disponibles en el ámbito académico y profesional. En consecuencia, nuestro análisis puede no reflejar las tendencias y comportamientos que pueden estar presentes en ontologías que no están incluidas en esta comunidad.
- **Dependencia de las herramientas de Python:** la utilización de Python y sus librerías para el análisis de los datos y la generación de visualizaciones introduce las limitaciones inherentes a estas herramientas.
- **Aspectos subjetivos en el análisis de conformidad:** aunque nos esforza-

mos por proporcionar un análisis objetivo y cuantitativo de la conformidad con los principios de construcción de ontologías, es importante reconocer que este aspecto del proyecto puede estar sujeto a interpretación y debate. Además, nos hemos centrado principalmente en explotar el principio de versionado, delegando en las métrica de OQuaRE el seguimiento del resto de principios.

- **Limitación en el rastreo de cambios:** a pesar de que nuestros métodos semiautomáticos están diseñados para capturar y describir meticulosamente las modificaciones entre distintas versiones de una ontología, algunas URLs no han podido ser procesadas debido a dependencias de los ficheros fuente que no pueden ser resueltas.
- **Dependencia del mantenimiento de los repositorios:** nuestro análisis depende de la continuidad en el mantenimiento y actualización de los repositorios. Interrupciones o cambios en las prácticas de mantenimiento pueden afectar la disponibilidad y la calidad de los datos.

4.2. Trabajo futuro

La plataforma O-Metrics representa un avance significativo en la visualización y análisis de datos relacionados con repositorios, ontologías y dominios. Utilizando un enfoque modular y aprovechando herramientas existentes como el Servicio Web OQuaRE, la API de GitHub y la biblioteca Plotly.js, hemos logrado crear una herramienta integral que facilita una amplia gama de actividades analíticas.

Sin embargo, como cualquier proyecto de software en evolución, hay oportunidades para expandir su alcance y mejorarlo.

1. **Desplegar el proyecto en un servidor accesible a través de Internet:** sería interesante conseguir financiación para cubrir los costes derivados del despliegue de la herramienta web en un dominio público.
2. **Mejora de algoritmos de análisis:** actualmente, el Servicio Web OQuaRE es un componente crucial para el procesamiento eficiente de datos en O-Metrics. A medida que las tecnologías de análisis de datos avanzan, hay oportunidades para incorporar algoritmos más rápidos y precisos que podrían mejorar aún más la eficiencia del sistema.
3. **Integración de otras APIs:** aunque la API de GitHub ha sido vital para extraer un gran volumen de datos, la integración con otras APIs como las de GitLab o Bitbucket podría ofrecer una perspectiva más holística del ecosistema de desarrollo de software.
4. **Automatización del proceso de actualización:** la incorporación de un sistema de actualización automática podría mantener las métricas y gráficos al día con mínima intervención manual, lo que facilitaría el mantenimiento a largo plazo de la plataforma.
5. **Personalización de gráficos:** la biblioteca Plotly.js ofrece un conjunto robusto de opciones de visualización, pero hay margen para ofrecer una mayor personalización, permitiendo a los usuarios adaptar las visualizaciones

de datos a sus necesidades específicas.

6. **Ampliación de métricas de repositorio:** el enfoque actual se centra en ontologías y dominios. Sin embargo, podrían incluirse métricas adicionales como la actividad de los colaboradores o la calidad del código para enriquecer el análisis.
7. **Implementación de Aprendizaje Automático:** la aplicación de técnicas de aprendizaje automático podría permitir a la plataforma prever tendencias futuras en base a datos históricos, lo que añadiría una nueva dimensión de utilidad para los usuarios. Por igual se podría integrar un análisis de clusters ya que notamos ciertos grupos que se formaban dentro de las ontologías.
8. **Soporte multilingüe:** dado que el desarrollo de software es una actividad global, la inclusión de soporte multilingüe podría ampliar el alcance de la plataforma a comunidades de desarrolladores no anglófonos.
9. **Análisis de tiempo de procesamiento:** dentro de nuestros algoritmos de procesamiento recopilamos información del tiempo que dura cada ontología en ser procesada en cada paso que se da. La idea sería poder sacar provecho de estos tiempos y realizar un análisis con esa información.
10. **Accesibilidad y UX:** con el fin de hacer que O-Metrics sea más inclusivo, podrían implementarse características de accesibilidad, como lectores de pantalla y opciones de alto contraste, para facilitar su uso a personas con discapacidades.

