Universidad Politécnica
de Madrid

Escuela Técnica Superior de
Ingenieros Informáticos

Grado en Ciencia de Datos e Inteligencia Artificial

Trabajo Fin de Grado

Aplicación de GNN sobre Grafos de Conocimiento de Cáncer de Pulmón

Autor: David Hernando González

Tutor: Antonio Jesús Díaz Honrubia

Co-tutora: Delia Aminta Moreno Perdomo

Madrid, junio, 2025

Este Trabajo Fin de Grado se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

Trabajo Fin de Grado
Grado en Ciencia de Datos e Inteligencia Artificial
Título: Aplicación de GNN sobre Grafos de Conocimiento de Cáncer de Pulmón
Junio, 2025

Autor: David Hernando González

Tutor: Antonio Jesús Díaz Honrubia
Departamento de Lenguajes y Sistemas Informáticos e Ingeniería de Software
ETSI Informáticos
Universidad Politécnica de Madrid

Resumen

En los últimos años, la investigación sobre el cáncer de pulmón ha generado enormes volúmenes de datos biomédicos dispersos en múltiples repositorios, lo que dificulta la identificación de factores de riesgo y relaciones biológicas relevantes. Los grafos de conocimiento ofrecen un marco para integrar información heterogénea en una misma estructura, mientras que las Graph Neural Networks (GNN) han demostrado un gran potencial para extraer conocimiento de grafos complejos, al aprender representaciones que conservan la topología y las interacciones entre entidades. Este Trabajo de Fin de Grado aprovecha dicha sinergia para aplicar GNN a un grafo de conocimiento biomédico previamente construido, con el objetivo de descubrir asociaciones inéditas y posibles factores de riesgo asociados al cáncer de pulmón.

Partiendo de un subgrafo enfocado en once clases clave del dominio, se ha procedido a extraer las tripletas relevantes mediante consultas SPARQL. Tras limpiar identificadores, generar las relaciones inversas y transformar las listas de adyacencia en tensores. A continuación, se ha desarrollado un modelo híbrido que combina reglas simbólicas extraídas con AnyBURL y una red neuronal de grafos basada en GCNConv. Dicha integración permite que las predicciones se sustenten tanto en patrones explícitos (reglas lógicas) como en las representaciones latentes aprendidas por la GNN.

El modelo resultante ha sido entrenado mediante muestreo negativo y validado con métricas estándar de predicción de enlaces (Mean Reciprocal Rank y Hits@K). Los resultados muestran un rendimiento competitivo. Además, el análisis cualitativo de las predicciones han resaltado asociaciones biológicas interesantes que coinciden con la literatura biomédica y sugieren nuevas hipótesis que podrían guiar investigaciones clínicas futuras.

Abstract

In recent years, lung cancer research has generated huge volumes of biomedical data scattered across multiple repositories, making it difficult to identify relevant risk factors and biological relationships. Knowledge graphs offer a framework for integrating heterogeneous information into a single structure, while Graph Neural Networks (GNNs) have shown great potential for extracting knowledge from complex graphs by learning topology-preserving representations and interactions between entities. This thesis takes advantage of this synergy to apply GNNs to a previously constructed biomedical knowledge graph, with the aim of discovering unpublished associations and possible risk factors associated with lung cancer.

Starting from a subgraph focused on eleven key classes of the domain, we proceeded to extract the relevant triples by means of SPARQL queries. After cleaning identifiers, generating the inverse relations and transforming the adjacency lists into tensors. Then, a hybrid model has been developed combining symbolic rules extracted with AnyBURL and a graph neural network based on GCNConv. This integration allows predictions to be based on both explicit patterns (logical rules) and latent representations learned by the GNN.

The resulting model has been trained by negative sampling and validated with standard link prediction metrics (Mean Reciprocal Rank and Hits@K). The results show a competitive performance. In addition, qualitative analysis of the predictions have highlighted interesting biological associations that are consistent with the biomedical literature and suggest new hypotheses that could guide future clinical research.

Tabla de contenidos

Índice de figuras

1 Introducción

En la actualidad, el cáncer de pulmón constituye uno de los desafíos más críticos de la salud pública a nivel mundial. Se trata de la principal causa de muerte por cáncer en el mundo, con aproximadamente 1,8 millones de fallecimientos anuales (18% de todas las muertes por cáncer) según datos recientes de la Organización Mundial de la Salud [1]. Este alto índice de mortalidad se debe en parte a que muchos casos se diagnostican en etapas avanzadas, debido a que en fases iniciales carece de síntomas evidentes, cuando las opciones terapéuticas son limitadas y la tasa de supervivencia disminuye drásticamente [1].

A pesar de los avances en tratamientos, como terapias dirigidas a mutaciones específicas (EGFR, ALK, entre otras) e inmunoterapias contra puntos de control inmunitario, el cáncer de pulmón presenta una de las tasas de supervivencia a cinco años más bajas entre todos los tipos de cáncer, en el orden del 10–20% [2]. Esta realidad evidencia la necesidad de nuevos enfoques que apoyen el diagnóstico temprano, la selección óptima de tratamientos y el descubrimiento de terapias innovadoras para mejorar la esperanza y calidad de vida de los pacientes. En este contexto, se han comenzado a emplear grafos de conocimiento biomédico, que representan nodos (entidades) y aristas (relaciones) para integrar información heterogénea (genómica, clínica, farmacológica, etc.) y capturar tanto los datos como sus conexiones explícitas e implícitas, facilitando así la generación de hipótesis, como se puede observar por ejemplo en la Figura 1 [4]. Este enfoque es especialmente valioso en biomedicina, donde muchas veces "el cáncer no es la consecuencia de una anomalía en un solo gen, sino el reflejo de una interacción compleja de muchos genes y factores moleculares" [3].

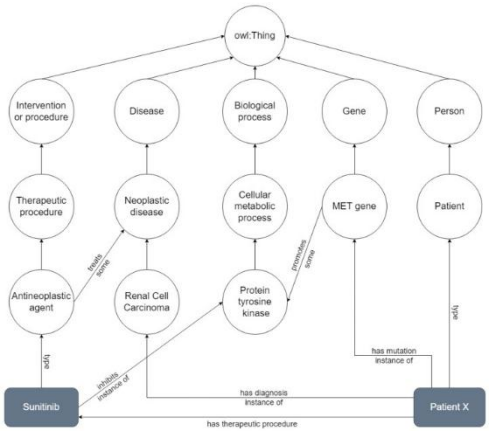


Figura 1: Ejemplo de grafo de conocimiento biomédico.

En paralelo al auge de los grafos de conocimiento, han surgido algoritmos de aprendizaje automático capaces de explotar estas estructuras de datos

complejas. En particular, las redes neuronales de grafos (Graph Neural Networks, GNN) están demostrando un gran potencial para minar conocimiento de grafos biomédicos. Las GNN son una familia de modelos de aprendizaje profundo diseñados específicamente para aprender representaciones de nodos y relaciones en grafos, preservando la topología de las conexiones. Estudios recientes destacan que las GNN pueden manejar datos biomoleculares estructurados (redes de interacción proteína-proteína, interacciones genéticas, redes de coexpresión, etc.) mejor que los métodos tradicionales, descubriendo patrones y módulos de genes asociados a enfermedades que antes no se detectaban [3].

1.1 Motivación y necesidad del proyecto

A la luz de lo expuesto, la motivación de este proyecto surge de unir estos dos mundos (biomedicina y aprendizaje automático) para abordar un problema de gran relevancia social y científica: el cáncer de pulmón. Por un lado, la necesidad es evidente dado el impacto de esta enfermedad y las limitaciones de las aproximaciones actuales. Mejorar la detección de blancos terapéuticos, identificar biomarcadores para personalizar tratamientos o descubrir medicamentos eficaces contra subtipos resistentes, son objetivos urgentes en la lucha contra el cáncer de pulmón. Lograr estos objetivos requiere aprovechar la ingente cantidad de conocimiento biomédico disponible, integrándolo de forma inteligente para extraer información accionable. Por otro lado, las oportunidades tecnológicas nunca han sido tan favorables: se dispone de bases de datos y ontologías ricas en conocimiento oncológico, así como de algoritmos avanzados capaces de manejar dicha complejidad. En consecuencia, este proyecto se plantea para investigar y demostrar cómo una aplicación de GNN sobre un grafo de conocimiento especializado en cáncer de pulmón puede generar nuevo conocimiento útil, apoyando la toma de decisiones en investigación biomédica.

1.2 Objetivos

El objetivo general de este proyecto es desarrollar un modelo basado en Graph Neural Networks capaz de analizar un grafo de conocimiento relativo al cáncer de pulmón para extraer información relevante sobre posibles factores de riesgo. A partir de este objetivo general, se definen los siguientes objetivos específicos (OE) que se abordan a lo largo del trabajo:

- **OE1**: estudiar las tecnologías necesarias para el desarrollo del modelo, incluyendo los fundamentos de Graph Neural Networks, el concepto de grafos de conocimiento y la terminología biomédica relevante sobre cáncer de pulmón.

- **OE2**: familiarizarse con los datos del grafo, comprendiendo su estructura, las tipologías de nodos (entidades biomédicas) y las distintas relaciones

que los enlazan, con el fin de entender plenamente la información disponible.

- **OE3**: transformar el grafo de conocimiento en un formato adecuado para su integración con el modelo de GNN, realizando las conversiones o preprocesamientos necesarios.

- **OE4**: entrenar y optimizar la red neuronal sobre el grafo, ajustando los hiperparámetros y mejorando el rendimiento del modelo en la tarea de descubrimiento de conocimiento.

- **OE5**: evaluar los resultados obtenidos con el modelo y validar la utilidad del mismo para identificar factores de riesgo, comparando las inferencias o hallazgos de la GNN con conocimiento biomédico establecido o con expectativas basadas en la literatura.

1.3 Alcance del proyecto

El alcance de este proyecto trasciende la mera implementación técnica de un modelo; busca generar un impacto significativo en la investigación biomédica y, potencialmente, en la salud pública. Al aplicar técnicas de inteligencia artificial para descubrir patrones en un grafo de conocimiento de cáncer de pulmón, se espera facilitar la identificación de factores de riesgo clave y sus interrelaciones. Esta información podría ayudar a los investigadores médicos a generar nuevas hipótesis sobre la etiología y progresión del cáncer de pulmón, orientando estudios posteriores. En un contexto más amplio, herramientas como la desarrollada en este TFG podrían integrarse en sistemas de apoyo a la decisión clínica o en plataformas de vigilancia epidemiológica, contribuyendo a una detección más temprana de poblaciones de riesgo y a la formulación de estrategias preventivas más efectivas. Todo ello repercutiría positivamente en la sociedad, ya que un mejor entendimiento de los factores que predisponen al cáncer de pulmón es fundamental para diseñar campañas de salud pública y políticas sanitarias enfocadas en la prevención.

