



UNIVERSIDAD
DE MURCIA



Facultad de Biología
Universidad de Murcia

CONSTRUCCIÓN, EXPLOTACIÓN Y PUBLICACIÓN DE UNA BASE DE CONOCIMIENTO SEMÁNTICA SOBRE EL VIRUS DE LA INMUNODEFICIENCIA HUMANA (VIH-1)

PRÁCTICA ENTREGABLE
EXPLOTACIÓN SEMÁNTICA DE DATOS

Daniel González Palazón

Máster en Bioinformática
Facultad de Biología – Universidad de Murcia
Curso 2024-2025

Índice

1. Introducción	2
1.1. El Virus de la Inmunodeficiencia Humana (VIH-1) y su Ciclo Replicativo .	2
1.2. Terapia Antirretroviral y Dianas Moleculares	3
2. Objetivos	4
3. Diseño del Grafo y Fuentes de Datos	4
3.1. Clases e Instancias Definidas	4
3.2. Predicados (Object Properties) y Relaciones Modeladas	5
4. Construcción del Grafo de Conocimiento con Protégé	6
4.1. Metodología de Construcción	6
4.2. Métricas Finales de la Ontología	7
5. Despliegue y Publicación de Datos (Arquitectura FAIR)	8
5.1. Preparación del Grafo para Publicación	8
5.2. Despliegue en Servidor <code>dayhoff</code>	8
5.3. Endpoints y Recursos Disponibles	9
6. Explotación de Datos con Consultas SPARQL	10
6.1. Consulta 1: Trazabilidad Completa desde el Fármaco hasta el Virus	10
6.2. Consulta 2: Dianas Derivadas de Poliproteínas y sus Fármacos	11
6.3. Consulta 3: Fármacos que NO son Inhibidores de Proteasa	12
6.4. Consulta 4: Dianas de Fármacos Específicos usando VALUES	12
6.5. Consulta 5: Resumen de Fármacos por Diana Viral (Agregación)	13
7. Conclusiones	14
8. Ubicación de los Ficheros del Proyecto	15
Bibliografía	15

1. Introducción

1.1. El Virus de la Inmunodeficiencia Humana (VIH-1) y su Ciclo Replicativo

El Virus de la Inmunodeficiencia Humana de tipo 1 (VIH-1) es el agente retroviral causante del Síndrome de Inmunodeficiencia Adquirida (SIDA), una de las pandemias más significativas de la historia reciente. Como retrovirus, su característica definitoria es la capacidad de convertir su genoma de ARN en ADN mediante un proceso de transcripción inversa, para luego integrarlo de forma permanente en el genoma de la célula huésped.

El ciclo de vida del VIH-1 es un proceso de múltiples etapas que depende de una serie de proteínas virales y humanas clave, las cuales constituyen las principales dianas de la terapia antirretroviral. El ciclo comienza con la unión de la glicoproteína de superficie viral gp120 al receptor primario CD4 presente en la superficie de los linfocitos T colaboradores y otras células inmunitarias. Esta primera interacción provoca un cambio conformacional en gp120 que expone un sitio de unión para un correceptor, que puede ser CCR5 o CXCR4. La unión al correceptor desencadena la acción de la glicoproteína transmembrana gp41, la cual media la fusión de la envoltura viral con la membrana celular, liberando el contenido del virus en el citoplasma. Una vez en el interior, la enzima viral Transcriptasa Inversa (RT) cataliza la síntesis de una copia de ADN de doble cadena a partir del genoma de ARN viral. Posteriormente, la enzima Integrasa (IN) se encarga de insertar este ADN proviral en el genoma de la célula huésped. El paso final de la replicación es la maduración, donde la Proteasa (PR) viral procesa las poliproteínas precursoras Gag-Pol y Env, cortándolas para liberar las proteínas estructurales y enzimas maduras que formarán los nuevos viriones infecciosos (Engelman & Cherepanov, 2012). Este ciclo de vida, con sus pasos enzimáticos específicos, se resume en la Figura 1.