En resumen, aunque O-Metrics ha demostrado ser una herramienta imprescindible para el análisis de repositorios, ontologías y dominios, hay varias direcciones prometedoras para futuras investigaciones y desarrollos. Estas expansiones y mejoras no solo aumentarían las capacidades de la plataforma, sino que también la harían más accesible y útil para una gama más amplia de usuarios. Con una ejecución cuidadosa de estos trabajos futuros, O-Metrics tiene el potencial de convertirse en una herramienta estándar en la industria para el análisis y la visualización de datos relacionados con el desarrollo de ontologías.

Referencias

- [1] ISO/IEC 25000:2005 Software Engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE. ISO/IEC 25000:2005, 2005. ISO/IEC Standard.
- [2] Use Postman for invoking the web service. <https://semantics.inf.um.es/ontology-metrics/doc-tutorial11.html>, 2018. Accessed: 2023-09-13.
- [3] Antonella Soledad Antonini, María Luján Gánuza, and Silvia Mabel Castro. VISUEL-A Web Dynamic Dashboard for DataVisualization. *Journal of Computer Science Technology*, 22(1), 2022.
- [4] Graciela Barchini, Margarita Álvarez, Susana Herrera, and Melina Trejo. El rol de las ontologías en los SI. *Revista Ingeniería Informática*, 14:1–12, 2007.
- [5] Jonathan Bard, Seung Y Rhee, and Michael Ashburner. An ontology for cell types. *Genome biology*, 6:1–5, 2005.
- [6] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, 284:28–37, 5 2001.
- [7] Bokeh. Bokeh. <http://bokeh.org/>. Accessed: 2023-09-12.

- [8] Bootstrap. Build fast, responsive sites with Bootstrap. <https://getbootstrap.com/>. Accessed: 2023-09-12.
- [9] Helder Da Rocha. *Learn Chart.js: Create interactive visualizations for the web with chart.js* 2. Packt Publishing Ltd, 2019.
- [10] Astrid Duque-Ramos, Jesualdo Fernandez-Breis, Robert Stevens, and Nathalie Aussenac-Gilles. OQuaRE: A SQuaRE-based approach for evaluating the quality of ontologies. *Journal of Research and Practice in Information Technology*, 43:159–176, 05 2011.
- [11] Astrid Duque-Ramos, Jesualdo Fernandez-Breis, Robert Stevens, and Nathalie Aussenac-Gilles. OQuaRE: A SQuaRE-based approach for evaluating the quality of ontologies. *Journal of Research and Practice in Information Technology*, 43:159–176, 05 2011.
- [12] GitHub. GitHub API REST Documentation. <https://docs.github.com/en/rest/guides/getting-started-with-the-rest-api?apiVersion=2022-11-28>. Accessed: 2023-09-07.
- [13] GitHub. GitHub Home Page. <https://www.github.com>. Accessed: 2023-09-07.
- [14] Thomas R Gruber. Toward principles for the design of ontologies used for knowledge sharing? *International Journal of Human-Computer Studies*, 43:907–928, 1995.

- [15] Tom Gruber. What is an Ontology. https://queksiewkhoon.tripod.com/ontology_01.pdf, 1993. Accessed: 2023-09-12.
- [16] Nicola Guarino. *Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy*, volume 46. IOS press, 1998.
- [17] Asunción Gómez-Pérez, Mariano Fernández-López, and Oscar Corcho. *Ontological Engineering*. Springer-Verlag, 1 edition, 2004.
- [18] A. Hogan, M. Egan, and N.F. Noy. Visualizing the evolution of a large biomedical ontology using a Linked Data approach. *Bioinformatics*, 32(10):1535–1544, 2016.
- [19] Michal Hron and Nikolaus Obwegeser. Scrum in practice: an overview of Scrum adaptations. *Proceedings of the 51st Hawaii International Conference on System Sciences*, 2018.
- [20] IBM. What is a web service? <https://www.ibm.com/docs/en/cics-ts/5.1?topic=services-what-is-web-service>, Mar 2021. Accessed: 2023-09-13.
- [21] Rebecca Jackson, Nicolas Matentzoglu, James A Overton, Randi Vita, James P Balhoff, Pier Luigi Buttigieg, Seth Carbon, Melanie Courtot, Alexander D Diehl, Damion M Dooley, William D Duncan, Nomi L Harris, Melissa A Haendel, Suzanna E Lewis, Darren A Natale, David Osumi-Sutherland,

Alan Ruttenberg, Lynn M Schriml, Barry Smith, Christian J Stoeckert Jr., Nicole A Vasilevsky, Ramona L Walls, Jie Zheng, Christopher J Mungall, and Bjoern Peters. OBO Foundry in 2021: operationalizing open data principles to evaluate ontologies. *Database*, 2021, 9 2021.