Asimismo, el proyecto guarda alineación con objetivos globales de salud. En particular, conecta con la meta 3.4 de los Objetivos de Desarrollo Sostenible (ODS) de Naciones Unidas, que busca reducir en un tercio la mortalidad prematura por enfermedades no transmisibles de aquí a 2030 mediante la prevención y el tratamiento [5]. Dado que el cáncer de pulmón es una de las principales causas de mortalidad prematura por cáncer, avanzar en su control y comprensión es imprescindible para acercarse a dicho objetivo [6]. En este sentido, iniciativas que combinen tecnología e investigación biomédica, como la presentada, aportan una vía innovadora para combatir enfermedades de alto impacto. Si bien los resultados de este TFG son de naturaleza exploratoria, representan un paso hacia nuevas herramientas basadas en conocimiento que podrían potenciar la investigación oncológica y, a largo plazo, mejorar la calidad de vida y pronóstico de los pacientes.

Finalmente, cabe destacar que la metodología desarrollada, integración de grafos de conocimiento con GNN, es generalizable. Esto significa que, con las adaptaciones pertinentes, podría aplicarse a otros dominios biomédicos o a otras enfermedades complejas donde intervienen múltiples factores, amplificando así el alcance y la utilidad futura de la herramienta más allá del caso concreto de cáncer de pulmón.

1.4 Planificación

El presente proyecto se organiza en tres fases principales: investigación, desarrollo y análisis, que recogen las tareas planificadas y su duración. A continuación, se describen brevemente estas fases:

Fase de Investigación

1. Investigación y Planificación. Se realiza una búsqueda bibliográfica en profundidad para conocer tanto el estado actual de investigación relativo al cáncer de pulmón como los fundamentos de los grafos de conocimiento y las Graph Neural Networks (GNN). Asimismo, se definen los objetivos iniciales y la metodología a seguir, estimando los recursos y el cronograma del proyecto.

2. Familiarización con los Datos del Grafo. Se explora el conjunto de datos, identificando los tipos de nodos y las relaciones existentes. En este punto, se consolida la comprensión de la ontología empleada y se establecen los requisitos para su posterior uso en la GNN.

Fase de Desarrollo

3. Transformación del Grafo para su Integración con GNN. Se llevan a cabo los procesos de integración de datos necesarios para adaptar la estructura del grafo al modelo de redes neuronales de grafos. Se definen, entre otros aspectos, las representaciones internas de los nodos y las matrices de adyacencia, así como las funcionalidades requeridas para el posterior entrenamiento del modelo.

4. Entrenamiento y Optimización del Modelo. Una vez transformado el grafo, se diseña la arquitectura de la GNN. Seguidamente, se entrena la red neuronal ajustando hiperparámetros y se implementan técnicas de optimización para mejorar la precisión y eficiencia del descubrimiento de información relevante sobre el cáncer de pulmón.

Fase de Análisis

5. Evaluación y Validación de Resultados. Se evalúa el rendimiento del modelo entrenado mediante métricas adecuadas (MRR y Hits@k) y se comparan los hallazgos con estudios previos o con el conocimiento biomédico existente. Asimismo, se investiga la interpretabilidad de las predicciones para facilitar su comprensión por parte de investigadores médicos.

6. Escritura de la Memoria. Finalmente, se documentan detalladamente los resultados y las conclusiones en la memoria del proyecto. Se revisan, además, los aprendizajes obtenidos y las limitaciones encontradas, proponiendo posibles líneas futuras de investigación.

Esta estructura secuencial garantiza un flujo ordenado de trabajo, asegurando la coherencia entre la fase de estudio preliminar, la implementación práctica y la valoración crítica de los resultados. En la Figura 2 se muestra el diagrama de Gantt donde se reflejan las tareas y su duración estimada, sirviendo como hoja de ruta para la consecución de los objetivos descritos.

	SEMANA 1	SEMANA 2	SEMANA 3	SEMANA 4	SEMANA 5	SEMANA 6	SEMANA 7	SEMANA 8	SEMANA 9	SEMANA 10	SEMANA 11	SEMANA 12
TAREA 1												
TAREA 2												
TAREA 3												
TAREA 4												
TAREA 5												
TAREA 6												

Figura 2: Diagrama de Gantt

1.5 Estructura de la memoria

En esta sección, se explica brevemente la estructura que sigue esta memoria. El resto del documento se compone de seis capítulos, cada uno abordando un aspecto fundamental del proyecto:

Capítulo 2. Estado del arte

Se presenta un panorama de los antecedentes en la investigación del cáncer de pulmón, así como una introducción a los conceptos de grafos de conocimiento y Graph Neural Networks (GNN). Se revisan los trabajos más relevantes en este ámbito para situar la propuesta del proyecto en el contexto de la literatura existente.

Capítulo 3. Tecnologías empleadas

En este capítulo se enumeran y justifican, de forma general, las herramientas y tecnologías elegidas para desarrollar el proyecto, abarcando el entorno de programación, los mecanismos de acceso y consulta al grafo, la extracción de reglas simbólicas y el soporte para entrenamiento de modelos. También se mencionan brevemente las alternativas consideradas y los motivos por los que se optó por cada una de las tecnologías seleccionadas.

Capítulo 4. Implementación de la GNN

Se describe de manera global el flujo de trabajo seguido para construir el modelo de red neuronal sobre grafos: desde la obtención y preparación de los datos del grafo hasta el diseño del modelo híbrido con componentes simbólicos y su posterior entrenamiento y validación.

Capítulo 5. Evaluación

Se evalúan los resultados obtenidos tras la implementación del modelo. Se describe la metodología de evaluación y las métricas utilizadas, presentando los resultados cuantitativos y cualitativos más relevantes. Asimismo, se discuten las fortalezas y debilidades del enfoque propuesto, tanto desde la perspectiva técnica como desde la óptica biomédica.

Capítulo 6. Conclusiones y trabajo futuro

En este capítulo se sintetizan las conclusiones globales del proyecto, relacionándolas con los objetivos establecidos inicialmente. Se reflexiona sobre el grado de consecución de dichos objetivos, la aportación del modelo a la investigación del cáncer de pulmón y posibles mejoras que podrían aplicarse. Por último, se exponen las líneas futuras de investigación, ampliaciones y variaciones del modelo que podrían seguir explorándose.

Capítulo 7. Análisis de impacto

Finalmente, se estudia el impacto social, medioambiental y cultural de este trabajo, incidiendo en sus posibles aplicaciones en la investigación biomédica y la eventual repercusión en la prevención o tratamiento del cáncer de pulmón. Este análisis conecta los avances propuestos con los Objetivos de Desarrollo Sostenible (ODS) y el potencial beneficio que podrían aportar a la sociedad.

Tras estos capítulos, se presenta la **Bibliografía**, la cual recopila todas las referencias utilizadas a lo largo del documento siguiendo el estilo de citación IEEE.

2 Estado del arte

En este capítulo se presenta una revisión exhaustiva del estado actual en la investigación del cáncer de pulmón, incluyendo sus antecedentes clínicos, diagnóstico y avances terapéuticos recientes. Además, se describen las herramientas computacionales más prometedoras para avanzar en este campo, destacando los grafos de conocimiento y las redes neuronales de grafos (GNN).

2.1 Antecedentes en la investigación del cáncer de pulmón

El cáncer de pulmón es uno de los tumores más estudiados en oncología dada su elevada incidencia y mortalidad. A nivel global, se ha consolidado como la principal causa de muerte por cáncer, tanto en hombres como en mujeres [1]. Una característica preocupante es que el diagnóstico suele ocurrir en etapas avanzadas de la enfermedad, cuando las opciones terapéuticas son más limitadas y el pronóstico es pobre [1]. De hecho, la tasa de supervivencia a 5 años es de las más bajas; aunque ha mejorado en años recientes, en países como Estados Unidos todavía ronda solo el 24% [7].

En cuanto a tipos histológicos, el cáncer de pulmón se clasifica principalmente en carcinoma de pulmón microcítico (de células pequeñas, SCLC) y no microcítico (NSCLC). Este último representa alrededor del 85% de los casos e incluye subtipos como adenocarcinoma y carcinoma escamoso [1]. Los factores de riesgo más importantes son el tabaquismo (presente en el 85% de los casos) y la exposición a carcinógenos ambientales [1]. Es por ello que las estrategias de prevención (control del tabaquismo, reducción de exposición a humo secundario y contaminantes) son esenciales para disminuir la incidencia.

En la última década se han logrado avances significativos en el diagnóstico precoz y el tratamiento del cáncer de pulmón. En el plano diagnóstico, la introducción de programas de cribado (screening) con tomografía computarizada de baja dosis en poblaciones de alto riesgo ha permitido detectar más tumores en etapas iniciales, con potencial de mejorar sustancialmente las tasas de curación [1]. Sin embargo, la implementación del cribado no es uniforme; informes recientes señalan que la falta de acceso amplio a programas de detección temprana está frenando mayores progresos en supervivencia [7]. Otro factor clave es la identificación de biomarcadores moleculares en los tumores. Hoy en día, el análisis genómico de la biopsia permite buscar mutaciones accionables y marcadores como PD-L1, lo cual guía la selección de terapias dirigidas e inmunoterapias. Lamentablemente, existen brechas en la realización de estas pruebas de biomarcadores de forma universal, lo que impide que todos los pacientes se beneficien de un tratamiento personalizado óptimo [7].