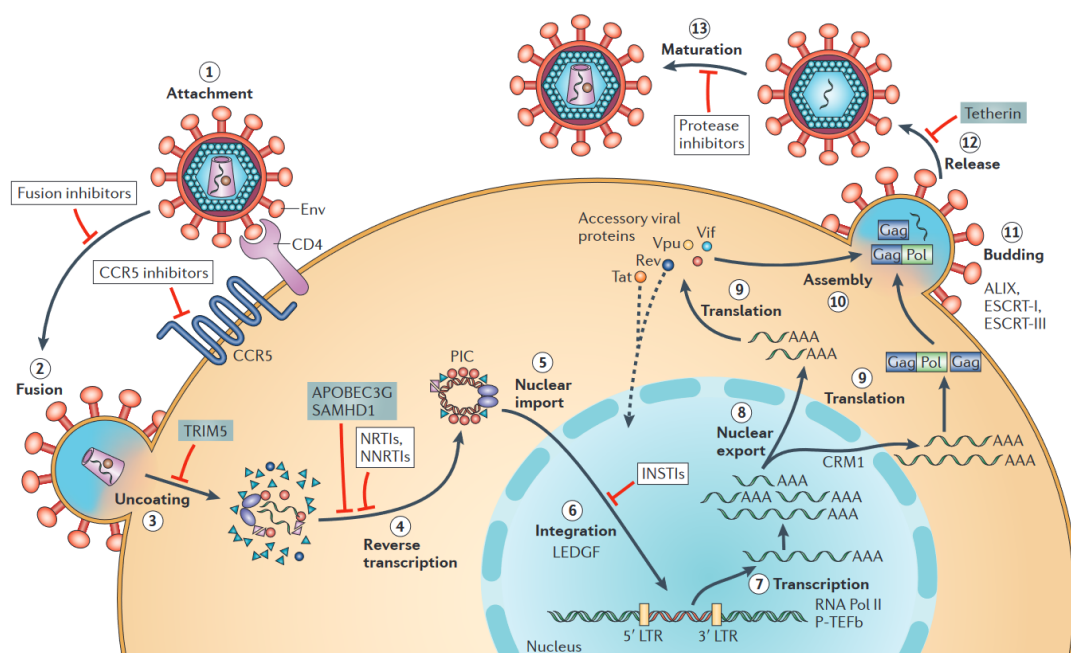


Figura 1: Esquema del ciclo replicativo del VIH-1. Se destacan las proteínas virales clave que actúan como dianas terapéuticas en diferentes etapas del ciclo. (Adaptado de Engelman & Cherepanov, 2012).

1.2. Terapia Antirretroviral y Dianas Moleculares

La dependencia del VIH-1 de sus propias enzimas hace que estas sean dianas excelentes para la inhibición farmacológica. La estrategia terapéutica actual, conocida como Terapia Antirretroviral de Gran Actividad (TARGA), se basa en la combinación de fármacos de diferentes clases para atacar simultáneamente varias etapas del ciclo viral, suprimiendo la replicación y minimizando el desarrollo de resistencias (De Clercq & Li, 2016). El objetivo de este trabajo es modelar las complejas relaciones entre estas entidades (virus, proteínas y fármacos) en un grafo de conocimiento semántico.

Las principales clases de fármacos modeladas en esta ontología, basadas en su mecanismo de acción, se resumen en la Tabla 1 y se describen a continuación (De Clercq & Li, 2016):

- **Inhibidores de la Transcriptasa Inversa (ITR):** Análogos de nucleós(t)idos (NRTI) como la Zidovudina, y no nucleósidos (NNRTI) como el Efavirenz. Su diana es la Transcriptasa Inversa.
- **Inhibidores de la Proteasa (IP):** Fármacos como el Ritonavir que bloquean la maduración de las nuevas partículas virales al inhibir la Proteasa viral.
- **Inhibidores de la Integrasa (INSTI):** Fármacos como el Raltegravir que impiden la inserción del ADN proviral al inhibir la Integrasa.
- **Inhibidores de Entrada y Fusión:** Antagonistas de correceptores como el Maraviroc (diana: CCR5 humano) e inhibidores de la fusión como la Enfuvirtida (diana: gp41 viral).

Cuadro 1: Resumen de las principales clases de fármacos antirretrovirales, con ejemplos de individuos modelados en la ontología y sus dianas moleculares primarias.

Clase de Fármaco	Ejemplo de Fármaco	Diana Molecular Primaria
Inhibidor de la Transcriptasa Inversa Análogo de Nucleósido (NRTI)	Zidovudina (AZT)	Transcriptasa Inversa (RT) del VIH-1
Inhibidor de la Transcriptasa Inversa No Nucleósido (NNRTI)	Efavirenz	Transcriptasa Inversa (RT) del VIH-1
Inhibidor de la Proteasa (PI)	Ritonavir	Proteasa (PR) del VIH-1
Inhibidor de la Integrasa (INSTI)	Raltegravir	Integrasa (IN) del VIH-1
Inhibidor de Entrada (Antagonista de CCR5)	Maraviroc	Proteína Humana CCR5
Inhibidor de Fusión	Enfuvirtida	Glicoproteína gp41 del VIH-1

La correcta representación de estas entidades y sus interacciones es fundamental para crear una herramienta computacional que pueda ser utilizada para la consulta de información y la generación de nuevas hipótesis en la investigación biomédica.

2. Objetivos

El objetivo de este trabajo es la construcción y publicación de un grafo de conocimiento que modele, integre y facilite la recuperación de información sobre el virus VIH-1, sus componentes proteicos clave y la farmacología de la terapia antirretroviral. La implementación se realizará siguiendo los principios FAIR (Findable, Accessible, Interoperable, Reusable) y utilizando formatos estándar de la web semántica y consultas SPARQL para la explotación de los datos.