[22] Lamy Jean-Baptiste. *Ontologies with Python*. Apress, 1 edition, 2021.

[23] Joseph Kuan. *Learning Highcharts*. Packt Publishing Ltd, 2012.

[24] Addepalli Lavanya, Lokhande Gaurav, Sakinam Sindhuja, Hussain Seam, Mookerjee Joydeep, Vamsi Uppalapati, Waqas Ali, and Vidya Sagar SD. Assessing the Performance of Python Data Visualization Libraries: A Review. *International Journal of Computer Engineering in Research Trends*, 2023.

[25] Meng-Tse Lee, Fong-Ci Lin, Szu-Ta Chen, Wan-Ting Hsu, Samuel Lin, Tzer-Shyong Chen, Feipei Lai, and Chien-Chang Lee. Web-based Dashboard for the Interactive Visualization and Analysis of National Risk Standardized Mortality Rates of Sepsis in the US. <https://sepsismap.shinyapps.io/index2/>. Accessed: 2023-09-12.

[26] Meng-Tse Lee, Fong-Ci Lin, Szu-Ta Chen, Wan-Ting Hsu, Samuel Lin, Tzer-Shyong Chen, Feipei Lai, and Chien-Chang Lee. Web-based dashboard for the interactive visualization and analysis of national risk-standardized mortality rates of sepsis in the US. *Journal of Medical Systems*, 44:1–9, 2020.

- [27] Paulo Vélez León. ¿ Ontología u ontologías? *Disputatio. Philosophical Research Bulletin*, 4:299–339, 2015.
- [28] Miguel Linares Santana. Repositorio O-Metrics. https://bitbucket.org/syliconlab/2223_linares_santana/src/master/. Accessed: 2023-09-13.
- [29] James Malone, Ele Holloway, Tomasz Adamusiak, Misha Kapushesky, Jie Zheng, Nikolay Kolesnikov, Anna Zhukova, Alvis Brazma, and Helen Parkinson. Modeling sample variables with an Experimental Factor Ontology. *Bioinformatics*, 26(8):1112–1118, 03 2010.
- [30] Matplotlib. Matplotlib. <https://matplotlib.org/>. Accessed: 2023-09-12.
- [31] Elijah Meeks. *D3.js in Action: Data visualization with JavaScript*. Simon and Schuster, 2017.
- [32] Mozilla. CSS. <https://developer.mozilla.org/es/docs/Web/CSS>. Accessed: 2023-09-12.
- [33] Mozilla. HTML: Lenguaje de etiquetas de hipertexto. <https://developer.mozilla.org/es/docs/Web/HTML>. Accessed: 2023-09-12.
- [34] M.A. Musen, N.F. Noy, N. Shah, R. Stevens, S.W. Tu, and C. Welty. Challenges in managing and evolving large ontologies. *Journal of Biomedical Informatics*, 59:11–22, 2015.

- [35] Mark A Musen and Protégé Team. The Protégé Project: A Look Back and a Look Forward. *AI Matters*, 1:4–12, 6 2015.
- [36] Robert Neches, Richard E Fikes, Tim Finin, Thomas Gruber, Ramesh Patil, Ted Senator, and William R Swartout. Enabling technology for knowledge sharing. *AI magazine*, 12(3):36–36, 1991.
- [37] N F Noy, M Sintek, S Decker, M Crubézy, R W Fergerson, and M A Musen. Creating Semantic Web contents with Protege-2000. *IEEE Intelligent Systems*, 16:60–71, 2001.
- [38] Natalya F Noy, Nigam H Shah, Patricia L Whetzel, Benjamin Dai, Michael Dorf, Nicholas Griffith, Clement Jonquet, Daniel L Rubin, Margaret-Anne Storey, Christopher G Chute, and Mark A Musen. BioPortal: ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Res*, 37:W170–3, 5 2009.
- [39] Numpy. Numpy. <https://numpy.org/>. Accessed: 2023-09-12.
- [40] University of Manchester. OWL API Documentation. <https://github.com/owlcs/owlapi/wiki/Documentation>. Accessed: 2023-09-12.
- [41] Owlready2. Welcome to Owlready2’s documentation! <https://owlready2.readthedocs.io/en/v0.42/>. Accessed: 2023-09-13.
- [42] Pandas. Pandas. <https://pandas.pydata.org/>. Accessed: 2023-09-12.