En el tratamiento, la revolución más notable ha venido de la mano de las terapias sistémicas de nueva generación. Tradicionalmente, la cirugía y la radioterapia son curativas en enfermedad localizada, mientras que la

quimioterapia ha sido la piedra angular para enfermedad avanzada. Actualmente, la inmunoterapia y los fármacos dirigidos han transformado el panorama terapéutico del cáncer de pulmón metastásico. En concreto, la inmunoterapia basada en inhibidores de puntos de control inmunitario (anti-PD-1, anti-PD-L1, CTLA-4) ha revolucionado el tratamiento del cáncer de pulmón avanzado [1], logrando respuestas duraderas en un subconjunto de pacientes que previamente tenían muy pocas opciones. Paralelamente, los tratamientos dirigidos a alteraciones genéticas específicas han mostrado tasas de respuesta elevadas en pacientes seleccionados por biomarcadores, bloqueando las vías de señalización que impulsan el crecimiento tumoral [1]. Gracias a estos avances, más pacientes viven más tiempo y con mejor calidad de vida que hace una década. Además, actualmente existen numerosos fármacos aprobados que han ampliado significativamente las opciones para cáncer de pulmón avanzado [8].

A pesar de estos progresos, persisten retos importantes. El pronóstico en enfermedad metastásica sigue siendo pobre en la mayoría de los casos, las tasas de curación y supervivencia a largo plazo permanecen bajas [8]. Muchos pacientes eventualmente desarrollan resistencia a las terapias dirigidas o a la inmunoterapia, lo que limita la duración de la respuesta clínica. Además, no todos los pacientes responden a los nuevos tratamientos. Otro desafío es la heterogeneidad tumoral: los cánceres de pulmón presentan múltiples subclonas genéticas y características moleculares variables, dificultando los tratamientos uniformes. Sumado a ello, las metástasis cerebrales frecuentes en cáncer de pulmón (especialmente en subtipos como adenocarcinoma) siguen representando un obstáculo, ya que la penetración de fármacos al sistema nervioso central es limitada. En el ámbito del diagnóstico, mejorar la sensibilidad de las técnicas de imagen y desarrollar biomarcadores circulantes (como ADN tumoral circulante) para monitorear la enfermedad mínima residual son áreas de investigación activa. Por último, persisten disparidades en el acceso a los avances: a nivel mundial y regional, no todos los pacientes pueden acceder a centros con programas de cribado, pruebas moleculares y nuevas terapias, lo cual resulta en diferencias significativas en supervivencia [7].

De forma general, el cáncer de pulmón continúa siendo un problema de salud global de primera magnitud, con tendencias recientes alentadoras pero aún con desafíos pendientes. La detección temprana, la personalización del tratamiento según biomarcadores y el desarrollo de nuevos abordajes (como terapias celulares o vacunas anticáncer) representan las líneas principales para seguir mejorando el pronóstico de esta enfermedad en los próximos años. Además, la enorme complejidad biológica del cáncer de pulmón requiere asimismo enfoques integradores de datos y conocimiento. En este contexto surgen tecnologías como los grafos de conocimiento biomédico y las redes neuronales de grafos, que se exploran en las siguientes secciones, para aprovechar la gran cantidad de información disponible y apoyar nuevos descubrimientos en la investigación del cáncer de pulmón.

2.2 Grafos de conocimiento

En los últimos años, el uso de grafos de conocimiento (Knowledge Graphs, KG) se ha consolidado en informática biomédica como un medio para representar y explotar información compleja. Un KG es una red dirigida de nodos (entidades) y aristas (relaciones) cuyo propósito es acumular y comunicar conocimiento estructurado del mundo real [9]. Así, un hecho del mundo real puede describirse mediante una tripleta (sujeto, predicado, objeto) y al conjunto de muchas de estas relaciones interconectadas lo denominamos grafo de conocimiento. El concepto no es completamente nuevo (se relaciona con las antiguas redes semánticas de la Web Semántica), pero fue popularizado a gran escala en 2012 cuando Google anunció su Knowledge Graph para enriquecer los resultados de su buscador [9]. Desde entonces, otras grandes plataformas (Bing, Yahoo, etc.) y numerosos proyectos académicos han adoptado este enfoque para integrar y conectar datos de distintas fuentes.

Características y estructura

Los grafos de conocimiento suelen incluir no solo las entidades y sus relaciones, sino también información de tipo semántico (por ejemplo, jerarquías de clases u ontologías) que aporta contexto a los nodos y aristas. A menudo se construyen sobre esquemas formales (ontologías) que definen las categorías de entidades y tipos de relaciones permitidas [9]. Esto permite organizar el conocimiento de forma consistente y realizar inferencias lógicas. Gracias a esta estructura flexible, un grafo de conocimiento puede integrar datos heterogéneos: enlazar potencialmente cualquier entidad con cualquier otra mediante alguna relación significativa [10]. Otra característica es que permiten relaciones multiniveles o multirrelacionales; es decir, se pueden describir diferentes tipos de conexiones entre las mismas entidades sin problemas. Esto es especialmente útil en biomedicina, donde por ejemplo dos genes pueden relacionarse de múltiples formas. Muchos grafos de conocimiento incorporan también mecanismos para deducir nuevo conocimiento implícito a partir del existente, usando motores de inferencia o reglas lógicas [10].

Usos generales

Por su capacidad para modelar conocimiento de forma estructurada, los grafos de conocimiento se han empleado en muy diversos ámbitos. En motores de búsqueda y asistentes inteligentes, permiten responder preguntas complejas conectando información de distintas fuentes. En finanzas, se usan para relacionar entidades económicas y detectar fraudes; en industria, para gestión del conocimiento y mantenimiento predictivo, etc. Su valor radica en que facilitan la navegación y consulta relacional de datos. Asimismo, los grafos de conocimiento sirven de base para algoritmos de aprendizaje automático que requieren datos estructurados: por ejemplo, recomendaciones (link prediction), búsqueda semántica, o aprendizaje de representaciones (embeddings) de las entidades.

Grafos de conocimiento en biomedicina

El campo biomédico ha adoptado con entusiasmo esta herramienta debido a la enorme cantidad de datos dispersos en bases de datos, publicaciones y registros clínicos. Un grafo de conocimiento biomédico integra conceptos biomédicos y relaciones entre ellos representándolos como un grafo unificado [10]. Esto permite tener una visión global del conocimiento que antes estaba fragmentado. Históricamente, muchos de estos grafos se construyeron integrando bases de datos curadas manualmente por expertos (bases de conocimiento como UniProt para proteínas, OMIM para genes y enfermedades, DrugBank para fármacos, etc.) [10]. Una vez unificados en un grafo, se puede explorar, por ejemplo, qué genes están asociados a qué patologías, o qué rutas metabólicas son alteradas por ciertos fármacos, todo dentro del mismo marco de datos. En años recientes, con los avances en procesamiento de lenguaje natural, también se emplean métodos automáticos para extraer relaciones desde la literatura científica (text mining) e incorporarlas al grafo [10], enriqueciendo así la base de conocimiento de forma más rápida.

2.3 Graph Neural Networks (GNN)

Una Graph Neural Network (GNN) es una familia de arquitecturas de aprendizaje profundo diseñadas para operar directamente sobre estructuras de grafo, donde cada nodo actualiza su representación vectorial mediante un proceso iterativo de paso de mensajes (*message passing*) con sus vecinos y una función de agregación invariante a permutaciones [11]. Este mecanismo permite capturar dependencias no euclidianas presentes en la topología del grafo y aprovecharlas para tareas como clasificación de nodos, predicción de enlaces y generación de nuevas estructuras [12], lo que la hace adecuada para una amplia variedad de dominios, desde redes sociales hasta química computacional [13]. Gracias a su flexibilidad y a la disponibilidad de bibliotecas como PyTorch Geometric y DGL, las GNN se han convertido en el paradigma dominante para el aprendizaje basado en grafos en la última década.

Estado actual de la investigación

En los últimos años se ha consolidado el uso de grafos de conocimiento en el ámbito biomédico para descubrir relaciones inéditas entre entidades como fármacos, genes y enfermedades. Diversos estudios han aplicado métodos de descubrimiento de información sobre estos grafos con resultados prometedores. Aisopos y Paliouras (2023) comparan enfoques de *embeddings* de grafo con métodos basados en rutas lógicas para predecir interacciones fármaco-gen desconocidas, evidenciando un compromiso entre la precisión predictiva y la explicabilidad de los resultados [14]. En la misma línea, Bang et al. (2023) proponen extender el principio de “culpabilidad por asociación” en múltiples capas de un grafo biomédico para el *drug repurposing*, mejorando hasta un 16,8% la precisión en la predicción de asociaciones fármaco-enfermedad respecto a modelos previos [15]. Estas investigaciones ilustran cómo los grafos de conocimiento pueden impulsar el avance biomédico. Al mismo tiempo, han

resaltado desafíos, como la necesidad de balancear rendimiento con interpretabilidad en modelos puramente basados en representaciones latentes. En este contexto, las Graph Neural Networks (GNN) han emergido como una herramienta poderosa para aprender directamente de la estructura del grafo: el modelo *Decagon* de Zitnik et al. marcó un hito al utilizar convoluciones en grafos sobre una red multimodal de proteínas y fármacos para predecir efectos secundarios por interacciones farmacológicas, superando baselines convencionales hasta en un 69% [16]. Las GNN capturan patrones complejos y relacionales en los datos biomédicos; sin embargo, con frecuencia lo hacen a costa de una menor transparencia en la toma de decisiones. Como respuesta, han cobrado fuerza los enfoques híbridos neuro-simbólicos que combinan el aprendizaje profundo en grafos con técnicas simbólicas basadas en reglas lógicas, buscando aprovechar lo mejor de ambos mundos [17]. Un ejemplo representativo es AnyBURL, una técnica *bottom-up* capaz de extraer reglas lógicas de un grafo mediante recorridos aleatorios y evaluar su confianza para inferir nuevas relaciones [18]. Integrar este tipo de reglas con modelos neuronales en grafos permite enriquecer las predicciones con explicaciones basadas en conocimiento explícito, a la vez que se mantiene un alto desempeño predictivo, como se puede apreciar en la Figura 3 [22][23]. De hecho, se considera que la combinación de GNN con razonamiento lógico puede mejorar la fiabilidad e interpretabilidad de la inferencia en grafos de conocimiento [17]. Siguiendo esta tendencia, el presente trabajo adopta un modelo híbrido (AnyBURL + GNN) para el problema de *path ranking multi-hop* en un grafo biomédico de cáncer de pulmón. La evaluación se realiza con métricas estándar como Mean Reciprocal Rank (MRR) y Hits@k, que miden la calidad del ranking de las predicciones.