Para alcanzar esta meta principal, se han establecido los siguientes sub-objetivos específicos:

1. **Diseñar un grafo de conocimiento** capaz de representar y relacionar las entidades biológicas fundamentales para la caracterización de las interacciones entre el VIH-1, sus proteínas y los fármacos antirretrovirales.
2. **Reutilizar identificadores (URIs) estándar** de repositorios y ontologías públicas de referencia en el ámbito biomédico (como UniProt, DrugBank y Gene Ontology), con el fin de maximizar la interoperabilidad y la reutilización de los datos generados, en línea con los principios FAIR.
3. **Diseñar y ejecutar un conjunto de cinco consultas SPARQL complejas** que demuestren la capacidad del grafo para responder a preguntas biológicas relevantes, lanzadas y documentadas a través de un script de R.

3. Diseño del Grafo y Fuentes de Datos

El diseño conceptual del grafo de conocimiento se basa en la información recopilada en la revisión sobre el VIH-1, sus componentes y su farmacología (De Clercq & Li, 2016; Engelman & Cherepanov, 2012). El objetivo del modelo es representar de forma clara y explícita las entidades biológicas clave y, sobre todo, las complejas interacciones que existen entre ellas, desde la replicación viral hasta la acción de los fármacos.

3.1. Clases e Instancias Definidas

Para estructurar el conocimiento, se ha definido una jerarquía de clases que permite categorizar cada entidad de forma precisa. Las clases principales son:

- **Virus:** Representa al agente viral. Su única instancia en este proyecto es `:VIH_1`, identificado con su TaxID de NCBI para asegurar la correcta identificación taxonómica.
- **Organismo:** Modela al huésped de la infección, cuya instancia es `:Homo_sapiens`.
- **Proteína:** Es una superclase general que se especializa en dos ramas para distinguir el origen de las proteínas:
 - *Proteína Viral:* Contiene las proteínas codificadas por el virus, como `:Transcriptasa_Inversa`. Una subclase importante es *Poliproteína Viral*, para diferenciar a los precursores.

- *ProteínaHumana*: Incluye las proteínas del huésped relevantes para el ciclo viral, como el receptor :CD4_Humano.
- **Farmaco**: Superclase que agrupa a los fármacos antirretrovirales. Se ha creado una jerarquía de subclases basada en el mecanismo de acción (:PI, :NRTI, etc.).

La jerarquía de clases resultante proporciona una estructura lógica y semánticamente rica, como se muestra en la Figura 2.

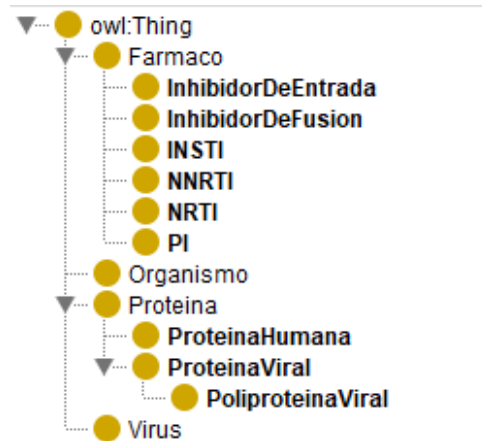


Figura 2: Jerarquía de las principales clases definidas en Protégé para la ontología del VIH-1. Se destaca la especialización de las clases *Proteína* y *Farmaco*.

3.2. Predicados (Object Properties) y Relaciones Modeladas

Para conectar las instancias de las clases anteriores, se han definido un conjunto de propiedades de objeto (predicados). Se ha priorizado la claridad semántica y la definición de relaciones inversas (*owl:inverseOf*) para facilitar la navegación y las consultas desde cualquier dirección del grafo. Las relaciones principales son:

- **:tieneDianaViral** y **:tieneDianaHumana**: Vinculan un fármaco con su proteína diana. Se crearon como sub-propiedades de una propiedad general **:tieneDiana**.
- **:interactuaCon**: Describe la interacción física entre proteínas. Se definió como una *owl:SymmetricProperty* para permitir la inferencia automática de la relación en ambas direcciones.
- **:derivaDe**: Modela la relación entre una proteína madura y su poliproteína precursora.
- **:esCodificadaPor**: Relaciona una proteína viral con el virus que contiene su información genética.

La jerarquía de propiedades implementada se visualiza en la Figura 3.