- [43] Daniel Petras, Vanessa V Phelan, Deepa Acharya, Andrew E Allen, Allegra T Aron, Nuno Bandeira, Benjamin P Bowen, Deirdre Belle-Oudry, Simon Boecker, Dale A Cummings Jr, et al. GNPS Dashboard. <https://dashboard.gnps2.org/>. Accessed: 2023-09-12.
- [44] Daniel Petras, Vanessa V Phelan, Deepa Acharya, Andrew E Allen, Allegra T Aron, Nuno Bandeira, Benjamin P Bowen, Deirdre Belle-Oudry, Simon Boecker, Dale A Cummings Jr, et al. GNPS Dashboard: collaborative exploration of mass spectrometry data in the web browser. *Nature methods*, 19(2):134–136, 2022.
- [45] Plotly. Plotly. <https://plotly.com/>. Accessed: 2023-09-12.
- [46] María Poveda-Villalón, Asunción Gómez-Pérez, and Mari Carmen Suárez-Figueroa. Oops!(ontology pitfall scanner!): An on-line tool for ontology evaluation. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 10(2):7–34, 2014.
- [47] Pallets Projects. Welcome to Flask. <https://flask.palletsprojects.com/en/2.3.x/#>. Accessed: 2023-09-12.
- [48] D. Teja Santosh, V. Kakulapati, and K. Basavaraju. Ontology-based sentimental knowledge in predicting the product recommendations: A data science approach. *Journal of Discrete Mathematical Sciences and Cryptography*, 23(1):1–18, 2020.

- [49] L. M. Schriml, E. Mitraka, J. Munro, B. Tauber, M. Schor, L. Nickle, et al. Human Disease Ontology 2018 update: classification, content and workflow expansion. *Nucleic acids research*, 47(D1):D955–D962, 2019.
- [50] L. M. Schriml, E. Mitraka, B. Tauber, M. Schor, L. Nickle, V. Felix, et al. Human Disease Ontology: updates from COVID-19 to Phenotypes. *Nucleic acids research*, 49(D1):D1037–D1043, 2021.
- [51] SciPy. Scipy. <https://scipy.org/>. Accessed: 2023-09-12.
- [52] Seaborn. Seaborn. <https://seaborn.pydata.org/>. Accessed: 2023-09-12.
- [53] Akshansh Sharma, Firoj Khan, Deepak Sharma, Sunil Gupta, and FY Student. Python: the programming language of future. *Int. J. Innovative Res. Technol*, 6(2):115–118, 2020.
- [54] Shiny. Easy web apps for data science without the compromises. <https://shiny.posit.co/>. Accessed: 2023-09-12.
- [55] Barry Smith. Ontology. *The furniture of the world*, pages 47–68, 2012.
- [56] Barry Smith, Michael Ashburner, Cornelius Rosse, Jonathan Bard, William Bug, Werner Ceusters, Louis J Goldberg, Karen Eilbeck, Amelia Ireland, Christopher J Mungall, OBI Consortium, Philippe Leontis Neocles, Rocca-Serra, Alan Ruttenberg, Susanna-Assunta Sansone, Richard H Scheuer-mann, Nigam Shah, Patricia L Whetzel, and Suzanna Lewis. The OBO

- Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nat Biotechnol*, 25:1251–1255, 11 2007.
- [57] KR Srinath. Python—the fastest growing programming language. *International Research Journal of Engineering and Technology*, 4(12):354–357, 2017.
- [58] Igor Stančin and Alan Jović. An overview and comparison of free Python libraries for data mining and big data analysis. In *2019 42nd International convention on information and communication technology, electronics and microelectronics (MIPRO)*, pages 977–982. IEEE, 2019.
- [59] Boontawee Suntisrivaraporn. *Polynomial-Time Reasoning Support for Design and Maintenance of Large-Scale Biomedical Ontologies*. PhD thesis, Technische Universität Dresden, 08 2023.
- [60] S. E. Sánchez López. Modelo de indexación de formas en sistemas VIR basado en ontologías. Master’s thesis, Universidad de las Américas Puebla, Puebla, México, mayo 2007.
- [61] Samir Tartir, Ismailcem Arpinar, Michael Moore, Amit Sheth, and Boanerges Aleman-Meza. OntoQA: Metric-Based Ontology Quality Analysis. 11 2005.
- [62] LV Tulchak and AO Marchuk. History of Python. <http://ir.lib.vntu.edu.ua/bitstream/handle/123456789/10471/461.pdf>. Accessed: 2023-09-13.