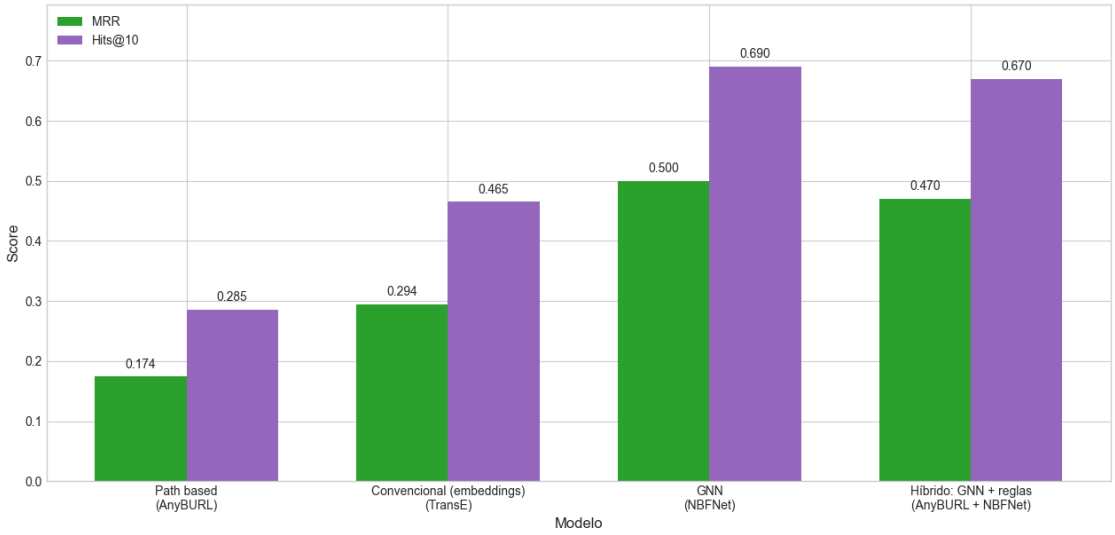


Figura 3: Comparación de modelos sobre el benchmark FB15k-237

En el ámbito oncológico, el uso de grafos de conocimiento y GNN ha demostrado un potencial significativo para descubrir información relevante y acelerar hallazgos biomédicos. Inicialmente, muchos esfuerzos se enfocaron en tareas generales como la reproposición de fármacos o la predicción de interacciones moleculares, pero más recientemente han surgido aplicaciones enfocadas en casos concretos de cáncer. Gogleva et al. (2022) construyeron un sistema de recomendación sobre un grafo heterogéneo de gran escala (integrando 37 fuentes de datos clínicos, preclínicos y bibliográficos) para priorizar genes potencialmente implicados en la resistencia a terapias dirigidas contra el cáncer de pulmón no microcítico [19]. Este enfoque basado en grafo permitió identificar 57 marcadores de resistencia entre más de 3.000 genes candidatos, incluyendo mecanismos previamente no reportados, reduciendo el tiempo de identificación de meses a minutos [19]. Asimismo, otras investigaciones han explorado factores epidemiológicos y de riesgo mediante grafos: Chen et al. (2022) combinaron datos de historias clínicas electrónicas con la ontología UMLS para construir un grafo centrado en pacientes de cáncer de pulmón, mapeando la distribución de factores de salud con el objetivo expreso de detectar indicadores clave de riesgo en el cribado preventivo [20]. En paralelo, modelos basados en grafos heterogéneos han logrado incorporar información clínica y molecular para mejorar el diagnóstico temprano; por ejemplo, un modelo GNN reciente integró datos metabolómicos y demográficos en un grafo para distinguir pacientes con cáncer de pulmón inicial, alcanzando un *test accuracy* del 89% y señalando la edad y ciertos biomarcadores metabólicos como predictores prominentes [21]. En conjunto, estos avances evidencian la versatilidad de las técnicas basadas en grafos en el dominio del cáncer: desde la identificación de nuevas relaciones fármaco-objetivo o genes asociados a resistencia, hasta la detección de patrones de riesgo en poblaciones de pacientes. Todo ello refuerza la motivación y el enfoque de este proyecto, que se sitúa sobre los hombros de dichas contribuciones previas.

3 Análisis de requisitos y tecnologías empleadas

En este capítulo se describen las decisiones tecnológicas tomadas para desarrollar el proyecto, justificando la selección de herramientas y metodologías en base a los requisitos planteados. Además, se discuten brevemente otras alternativas consideradas y las razones por las que fueron descartadas.

3.1 Análisis de requisitos del proyecto

Antes de seleccionar las tecnologías, se realizó un análisis de requisitos para entender qué demandas funcionales y no funcionales debía cumplir el sistema. A continuación se detallan dichos requisitos.

3.1.1 Requisitos funcionales

Los requisitos funcionales definen las capacidades y comportamientos específicos que el sistema debe ofrecer:

- **RF1:** integración de grafo de conocimiento biomédico. El sistema debe poder conectarse al grafo de conocimiento sobre cáncer de pulmón proporcionado por la cotutora del TFG y extraer de él la información relevante de entidades y sus relaciones. Esto implica consultar una base de datos de grafos mediante SPARQL, obteniendo como resultado un conjunto de tripletas (sujeto, relación, objeto) que servirán de entrada al modelo.

- **RF2:** modelo híbrido neuronalsimbólico. El núcleo del sistema es un modelo de aprendizaje que debe combinar técnicas de aprendizaje profundo (Graph Neural Networks) con reglas lógicas simbólicas extraídas del grafo. Funcionalmente, el modelo debe aceptar como entrada tanto la estructura del grafo como un conjunto de reglas o patrones lógicos, integrándolos para predecir nuevas asociaciones no explícitas en el grafo.

- **RF3:** entrenamiento y descubrimiento de información. El sistema debe permitir entrenar el modelo híbrido con datos históricos (tripletas existentes en el grafo) y evaluar su capacidad de descubrimiento de caminos asociados a los distintos tipos de cáncer de pulmón. Esto implica generar ejemplos positivos y negativos (mediante *negative sampling*) y optimizar el modelo para distinguir asociaciones verdaderas de las aleatorias. Además, se deben calcular métricas de rendimiento,

como Mean Reciprocal Rank (MRR) y Hits@K, durante la validación y finalmente obtener, para cada enfermedad del grafo, un *ranking* de los posibles factores de riesgo (entidades candidatas) con su puntaje de confianza.

- **RF4:** persistencia de resultados y modelo. Tras el entrenamiento, el sistema debe guardar el modelo aprendido para futuras consultas, así como exportar los resultados en un formato adecuado para posibles futuros análisis por expertos.

- **RF5:** descubrimiento de factores de riesgo mediante *path ranking*. Debe identificar y puntuar caminos multi-hop (*metapaths*) en el grafo que conecten entidades candidatas con las enfermedades objetivo, destacando aquellos factores de riesgo potenciales. El sistema ha de ser capaz de explorar la estructura del grafo en múltiples saltos para encontrar patrones de relaciones asociados a la enfermedad.

3.1.2 Requisitos no funcionales

Los requisitos no funcionales describen criterios de calidad y restricciones técnicas que debe respetar la solución:

- **RNF1:** rendimiento y escalabilidad. El volumen de datos en el grafo biomédico es considerable (millones de tripletas). El sistema debe ser capaz de procesar estas tripletas y entrenar el modelo en tiempos razonables. Esto exige utilizar algoritmos y estructuras de datos eficientes, así como aprovechar aceleración por *GPU* cuando sea posible.

- **RNF2:** mantenibilidad y reproducibilidad. El pipeline del proyecto debe estructurarse en fases claras (extracción de datos, preparación, entrenamiento, evaluación) con código modular, facilitando futuras modificaciones o mejoras. Se ha priorizado el uso de librerías estándar y configuraciones externas (archivos de propiedades, YAML) para que los experimentos sean reproducibles y los parámetros ajustables sin cambiar el código fuente.

- **RNF3:** portabilidad. Las herramientas escogidas deben ser multiplataforma y de código abierto en la medida de lo posible, evitando dependencias privativas. Esto permitirá ejecutar el proyecto en diferentes entornos (entorno local, servidores de la universidad, etc.) con mínima fricción. Python cumple este requisito, y librerías como PyTorch o pandas son ampliamente soportadas.

- **RNF4:** precisión del modelo. Aunque es un requisito de calidad más que una característica técnica, se estableció como objetivo que el modelo híbrido alcance al menos un rendimiento comparable al estado del arte en problemas similares de caminos multi-hop en grafos de conocimiento. Esto guió la selección de algoritmos robustos que, según la literatura, aportan mejoras de precisión y cobertura de predicciones al combinarse

- **RNF5:** interpretabilidad. Dada la naturaleza sensible del dominio biomédico, es deseable que el modelo proporcione explicaciones o al menos indicios sobre por qué una entidad ha sido propuesta como factor de riesgo. Por ello, integrar reglas simbólicas debe mejorar la interpretabilidad de las predicciones, permitiendo rastrear qué caminos o patrones en el grafo respaldan una determinada inferencia.

3.2 Tecnologías y herramientas seleccionadas

Con base en los requisitos anteriores, se eligieron las siguientes tecnologías

3.2.1 Python

Python se adopta como lenguaje principal de desarrollo por su amplia popularidad en ciencia de datos e inteligencia artificial, así como por su ecosistema de bibliotecas científicas [24]. Python permite integrar fácilmente módulos de análisis de datos (como Pandas o NumPy) y bibliotecas de grafos o aprendizaje profundo en un solo entorno. Además, su sintaxis clara y su naturaleza interpretada facilitan la experimentación iterativa y la documentación. Python es multiplataforma y cuenta con numerosas herramientas para procesamiento numérico y de grafos [25].

Bibliotecas utilizadas

- **Argparse**: Permite definir y analizar argumentos de línea de comandos dentro de la librería estándar, generando automáticamente mensajes de ayuda y validación de tipos [26]. Es la base de muchas herramientas CLI en Python porque evita el *boiler-plate* y documenta los parámetros desde el propio código.
  Uso en el proyecto: Se utiliza para controlar parámetros como la ruta al grafo, número de épocas, tamaño de lote o semilla, de modo que los experimentos sean reproducibles sin necesidad de modificar el script.

- **Os**: El módulo ofrece una abstracción portable del sistema operativo: paths, variables de entorno, permisos, procesos, etc. Se combina a menudo con pathlib y shutil para orquestar flujos de trabajo basados en ficheros [27].
  Uso en el proyecto: Se emplea para descubrimiento de directorios de datos y creación carpetas de resultados.