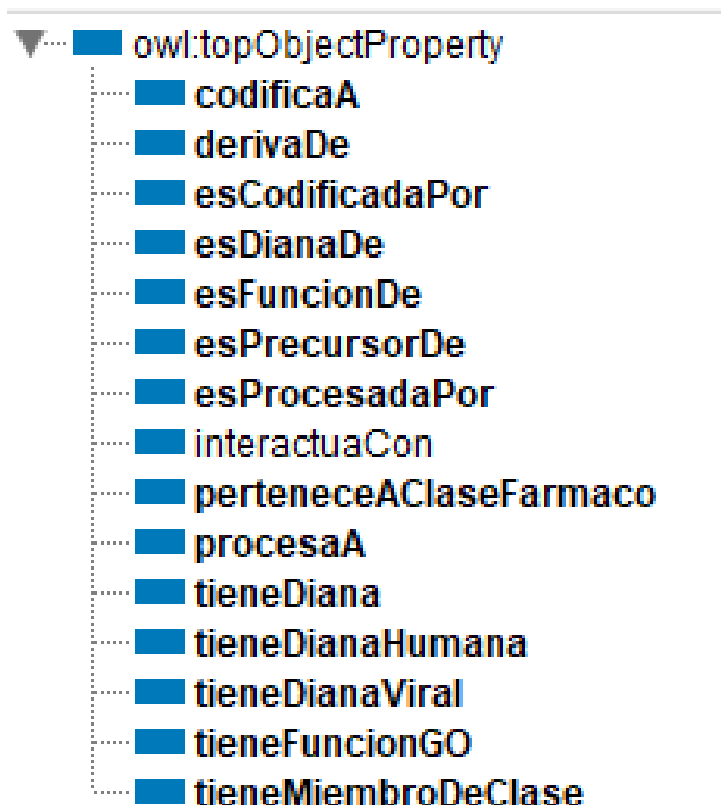


Figura 3: Jerarquía de las propiedades de objeto definidas en Protégé.

Para la construcción del grafo se han utilizado fuentes de datos externas y estándar en el campo. Los identificadores de las proteínas se han obtenido de UniProt, los de los fármacos de DrugBank, y los taxonómicos de NCBI Taxonomy, asegurando la máxima interoperabilidad posible.

4. Construcción del Grafo de Conocimiento con Protégé

Una vez definido el diseño conceptual, se procedió a la implementación del grafo utilizando el editor de ontologías Protégé (versión 5.5.0). El objetivo de esta fase fue traducir el modelo teórico a una estructura formal y computable en el lenguaje OWL.

4.1. Metodología de Construcción

El proceso de construcción se llevó a cabo de manera sistemática para asegurar la coherencia y riqueza del grafo:

1. **Creación de la Jerarquía de Clases:** Se implementó la estructura de clases y subclases (`owl:Class`, `rdfs:subClassOf`) descrita en la sección anterior, creando una espina dorsal taxonómica para categorizar todas las entidades del dominio.
2. **Definición de Propiedades:** Se definieron las propiedades de objeto (`owl:ObjectProperty`) para representar las relaciones biológicas. Para enriquecer la semántica del grafo, se hizo un uso sistemático de características avanzadas de OWL, como el establecimiento de propiedades inversas (`owl:inverseOf`) y la definición de propiedades

simétricas (`owl:SymmetricProperty`), para facilitar la navegación y permitir la inferencia automática de relaciones.

3. **Población con Individuos:** Se crearon las instancias (`owl:NamedIndividual`) para cada clase, representando las entidades concretas del estudio (ej. el fármaco `:Ritonavir`, la proteína `:Proteasa_VIH1`, etc.).
4. **Enlazado a Datos Externos (Linked Data):** Siguiendo los principios FAIR, cada individuo local fue sistemáticamente enlazado a su correspondiente entidad en bases de datos públicas de referencia (UniProt, DrugBank, NCBI Taxonomy). Esto se realizó mediante la aserción `owl:sameAs`, conectando nuestro grafo con el ecosistema global de datos enlazados.
5. **Anotación con Etiquetas Legibles:** Finalmente, se añadió una etiqueta legible (`rdfs:label`) a cada individuo para mejorar la usabilidad y la claridad de los resultados en las fases posteriores de consulta y visualización.

4.2. Métricas Finales de la Ontología

Para cuantificar la complejidad y el tamaño del grafo final, se presentan las métricas obtenidas directamente desde la pestaña *Ontology metrics* de Protégé (Figura 4).

Metrics	
Axiom	140
Logical axiom count	76
Declaration axioms count	64
Class count	13
Object property count	15
Data property count	0
Individual count	36
Annotation Property count	0

Figura 4: Métricas de la ontología del VIH-1 generadas por Protégé.

El grafo resultante consta de **140 axiomas** en total, con **13 clases**, **15 propiedades de objeto** y **36 individuos**. Este volumen de datos y relaciones supera ampliamente el mínimo de 20 tripletas requerido por la práctica, demostrando la riqueza semántica del modelo construido.

5. Despliegue y Publicación de Datos (Arquitectura FAIR)

La creación de un grafo de conocimiento coherente es el primer paso, pero para que los datos sean de utilidad para la comunidad, deben ser publicados de acuerdo con los principios FAIR (Findable, Accessible, Interoperable, and Reusable) [?]. Esta sección detalla la arquitectura técnica implementada en el servidor `dayhoff.inf.um.es` para asegurar una publicación de datos accesible y estándar.

5.1. Preparación del Grafo para Publicación

Para asegurar la máxima compatibilidad con las herramientas de publicación, se estandarizó la URI base de la ontología en el fichero fuente (`VIH_DGP_ontology.ttl`), estableciendo que terminase en almohadilla (`#`) como separador estándar.

A diferencia de otros flujos de trabajo que contextualizan los datos en grafos nombrados mediante N-Quads, para este despliegue se optó por un método más directo y robusto que alinea la configuración con los ejemplos vistos en la asignatura. Se utilizó el fichero `VIH_DGP_ontology.ttl` final para cargarlo directamente en el grafo por defecto del *triple store*, simplificando la configuración de la comunicación entre servicios.

5.2. Despliegue en Servidor dayhoff

El despliegue de los servicios se realizó íntegramente en el servidor `dayhoff` utilizando la tecnología de contenedores Docker, orquestada mediante un fichero `docker-compose.yml` proporcionado en el repositorio de la asignatura. Esta arquitectura permite lanzar de forma coordinada los dos componentes principales:

- **Blazegraph:** Actúa como el *triple store* o base de datos RDF, almacenando el grafo y exponiendo un *endpoint* para realizar consultas SPARQL.
- **Trifid:** Funciona como el *frontend* o servidor de Linked Data, proporcionando una interfaz web navegable para explorar los recursos del grafo.

La configuración del fichero `docker-compose.yml` se personalizó para este proyecto, realizando los siguientes ajustes clave:

- **Maapeo de Puertos:** Se asignaron los puertos personales correspondientes: el puerto **3038** para Blazegraph y el puerto **8178** para Trifid.
- **Variables de Entorno de Trifid:** Se configuró la variable `SPARQL_ENDPOINT_URL` para apuntar al *namespace* por defecto de Blazegraph, **kb**, y la variable `DATASET_BASE_URL` para que coincidiera con la URI base de la ontología.


Una vez configurado, los servicios se levantaron con el comando `docker compose -p daniel_virology up -d`. Finalmente, se procedió a limpiar el *namespace kb* y a cargar en él el fichero `VIH_DGP_ontology.ttl` definitivo.

5.3. Endpoints y Recursos Disponibles

Para facilitar la corrección, los servicios se han dejado lanzados y están disponibles de forma permanente en las siguientes direcciones:

- Interfaz Web de Linked Data (Trifid): <http://dayhoff.inf.um.es:8178/>
- Endpoint SPARQL (Blazegraph): <http://dayhoff.inf.um.es:3038/>

La Figura 5 muestra la página generada por Trifid para el individuo :VIH_1, demostrando la correcta publicación de los datos y su navegabilidad.



Human immunodeficiency virus type 1

http://dayhoff.inf.um.es:8178/VIH_1

Human immunodeficiency virus type 1	
rdf:type	owl#NamedIndividual Virus
rdfs:label	Human immunodeficiency virus type 1 <small>not string</small>
owl:sameAs	NCBITaxon_11676
Gag_Pol_Precursor	
esCodificadaPor	Human immunodeficiency virus type 1
gp160_Precursor	
esCodificadaPor	Human immunodeficiency virus type 1

Number of results per named graph

Graph name	Number of results
Default graph	6

Figura 5: Interfaz de Linked Data generada por Trifid para el recurso :VIH_1 en http://dayhoff.inf.um.es:8178/VIH_1.

Nota para la corrección: Durante las pruebas, se detectó que algunos navegadores con configuraciones de seguridad estrictas intentaban forzar una conexión HTTPS, lo que puede dar un error de conexión (SSL_ERROR...). Si esto ocurre, se recomienda acceder a las URLs desde una ventana de incógnito o asegurarse de que la dirección comienza con http://.

6. Explotación de Datos con Consultas SPARQL

Una vez construido el grafo de conocimiento y almacenado en el *triple store* Blaze-graph, se procedió a su explotación mediante el lenguaje de consulta SPARQL. El objetivo de esta fase es validar el modelo y demostrar su capacidad para responder a preguntas biológicas complejas, cumpliendo con los requisitos de la práctica.

Las siguientes cinco consultas se diseñaron con una complejidad creciente y se ejecutaron a través de un script de R. Para mayor claridad en este informe, se presentan los resultados obtenidos directamente desde la interfaz de Blazegraph.

6.1. Consulta 1: Trazabilidad Completa desde el Fármaco hasta el Virus

Pregunta: Para cada fármaco con diana viral, ¿cuál es su diana, de qué poliproteína precursora deriva dicha diana y qué virus la codifica?