- **Requests/yaml**: *Requests* simplifica las peticiones HTTP, mientras que *yaml* se encarga de leer ficheros YAML. Juntos suelen aparecer cuando se consultan APIs REST y se gestionan configuraciones externas.
  Uso en el proyecto: *Requests* lanza consultas SPARQL al endpoint del grafo de conocimiento y recupera los resultados. Los textos de las queries y los *endpoints* viven en ficheros *.yaml*, permitiendo cambiar la fuente de datos o añadir filtros sin editar código.

- **Pandas/Numpy**: *NumPy* aporta matrices *n-dim* de alta eficiencia y operaciones vectorizadas; *Pandas* construye sobre ello estructuras *Series/DataFrame* orientadas a análisis tabular [28]. Son el núcleo del ecosistema científico de Python y alimentan desde *ETL* hasta *machine learning*.
  Uso en el proyecto: Preprocesamiento de los resultados provenientes de SPARQL y preparación de tensores para PyTorch.

- **PyTorch**: Biblioteca de aprendizaje profundo desarrollada por Facebook AI que proporciona cálculo automático de derivadas (autograd), soporte para ejecución en GPU y un ecosistema de alto nivel (como torchvision, torchaudio, entre otros) [29]. Su enfoque *define-by-run* permite una mayor flexibilidad para la depuración y facilita la construcción de modelos dinámicos, a diferencia de los enfoques basados en grafos estáticos.
  Uso en el proyecto: Utilización de *Dataset* y *DataLoader* para *batching*, así como *pack_padded_sequence* y *pad_packed_sequence* para tratar descripciones de entidades de longitud variable. Además, se utiliza *torch.nn.functional* para concentrar activaciones, pérdidas y utilidades que implementan el entrenamiento de las GNN.

- **PyG (PyTorch Geometric)**: Librería construida sobre PyTorch para escribir y entrenar fácilmente Graph Neural Networks (GNNs) para un amplio rango de aplicaciones relacionadas con datos estructurados. Consta de varios métodos para el aprendizaje profundo en grafos y otras estructuras irregulares, también conocido como aprendizaje profundo geométrico, a partir de una variedad de artículos publicados [30].
  Uso en el proyecto: Utilización de GCNConv para propagar información entre nodos del grafo de cáncer de pulmón. Al integrarse con PyTorch, el mismo *optimizer* y ciclos de entrenamiento sirven tanto para redes clásicas como para la GCN del proyecto.

- **NetworkX**: Paquete de Python para la creación, manipulación y estudio de la estructura, dinámica y funciones de redes complejas. Resulta ligera y flexible para prototipado y visualización, aunque no está pensada para entrenamiento GPU [31].
  Uso en el proyecto: Ilustración de subgrafos explicativos, mejorando la intepretación de resultados

3.2.2 AnyBurl

AnyBURL (*Any-time Bottom-Up Rule Learning*) es un algoritmo propuesto por Meilicke et al. que extrae, a partir de trayectorias muestreadas en un grafo de conocimiento, reglas de Horn (cláusula de lógica de primer orden que contiene a lo sumo un literal positivo) con estimaciones aproximadas de confianza [32]. Su carácter *any-time* le permite devolver un conjunto inicial de reglas a los pocos segundos y refinarlo progresivamente, lo que resulta idóneo en grafos biomédicos de gran tamaño donde el tiempo de cómputo es limitado. En el marco del presente Trabajo Fin de Grado se ha invocado sobre el subgrafo extraído mediante el fichero de configuración *config-learn.properties*, obteniéndose reglas que explican de forma simbólica las correlaciones entre factores de riesgo y el cáncer de pulmón; estas reglas se emplean tanto para derivar nuevas hipótesis como para contrastar la coherencia de las predicciones numéricas de la GNN.

3.2.3 SPARQL

SPARQL es la especificación del W3C que define el lenguaje declarativo y el protocolo de consulta para grafos RDF, proporcionando operadores de coincidencia de patrones, agregación y construcción de nuevos grafos. Su sintaxis permite expresar consultas complejas de forma composicional y neutra respecto al almacenamiento, de manera que puede interrogar fuentes distribuidas o servicios federados [33]. En este proyecto se ha utilizado como puerta de entrada oficial al grafo LUCIA facilitado por la cotutora [40], formulando cláusulas *SELECT* para obtener subconjuntos de tripletas. Ello garantiza la correcta y segura extracción de los datos.

3.3 Tecnologías consideradas

Durante la fase de diseño se evaluaron también varias alternativas potenciales, que finalmente fueron descartadas por diversas razones:

- **PyKEEN**: Es un framework Python dedicado a *embeddings* de grafos de conocimiento (Knowledge Graph Embeddings) . Aunque PyKEEN facilita el entrenamiento de muchos modelos de embedding (TransE, ConvE, etc.), se estimó que no se ajusta bien al objetivo principal de este proyecto, que es el ranking de rutas multi-hop y la integración de reglas simbólicas. PyKEEN está orientado a aprendizaje sub-simbólico masivo, mientras que el enfoque híbrido elegido requiere manipulación explícita de reglas, por ello se descartó. Usar PyKEEN implicaría complejo trabajo adicional para extraer metapaths interpretables, y además desviaría la arquitectura hacia embeddings puros en lugar de razonamiento explicable.

- **Deep Graph Library (DGL)**: Esta biblioteca es un entorno alternativo a PyTorch Geometric para GNNs. DGL destaca por su eficiencia en grafos muy grandes y soporte multi-backend (TensorFlow, PyTorch). Se valoró DGL por sus optimizaciones y proyectos asociados (como DGL-KE para bioinformática). Sin embargo, se eligió PyTorch Geometric principalmente por la familiaridad con PyTorch y la simplicidad del API de PyG. Además, PyG ya cubría todos los requerimientos del modelo híbrido planificado. DGL no se utilizó, aunque podría explorarse en futuras iteraciones para escalabilidad extrema.

- **Neo4j**: Neo4j es una base de datos de grafos ampliamente conocida. Se evaluó la posibilidad de transladar el subgrafo biomédico extraído en Neo4j y aprovechar su lenguaje de consultas Cypher para extraer caminos. Sin embargo, se descartó porque el proyecto requería procedimientos de entrenamiento y análisis que van más allá de una consulta de base de datos relacional. Neo4j implicaría un entorno separado y aprendizaje específico de su API, además de posibles costos de licencia avanzados. Dado que las bibliotecas de Python ofrecen ya un manejo eficiente del grafo en memoria y la integración con modelos de ML, por lo que el uso de Neo4j no era necesario para lograr los objetivos investigativos.

- **Tableau y otras herramientas BI**: Se consideraron plataformas de análisis visual como Tableau o Power BI para la presentación de resultados al final del proyecto. Sin embargo, estas herramientas están orientadas a cuadros de mando interactivos más que a grafos complejos.

- **Otras bibliotecas y herramientas**: Algunas librerías adicionales fueron analizadas brevemente. Por ejemplo, StellarGraph (otro framework Python para grafos) o TensorFlow GNN podrían haber servido para modelos similares, pero no ofrecían ventajas claras sobre PyTorch Geometric en este caso. En cuanto a motores de visualización 3D o VR (p. ej. Gephi 3D), se descartaron por aumento de costo computacional al tratar con tantos datos.

En conclusión, las tecnologías seleccionadas cubren los requisitos del proyecto y cuentan con amplio respaldo en la literatura científica y técnica. Las alternativas consideradas resultaron menos adecuadas por limitaciones prácticas (integración, enfoque, etc.), por lo que se priorizó la stack elegida. El resultado es un sistema coherente y especializado para el análisis de ruta multi-hop en el grafo biomédico de cáncer de pulmón.

4 Metodología

Este capítulo detalla la metodología empleada para desarrollar el modelo híbrido. Inicialmente se describe la estructura del grafo, especificando las entidades y relaciones biomédicas seleccionadas, así como el proceso de obtención y preprocesamiento de los datos. Posteriormente, se explica la extracción y transformación de reglas mediante AnyBURL y su integración con redes neuronales de grafos (GNN). Finalmente, se presenta la arquitectura del modelo híbrido, incluyendo el entrenamiento, validación y evaluación mediante métricas estándar.

4.1 Estructura del grafo de conocimiento

La fuente de datos principal del TFG ha sido un grafo de conocimiento realizado por la cotutora Delia Aminta Moreno Perdomo como parte de su Tesis Doctoral [40].