Técnica Demostrada: Esta consulta demuestra la capacidad de SPARQL para realizar una navegación profunda en el grafo a través de joins implícitos. Se construye una cadena de cuatro patrones de tripletas que debe satisfacerse en su totalidad, mostrando cómo se puede trazar una relación desde un extremo del grafo (un fármaco) hasta otro muy distante (el virus) de forma eficiente.

```
1 PREFIX : <http://dayhoff.inf.um.es/ontologies/VIH_DGP#>
2 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3
4 SELECT ?farmaco_label ?diana_label ?precursor_label ?virus_label
5 WHERE {
6   ?farmaco :tieneDianaViral ?diana .
7   ?diana :derivaDe ?precursor .
8   ?precursor :esCodificadaPor ?virus .
9
10  ?farmaco rdfs:label ?farmaco_label .
11  ?diana rdfs:label ?diana_label .
12  ?precursor rdfs:label ?precursor_label .
13  ?virus rdfs:label ?virus_label .
14 }
15 ORDER BY ?farmaco_label
```

[Advanced features](#)

farmaco_label	diana_label	precursor_label	virus_label
Efavirenz	Transcriptasa Inversa (RT)	Poliproteína Gag-Pol	Human immunodeficiency virus type 1
Enfuvirtida	Glicoproteína de envoltura gp41 (TM)	Glicoproteína de envoltura gp160	Human immunodeficiency virus type 1
Raltegravir	Integrasa (IN)	Poliproteína Gag-Pol	Human immunodeficiency virus type 1
Ritonavir	Proteasa (PR)	Poliproteína Gag-Pol	Human immunodeficiency virus type 1
Zidovudina (AZT)	Transcriptasa Inversa (RT)	Poliproteína Gag-Pol	Human immunodeficiency virus type 1

Figura 6: Resultado de la consulta de trazabilidad. Se muestran las relaciones en cadena desde cada fármaco hasta el virus VIH-1.

6.2. Consulta 2: Dianas Derivadas de Poliproteínas y sus Fármacos

Pregunta: ¿Qué proteínas que derivan de una poliproteína viral son, además, diana de algún fármaco? Mostrar su nombre, el fármaco asociado, el precursor y su identificador externo en UniProt.

Técnica Demostrada: La complejidad de esta consulta radica en la convergencia de patrones sobre un mismo recurso (?proteina). Demuestra cómo filtrar un conjunto de entidades basándose en que deben satisfacer varias relaciones distintas simultáneamente, y cómo recuperar enlaces externos (owl:sameAs) para su posterior análisis.

```
1 PREFIX : <http://dayhoff.inf.um.es/ontologies/VIH_DGP#>
2 PREFIX owl: <http://www.w3.org/2002/07/owl#>
3 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
4 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
5
6 SELECT ?proteina_label ?farmaco_label ?precursor_label ?enlace_uniprot
7 WHERE {
8   # Buscamos una proteína que sea diana de un fármaco
9   ?farmaco :tieneDianaViral ?proteina .
10  # Y que esa misma proteína derive de un precursor
11  ?proteina :derivaDe ?precursor .
12  # Nos aseguramos que el precursor es una PoliproteínaViral
13  ?precursor rdf:type :PoliproteínaViral .
14
15  # Recuperamos el enlace a UniProt usando owl:sameAs
16  ?proteina owl:sameAs ?enlace_uniprot .
17
18  # Obtenemos las etiquetas para que el resultado sea legible
19  ?proteina rdfs:label ?proteina_label .
20  ?farmaco rdfs:label ?farmaco_label .
21  ?precursor rdfs:label ?precursor_label .
22 }
23 ORDER BY ?proteina_label
```

[Advanced features](#)

proteina_label	farmaco_label	precursor_label	enlace_uniprot
Glicoproteína de envoltura gp41 (TM)	Enfuvirtida	Glicoproteína de envoltura gp160	http://purl.uniprot.org/uniprot/Q53119
Integrasa (IN)	Raltegravir	Poliproteína Gag-Pol	http://purl.uniprot.org/uniprot/T1QYY3
Proteasa (PR)	Ritonavir	Poliproteína Gag-Pol	http://purl.uniprot.org/uniprot/Q6QK01
Transcriptasa Inversa (RT)	Efavirenz	Poliproteína Gag-Pol	http://purl.uniprot.org/uniprot/Q76441
Transcriptasa Inversa (RT)	Zidovudina (AZT)	Poliproteína Gag-Pol	http://purl.uniprot.org/uniprot/Q76441

Figura 7: Resultado de la consulta de dianas específicas. Se listan las proteínas que cumplen el doble rol, el fármaco que las ataca y su enlace a UniProt.

6.3. Consulta 3: Fármacos que NO son Inhibidores de Proteasa

Pregunta: ¿Qué fármacos de nuestro catálogo NO pertenecen a la clase Inhibidor de Proteasa (PI)?

Técnica Demostrada: Esta consulta combina la inferencia sobre jerarquías de clases (`rdfs:subClassOf`) con el filtrado lógico (`FILTER`). Primero, se identifican de forma genérica todos los individuos que son fármacos y, posteriormente, se excluyen explícitamente aquellos que pertenecen a la clase `:PI`, demostrando una técnica de negación robusta.

```
1 PREFIX : <http://dayhoff.inf.um.es/ontologies/VIH_DGP#>
2 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
4
5 SELECT ?farmaco_label ?clase_uri
6 WHERE {
7     # Buscamos cualquier individuo que pertenezca a una subclase de :Farmaco
8     ?farmaco rdf:type ?clase_uri .
9     ?clase_uri rdfs:subClassOf :Farmaco .
10
11     # Filtramos para quedarnos solo con aquellos cuya clase NO sea :PI
12     FILTER (?clase_uri != :PI)
13
14     # La etiqueta del fármaco es opcional por si alguno no la tuviera
15     OPTIONAL { ?farmaco rdfs:label ?farmaco_label . }
16 }
17 ORDER BY ?clase_uri
```

[Advanced features](#)

<	
farmaco_label	clase_uri
Raltegravir	http://dayhoff.inf.um.es/ontologies/VIH_DGP#INSTI
Maraviroc	http://dayhoff.inf.um.es/ontologies/VIH_DGP#InhibidorDeEntrada
Enfuvirtida	http://dayhoff.inf.um.es/ontologies/VIH_DGP#InhibidorDeFusion
Efavirenz	http://dayhoff.inf.um.es/ontologies/VIH_DGP#NNRTI
Zidovudina (AZT)	http://dayhoff.inf.um.es/ontologies/VIH_DGP#NRTI

Figura 8: Resultado de la consulta de exclusión. Se muestran todos los fármacos del grafo excepto aquellos clasificados como Inhibidores de Proteasa.

6.4. Consulta 4: Dianas de Fármacos Específicos usando VALUES

Pregunta: Para un conjunto específico de fármacos de interés (Ritonavir y Maraviroc), ¿cuáles son sus dianas y de qué tipo (Viral o Humana)?

Técnica Demostrada: Esta consulta introduce la cláusula `VALUES`, una forma muy potente y eficiente de realizar consultas dirigidas a un subconjunto específico de datos. Se combina con `UNION` para buscar en dos propiedades de relación distintas y con `BIND` para crear una columna descriptiva que enriquece el resultado.

```

1 PREFIX : <http://dayhoff.int.um.es/ontologies/VIH_DGP#>
2 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3
4 SELECT ?farmaco_label ?tipo_diana ?diana_label
5 WHERE {
6   # Con VALUES especificamos la lista de fármacos que nos interesan
7   VALUES ?farmaco { :Ritonavir :Maraviroc }
8
9   # Usamos UNION para buscar en las dos posibles propiedades de diana
10  {
11    ?farmaco :tieneDianaViral ?diana .
12    BIND('Viral' AS ?tipo_diana)
13  }
14  UNION
15  {
16    ?farmaco :tieneDianaHumana ?diana .
17    BIND('Humana' AS ?tipo_diana)
18  }
19
20  # Obtenemos las etiquetas para que el resultado sea legible
21  ?farmaco rdfs:label ?farmaco_label .
22  ?diana rdfs:label ?diana_label .
23 }

```

[Advanced features](#)

farmaco_label	tipo_diana	diana_label
Maraviroc	Humana	C-C chemokine receptor type 5 (CCR5)
Ritonavir	Viral	Proteasa (PR)

Figura 9: Resultado de la consulta con VALUES. Se muestran únicamente las dianas para los dos fármacos especificados.

6.5. Consulta 5: Resumen de Fármacos por Diana Viral (Agregación)

Pregunta: ¿Cuántos fármacos diferentes atacan a cada proteína viral de nuestro grafo y cuáles son sus nombres?

Técnica Demostrada: Es una consulta de agregación y resumen de datos. Utiliza GROUP BY para agrupar los resultados por diana viral, la función COUNT para contar el número de fármacos en cada grupo, y GROUP_CONCAT para crear una lista legible de los fármacos asociados. Demuestra la capacidad del grafo no solo para recuperar datos, sino para analizarlos y sintetizarlos.