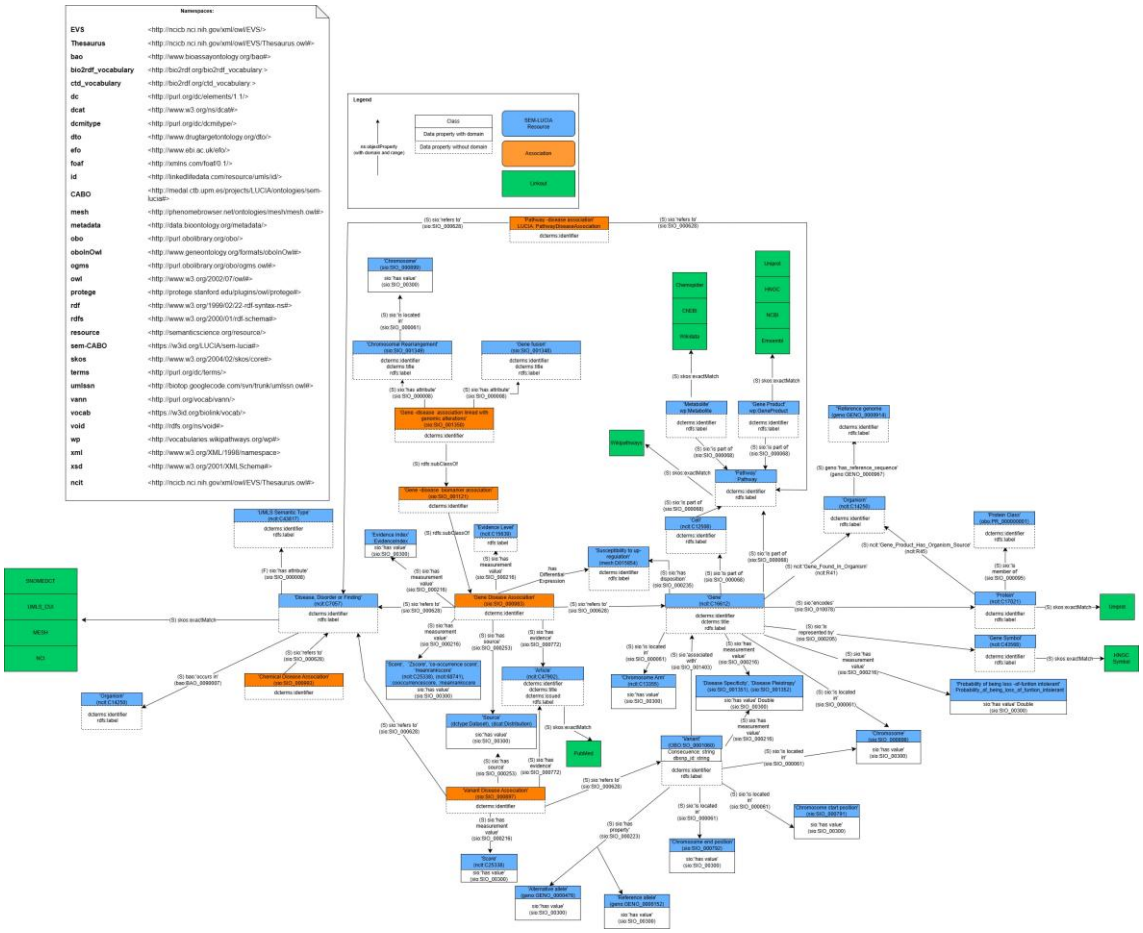


Figura 4: Esquema de la ontología LUCIA para cáncer de pulmón [40].

El grafo de conocimiento inicial se organiza siguiendo un esquema ontológico que modela entidades y relaciones relevantes para el cáncer de pulmón, como se puede apreciar en la Figura 4. En este grafo existen clases (*Disease*, *Gene*, *Protein*, etc.) que representan conceptos biomédicos fundamentales; propiedades de objeto o asociaciones que conectan estas clases entre sí (por ejemplo, la asociación de un gen con una enfermedad o de un gen con una proteína); propiedades de datos que enlazan entidades con valores literales concretos (como coordenadas numéricas en un cromosoma); y nodos literales que contienen valores específicos (cadenas de texto o números). De este modo, cada entidad semántica (nodo de clase) puede tener atributos cuantitativos o de texto asociados. En la práctica, este grafo integra información procedente de diversas ontologías y bases de datos biomédicas externas. Por ejemplo, utiliza vocabularios estandarizados como MeSH (Medical Subject Headings) o la NCIt (National Cancer Institute Thesaurus) para definir enfermedades y términos oncológicos, y recursos como UniProt o el catálogo de nombres genéricos para identificar genes y proteínas. Este enfoque fusiona datos curados de alto nivel (ontologías de enfermedades, repositorios de secuencias de proteínas, bases de datos de variantes genéticas, etc.), asegurando que el grafo contenga todas sus relaciones extraídas de fuentes reputadas. En términos generales, la literatura destaca que los conocimientos biomédicos relevantes para una enfermedad se encuentran dispersos en múltiples repositorios, por lo que la construcción de un grafo integrado como este aprovecha cruces ontológicos para unificar genes, proteínas, enfermedades, vías y otros tipos de entidades en un solo marco [34]. El resultado es un grafo rico en semántica, donde cada nodo de clase lleva identificadores y definiciones estandarizadas (proporcionadas por las ontologías externas) y cada relación refleja un dato biológico concreto.

Subgrafo reducido

El grafo original, con sus 226 millones de tripletas, resulta excesivamente voluminoso para el entrenamiento de un modelo de GNN: no sólo complica la computación, sino que introduce un alto grado de ruido al incorporar entidades y relaciones poco relevantes. Por ello, se ha optado por reducirlo, eliminando datos innecesarios y concentrando el análisis en las porciones más pertinentes al cáncer de pulmón, lo que facilita el entrenamiento y mejora la calidad de los resultados. Se ha seleccionado un subgrafo más pequeño enfocado en entidades clave para el cáncer de pulmón y el análisis con GNN, cuya representación visual se puede ver en la Figura 5.

Dentro del subgrafo reducido se consideran once entidades principales, cada una representada por una clase o propiedad relevante.

- La clase **Disease** corresponde a enfermedades; más concretamente representa las distintas variedades de cáncer de pulmón.

- La clase **Gene** representa a los genes, que son unidades de herencia formadas por secuencias de ADN. Cada gen contiene la información

necesaria para producir un producto funcional, típicamente una proteína. En el contexto del cáncer de pulmón, los genes seleccionados pueden ser oncogenes o supresores de tumores.

- La clase **GeneFusion** modela las fusiones génicas, es decir genes híbridos formados a partir de dos genes originales. Estas fusiones suelen originarse por reordenamientos cromosómicos (translocaciones, inversiones, etc.) y a menudo generan proteínas anormales asociadas a cáncer.

- La clase **Pathway** engloba a las vías biológicas o rutas metabólicas. Una vía biológica es una serie de interacciones entre moléculas dentro de la célula que conduce a un cambio específico, como la activación de ciertas funciones o la producción de moléculas nuevas

- La clase **Protein** corresponde a proteínas, que son moléculas grandes formadas por cadenas de aminoácidos. Las proteínas llevan a cabo la mayoría de las funciones biológicas esenciales en las células. En el grafo, una instancia de Protein refiere a una proteína específica derivada de un gen asociado al cáncer de pulmón.

- La clase **Chromosome** representa a los cromosomas, que son las estructuras organizadas de ADN y proteínas presentes en el núcleo celular. Un cromosoma es una larga hebra de ADN que contiene muchos genes y secuencias reguladoras. En el grafo se utiliza como referencia para ubicar posiciones genómicas de interés.

- La clase **ChromosomalRearrangement** modela los reordenamientos cromosómicos, es decir alteraciones en la estructura de un cromosoma. Estas modificaciones (como translocaciones, inversiones, duplicaciones o deleciones) cambian el orden normal de los genes en el cromosoma

- La entidad **SusceptibilityToUpDownRegulation** indica la propensión de un gen a ver alterada su expresión génica bajo ciertos factores. En otras palabras, refleja si un gen es susceptible a ser regulado al alza (up-regulation) o a la baja (down-regulation) por influencias genéticas o ambientales.

- La entidad **Variant** representa una variante genética: una diferencia en la secuencia de ADN de un gen con respecto a la referencia. Estas diferencias pueden ser mutaciones puntuales, inserciones, deleciones u otras alteraciones menores.

- Finalmente, **ChromosomeStartPosition** y **ChromosomeEndPosition** son propiedades de datos que indican las coordenadas genómicas exactas (valores numéricos) donde inicia y termina un elemento en el cromosoma. Estas posiciones literales permiten localizar con precisión cada evento en el genoma.

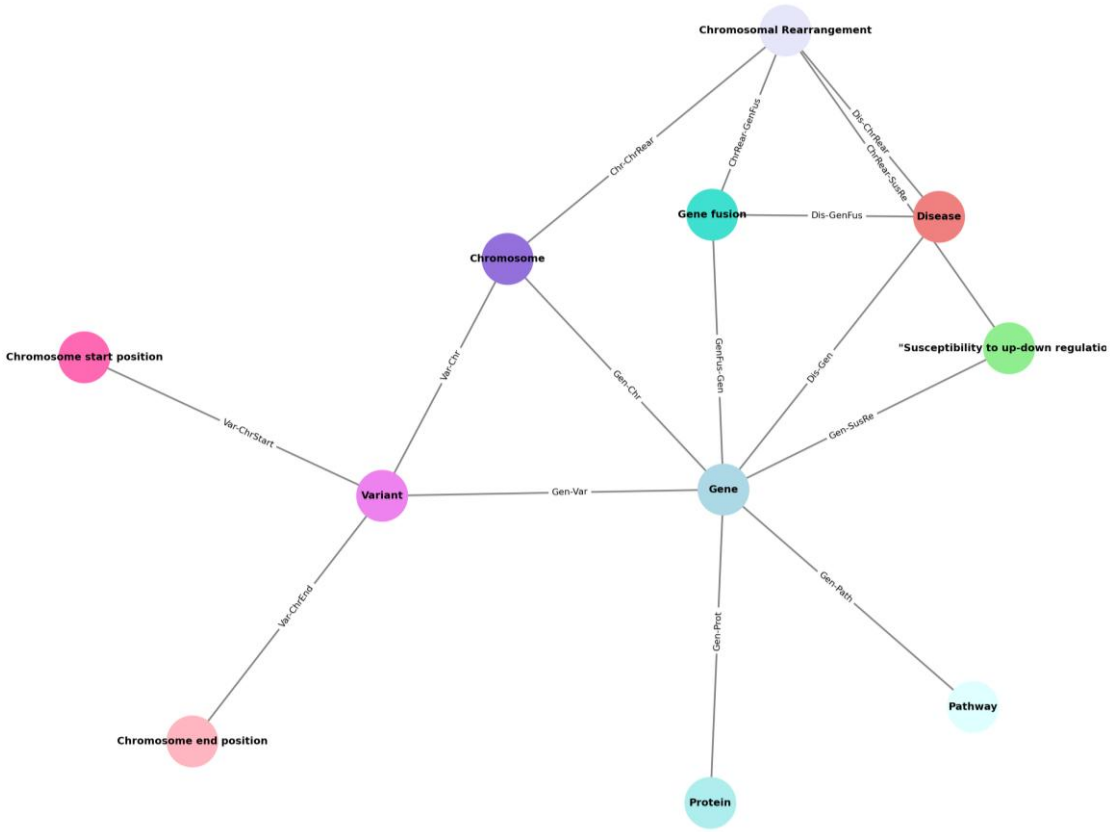


Figura 5: Subgrafo seleccionado para alimentar al modelo

En conjunto, estas entidades brindan una visión integrada de los elementos clave para estudiar el cáncer de pulmón mediante técnicas de grafos. Al centrarse en estas entidades, el subgrafo ofrece un esquema claro donde un algoritmo de GNN puede aprender patrones relevantes.

4.2 Extracción y preprocesamiento de datos

Este apartado describe detalladamente el proceso seguido para obtener y preparar los datos empleados en la construcción del modelo. Incluye la extracción mediante consultas SPARQL, el preprocesamiento de las tripletas obtenidas y su transformación a formatos adecuados para el entrenamiento con redes neuronales de grafos (GNN).