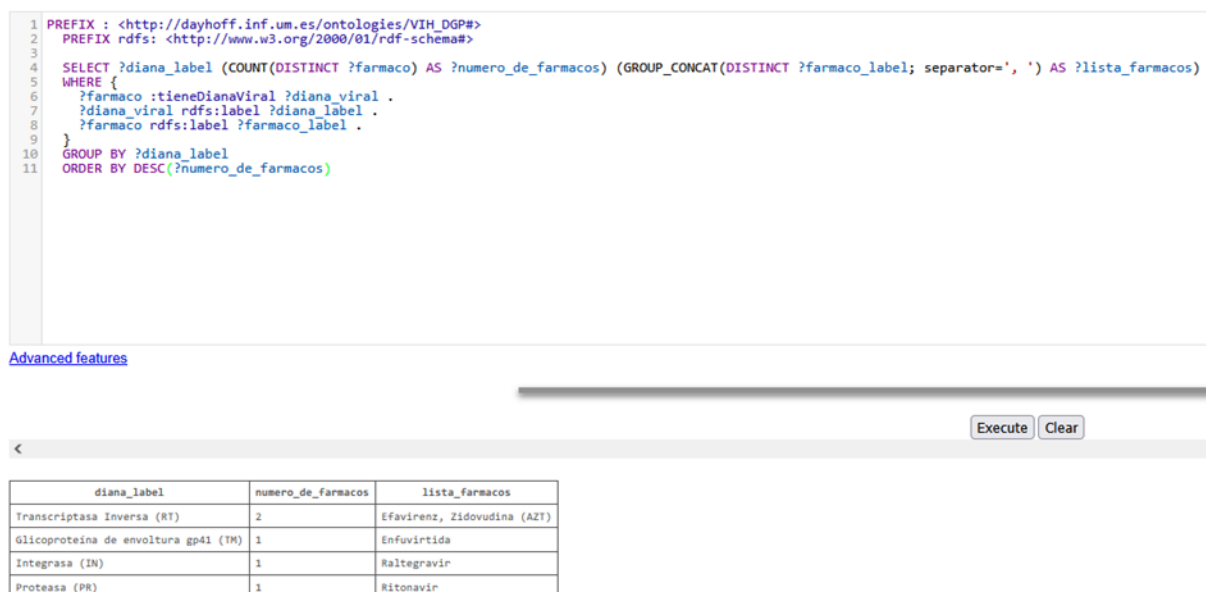


Figura 10: Resultado de la consulta de agregación, mostrando un resumen cuantitativo de fármacos por cada diana viral.

7. Conclusiones

En este trabajo se ha recreado con éxito el proceso completo de creación, publicación y explotación de datos semánticos en un dominio biomédico complejo, cumpliendo todos los objetivos propuestos en la práctica.

Se ha logrado diseñar y construir un grafo de conocimiento coherente y semánticamente rico sobre el virus VIH-1 y su farmacología, superando el mínimo de tripletas requerido. La adhesión a los principios FAIR se ha garantizado mediante la reutilización de vocabularios estándar (RDF, RDFS, OWL) y el enlazado de los individuos de la ontología a bases de datos públicas de referencia como UniProt y DrugBank.

La arquitectura de publicación, desplegada en el servidor **dayhoff** mediante Docker, ha resultado en un *endpoint* SPARQL funcional a través de Blazegraph y una interfaz de datos enlazados navegable con Trifid. Finalmente, se ha demostrado la capacidad de explotar el grafo para responder a preguntas complejas mediante un script de R, validando así la utilidad del modelo de datos creado.

Aunque el *dataset* actual es una prueba de concepto robusta, la ontología se ha diseñado con la extensibilidad en mente. Como trabajo futuro, el grafo podría ampliarse para incluir datos sobre mutaciones específicas en las proteínas virales y su relación con la resistencia a los fármacos, información más detallada sobre el metabolismo de los mismos en el huésped o la inclusión de más factores celulares que interactúan con el ciclo viral.

Este proyecto pone de manifiesto el enorme potencial de las tecnologías de la web semántica para integrar y analizar la información heterogénea del ámbito biomédico, sentando las bases para herramientas más potentes en la investigación y la medicina personalizada.

8. Ubicación de los Ficheros del Proyecto

Tal y como se solicita en el enunciado de la práctica, todos los ficheros generados para la resolución de este trabajo (ficheros `.ttl`, script de R, etc.) han sido depositados en el servidor `dayhoff` para su corrección. Los servicios de Blazegraph y Trifid se han dejado en ejecución.

- **Servidor Dayhoff:** `/home/alumno09/Explotacion_Semantica_Datos/Entrega_ESD/`
- **Repositorio GitHub:** <https://github.com/DanielGP121/Entrega-ESD>

Bibliografía

Referencias

- [1] De Clercq, E., & Li, G. (2016). Approved Antiviral Drugs over the Past 50 Years. *Clinical Microbiology Reviews*, 29(3), 695–747. <https://doi.org/10.1128/CMR.00102-15>
- [2] Engelman, A., & Cherepanov, P. (2012). The structural biology of HIV-1: Mechanistic and therapeutic insights. *Nature Reviews Microbiology*, 10(4), 279–290. <https://doi.org/10.1038/nrmicro2747>