4.2.1 Obtención de datos mediante SPARQL

La extracción de los datos se ha realizado mediante un script de Python que se encarga de lanzar las consultas SPARQL al endpoint del grafo de conocimiento RDF alojado en la nube. SPARQL es el lenguaje estándar para interrogar bases de datos RDF, y permite especificar patrones de tripletas (cabeza-relación-cola) para extraer la información deseada. Se han definido múltiples consultas SPARQL, en concreto, se han definido 15 consultas, una por cada relación existente en el subgrafo. Cada consulta ha sido ejecutada sobre el endpoint, recuperando el conjunto de tripletas correspondientes al patrón especificado. Dada la heterogeneidad en la cantidad de datos por relación y para evitar el desequilibrio de clases al entrenar modelos de predicción de enlaces, se ha aplicado un muestreo aleatorio sobre los resultados de cada consulta. En concreto, se ha limitado a 50.000 el número de tripletas obtenidas por cada tipo de relación. Este submuestreo reduce la dominancia de relaciones con mayor cantidad de datos y mejora la eficiencia computacional del proceso, al controlar el tamaño del dataset resultante. Los resultados de cada consulta se han almacenado en archivos CSV individuales con tres columnas: head, relation y tail, de modo que cada fila representa una tripleta completa. Posteriormente, dichos archivos parciales han sido concatenados para obtener un único fichero de dataset consolidado. El archivo final contiene todas las tripletas extraídas, organizadas en forma de filas (cabeza, relación, cola), listo para las etapas posteriores de preprocesamiento.

4.2.2 Preprocesamiento de los datos

El conjunto de datos crudo obtenido de las consultas SPARQL ha sido sometido a varias fases de preprocesamiento para garantizar su calidad y adecuarlo al modelado. En primer lugar se ha aplicado limpieza y normalización de nombres: se eliminan duplicados y se unifican los formatos de los identificadores de entidades y relaciones (se homogenizan mayúsculas/minúsculas, se eliminan caracteres especiales innecesarios, etc.), con el fin de evitar inconsistencias semánticas.

A continuación, se ha procedido a la adición de relaciones inversas. Por cada tripleta original *(cabeza, relación, cola)* se genera la tripleta inversa correspondiente *(cola, relación^{-1}, cabeza)*. Este duplicado simétrico de las relaciones es una práctica común en tareas de predicción de enlaces sobre grafos, ya que proporciona al modelo información bidireccional y mejora su capacidad de inferencia. En otras palabras, al incluir explícitamente la relación inversa, el modelo aprende que si la entidad A se relaciona con B, entonces también existe una relación inversa de B hacia A. Esta estrategia ayuda a que el grafo sea tratado de forma más simétrica.

Posteriormente, el dataset enriquecido con las relaciones inversas se ha dividido aleatoriamente en tres subconjuntos: entrenamiento, validación y prueba. Esta partición se ha realizado respetando proporciones típicas de evaluación (80 % para entrenamiento, 10 % validación y 10 % prueba), garantizando que cada

subconjunto contenga una representación equilibrada de los diferentes tipos de relaciones. Esta división permite entrenar los modelos sobre el conjunto de entrenamiento y evaluar su desempeño con las particiones de validación y prueba, evitando filtraciones de información.

A continuación se han creado diccionarios de mapeo que asignan un identificador numérico único (ID) a cada entidad y a cada relación presente en el grafo. Es decir, se recorre el grafo completo de tripletas y se enumeran todas las entidades, asignándoles IDs contiguos (enteros desde 0 hasta *N–1*), y se hace lo mismo con las relaciones. Este proceso genera dos tablas de correspondencia (entidad→ID y relación→ID) que permiten convertir las descripciones textuales en índices numéricos. La creación de estos diccionarios de IDs es una práctica estándar en el aprendizaje de grafos, pues facilita la conversión de datos simbólicos a formatos numéricos para los modelos.

4.2.3 Transformación a tensores

Una vez preprocesados los datos, se han transformado los datos a un formato entendible por la GNN.

Clase KnowledgeGraph

La clase KnowledgeGraph se encarga de representar internamente el grafo de conocimiento multipartito a partir de las tripletas de entrenamiento, validación y test. En concreto, carga por separado las tripletas (h, r, t) de cada partición y construye estructuras de listas de adyacencia separadas según el tipo de relación. Esto significa que para cada relación **r** se almacena la lista de aristas dirigidas correspondientes, típicamente como pares (cabeza, cola). A partir de estas listas se calcula el grado de salida de cada nodo, definido como el número de vecinos a los que se conecta dicho nodo. Además, se construyen diccionarios auxiliares que mapean cada par (cabeza, relación) a su conjunto de colas asociadas (y análogamente pares inversos (relación, cola) a cabezas), lo que permite acceder rápidamente a todas las colas verdaderas dadas una entidad y una relación. Estos diccionarios se utilizan más adelante para enmascarar tripletas observadas y evitar muestreos negativos inválidos durante el entrenamiento y la evaluación.

Este diseño permite acceder rápidamente a la estructura relacional del grafo, conocer los grados de los nodos y aplicar técnicas de muestreo y propagación basadas en las conexiones reales registradas.

Codificación y decodificación de pares

Se define un esquema de codificación combinada para pares de la forma (cabeza, relación) o (cabeza, cola): por ejemplo, se codifica el par (h, r) como un único índice en una tabla (por ejemplo id = h * n_relaciones + r) y también codificar (h, t) de forma similar. Esto permite un acceso directo a registros en estructuras de datos internas sin mantener como clave pares de Python. El proceso inverso,

de decodificación, convierte un índice único de vuelta en el par original, lo cual es útil en fases de interpretación o evaluación. La codificación de pares es clave para generar lotes de entrenamiento: se puede tratar cada par codificado como una instancia a la que asociar múltiples objetivos (colas verdaderas y negativas).

Conversión a tensores de adyacencia

Las listas de adyacencia por relación se convierten a tensores numéricos para ser usadas en la propagación de información en el modelo GNN. Por cada relación *r*, las listas de cabezas y colas se empaquetan en tensores de enteros (*edge_index* en PyTorch Geometric) que representan gráficamente las aristas. Estos tensores de adyacencia permiten que el modelo realice operaciones de *message passing*: en cada capa GNN, la representación de un nodo se actualiza mediante una función de agregación de mensajes enviados por sus vecinos, según la topología almacenada en esos tensores. En otras palabras, los tensores de adyacencia codifican la estructura del grafo para que las capas convolucionales puedan iterativamente propagar y combinar la información de los nodos a través de las aristas. Este uso tensorizado de la adyacencia hace posible aplicar eficientemente el aprendizaje profundo sobre el grafo sin bucles explícitos en Python.

Clases TrainDataset, ValidDataset y TestDataset

Estas clases heredan típicamente de *torch.utils.data.Dataset* y organizan los datos de entrenamiento, validación y test para el entrenamiento de los modelos. La idea general es agrupar las tripletas o pares (h, r) por relación y crear lotes (*batches*) de entrenamiento de forma que cada lote contenga ejemplos correspondientes a una o varias relaciones. Concretamente:

- Estructura por relación: los datos se agrupan por tipo de relación de modo que cada ejemplo contiene una cabeza y una relación (con una lista de colas verdaderas). De este modo, es sencillo generar en un lote tanto los ejemplos positivos (tripletas reales) como ejemplos negativos alterando las colas. Este diseño por relación mejora la eficiencia al generar ejemplos negativos consistentes con las relaciones.

- Construcción de lotes: cada lote reúne varios ejemplos (h, r) y sus colas correspondientes. En el caso del conjunto de entrenamiento, a cada ejemplo positivo se le añaden k ejemplos negativos (ver siguiente sección). Además, se define un método *__getitem__* que devuelve los tensores de cabezas, relaciones, colas verdaderas y máscaras de muestras negativas para un índice dado. También se define una función *collate_fn* estática que toma una lista de tales ejemplos y construye un minibatch concatenando tensores

- Máscaras de filtrado en evaluación: durante la evaluación (conjuntos de validación y test) se emplea una "máscara filtrada" para evitar contabilizar como falsos negativos aquellas colas que en realidad ya existen en el grafo. En la práctica, esto se logra usando los diccionarios auxiliares de tripletas verdaderas construidos en KnowledgeGraph. Al computar la puntuación de un candidato t para la consulta (h,r,?), se descartan de la lista de candidatos todas las colas que aparecen en la partición de entrenamiento (y validación, si se evalúa contra test) para el mismo (h,r). De esta forma, el modelo no es penalizado por rankear relativamente alto una cola verdadera distinta de la buscada.

Muestreo negativo

El muestreo negativo es una técnica empleada para generar ejemplos falsos durante el entrenamiento y facilitar el entrenamiento contrastivo. En cada paso de entrenamiento, a partir de una tripleta positiva (h, r, t), se generan varias tripletas negativas reemplazando la cola (o alternando la cabeza) por entidades aleatorias. El objetivo es que el modelo aprenda a asignar puntuaciones altas a las tripletas reales y bajas a las corruptas.

Clase RuleDataset

La clase RuleDataset está diseñada para manejar reglas de inferencia multi-hop (cadenas de relaciones) como entradas de datos. Cada regla se representa internamente como una secuencia ordenada de relaciones [r_1, r_2, ..., r_m, r_objetivo], donde r_1...r_m forman el cuerpo y r_objetivo la cabeza de la regla. Para permitir procesar reglas de diferentes longitudes en batch, se aplica *padding* con un token especial hasta una longitud máxima 3 (para evitar ciclos). El dataset construye lotes de reglas empaquetando varias secuencias (codificadas como índices de relación) en tensores de tamaño uniforme, junto con máscaras que indiquen la longitud real de cada regla. Esto permite alimentar estas reglas al modelo de manera vectorizada.

Funciones grounding y propagate

Las funciones grounding y propagate sirven para recorrer y verificar caminos en el grafo durante el aprendizaje y la inferencia. En concreto, ***grounding*** toma una regla (secuencia de relaciones) y un par (cabeza, posible cola) y verifica si existe un camino en el grafo que instancie esa regla. Esto implica encontrar un encadenamiento de entidades intermedias que conecte el nodo h con el candidato t siguiendo las relaciones de la regla. Si todas las aristas correspondientes existen en el grafo, se dice que la regla queda "groundeada" en ese par. Así, grounding efectivamente enumera todos los caminos de largo fijo entre h y t que coinciden con la regla propuesta.

La función ***propagate*** se refiere típicamente a la ejecución de estos recorridos de manera algorítmica, propagando una señal desde el nodo inicial a lo largo de las aristas del grafo. En un enfoque clásico de *path ranking*, se inicia en h y se expanden sucesivamente todos los vecinos por cada relación de la regla,

filtrando aquellos caminos que dejen de coincidir. De esta manera, *propagate* efectúa un recorrido por niveles en el grafo, actualizando los conjuntos de nodos alcanzables por cada paso de la regla. Tanto *grounding* como *propagate* son importantes en el *path ranking algorithm*: permiten identificar las rutas relacionales que conectan dos entidades. Al combinar estas rutas con puntuaciones de reglas o embeddings, el modelo puede razonar sobre conexiones multi-hop en el grafo y mejorar la inferencia de tripletas faltantes.

4.3 Generación de reglas AnyBurl

Como se ha introducido anteriormente, AnyBURL es un algoritmo *bottom-up* de aprendizaje lógico por caminos, diseñado para extraer reglas a partir de grandes grafos de conocimiento. Básicamente, AnyBURL toma como entrada un grafo con hechos (tripletas h,r,t) y genera reglas de inferencia con variables. Estas reglas describen patrones frecuentes en el dominio. AnyBURL actúa en modo *anytime*, lo que significa que puede interrumpirse en cualquier momento y ofrece reglas progresivamente mejores, proporcionando explicaciones simbólicas de sus predicciones.

En este proyecto se ha utilizado AnyBURL para extraer reglas lógicas específicas del grafo de conocimiento. Para ello se ha configurado un archivo denominado config-learn.properties. En este fichero se ha indicado explícitamente la ruta al archivo de hechos de entrenamiento, se han definido exclusivamente relaciones orientadas desde factores de riesgo hacia enfermedades, concretamente aquellas relaciones que enlazan genes, reordenamientos cromosómicos, susceptibilidad y fusiones génicas (las únicas 4 directamente relacionadas) con enfermedades. Con el fin de mantener una complejidad manejable y una interpretación clara de las reglas generadas, se restringió el algoritmo a producir únicamente metapaths acíclicos de hasta tres saltos explicativos, con el fin de poder llegar a todos los nodos pero sin aumentar exponencialmente los caminos posibles (evitando ruido y alto coste computacional), y se desactivaron completamente las rutas cíclicas. Adicionalmente, se estableció un límite máximo de 2500 segundos (~41 minutos) para extraer las reglas. El proceso también aprovechó la paralelización estableciendo el uso de cuatro hilos de procesamiento (WORKER_THREADS = 4). Toda esta configuración garantizó que las reglas generadas fueran relevantes, precisas y apropiadas para integrarse en las siguientes etapas del proyecto. La ejecución se realizó con el comando:

```
java -Xmx10G -cp AnyBURL-23-1.jar de.unima.ki.anyburl.Learn config-learn.properties
```

Tras la ejecución de AnyBURL se obtiene un fichero de reglas extraídas en formato textual. Sin embargo, estas reglas requieren post-procesamiento antes de integrarlas al modelo final. Para ello se leyeron las reglas generadas por AnyBURL y las transforma en un formato estándar. En primer lugar, filtra reglas

triviales (score mínimo de 0.5) o con baja cobertura, y normaliza la notación de variables y relaciones. El objetivo es convertir las reglas libres de formato ambiguo a estructuras consistentes.

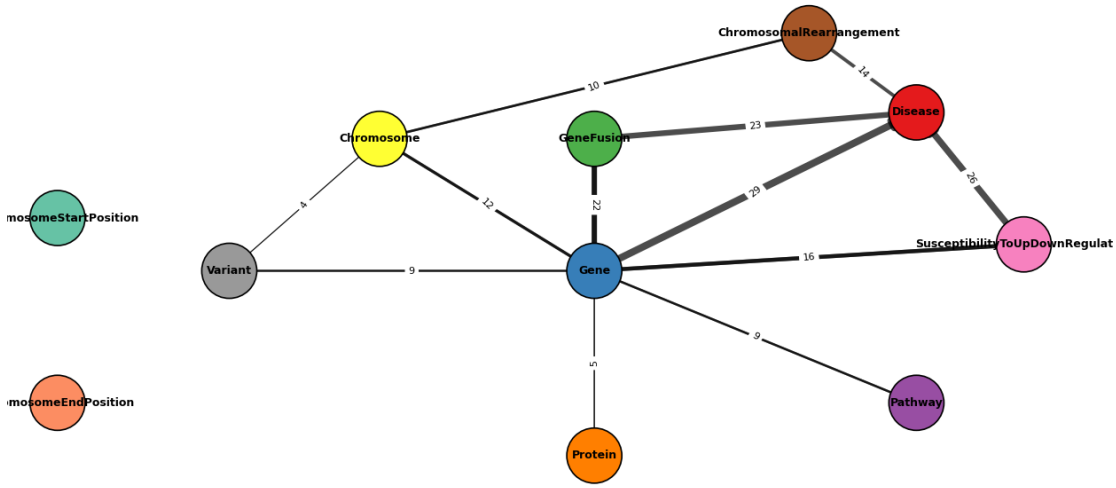


Figura 6: Grafo conectado mediante los metapaths minados por AnyBURL

La Figura 6 muestra los metapaths generales (sin entrar en entidades concretas) de las reglas extraídas. Se observa una red relativamente densa, varios nodos aparecen como centros de alta conectividad: en particular, se puede observar como se forma un triángulo de notable conectividad entre *Gene*, *GeneFusion* y *Disease*. Estos nodos indican que múltiples reglas involucran esas mismas entidades en sus antecedentes y conclusiones. Por otro lado, se puede apreciar como las coordenadas genómicas (*ChromosomeStartPosition* y *ChromosomeEndPosition*) parecen estar poco relacionadas con el cáncer de pulmón.

A continuación, estas reglas preprocesadas se integran en el modelo RNNLogic que se entrena con la red de conocimiento. Para conseguirlo, se convierte cada regla lógica en un predictor que pueda entender el modelo. En concreto, lee las reglas limpias del paso anterior y construye estructuras de datos que asocian cada regla a vectores. De este modo, cada regla se representa como un arreglo de índices (entidad–relación–entidad) con su confianza respectiva. Consecuentemente, se inicia el entrenamiento del modelo RNNLogic propiamente dicho. Este se encarga de cargar el grafo de conocimiento (hechos de entrenamiento) y los predictores generados, y de configurar los hiperparámetros del modelo. Durante su ejecución, se invoca la librería de RNNLogic, pasando las matrices de reglas como insumo. RNNLogic emplea estas reglas lógicas como conocimiento previo en la fase de predicción. En este paso final, las reglas lógicas se convierten en formas internas compatibles con la red neuronal, se construyen tensores que relacionan las entidades y relaciones de los predictores, junto con sus pesos asociados. De este modo, las reglas

originalmente extraídas se integran en la arquitectura del modelo de aprendizaje, permitiendo que el predictor lógico de RNNLogic las utilice para inferir hechos nuevos basados en los caminos definidos por dichas reglas.

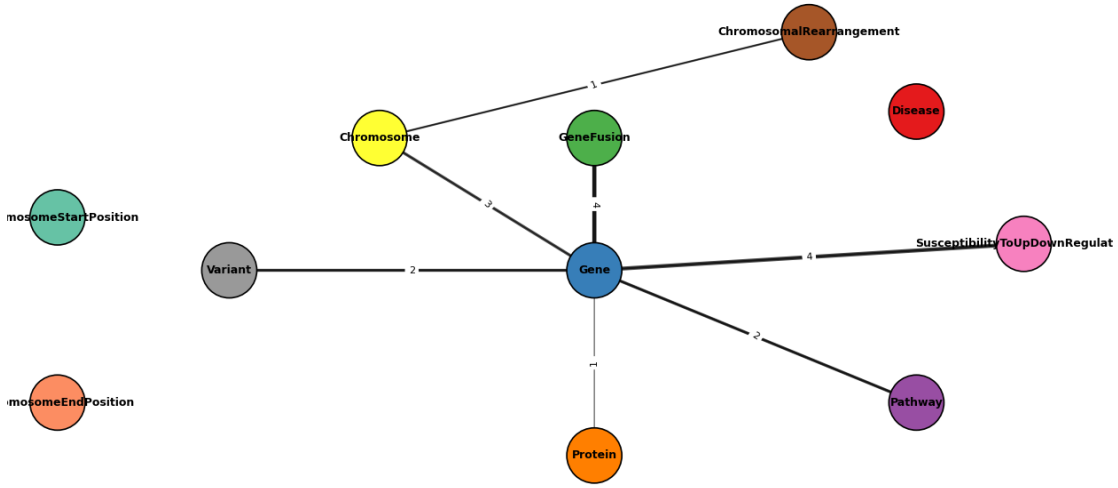


Figura 7: Grafo conectado mediante los metapaths finalmente evaluados

En esta Figura 7 de las reglas seleccionadas para el modelo final se aprecia una estructura más simplificada. Solo unas pocas entidades destacan como nodos centrales; muchas de las conexiones observadas en la Figura 6 han desaparecido o se han reducido. Esto se debe principalmente a dos razones, la primera y más notororia, se debe a que en estas reglas entrenadas se han eliminados las relaciones con el head, que correspondía al nodo *Disease,* esto se debe a que el modelo está diseñado para aprender patrones de relaciones que son predictivos, sin necesidad de especificar una relación final particular, y la segunda corresponde con la adición de un nuevo filtro de score sobre estas nuevas reglas seleccionadas después del entrenamiento.

4.4 Diseño del modelo híbrido

La arquitectura general integra una red neuronal de grafos (GNN) con predictores lógicos basados en reglas. Esto se justifica en que una GNN puede explotar la estructura topológica del grafo para aprender patrones latentes, mientras que los predictores simbólicos derivados de reglas inferencia lógica explícita. De este modo, se aprovechan las fortalezas de ambos paradigmas: la capacidad de generalizar desde datos numéricos complejos y la interpretabilidad de la inferencia basada en reglas. El modelo implementado define una red neuronal de grafo híbrida denominada **LungRiskHybridGNN**. Esta arquitectura se compone de varios bloques secuenciales, cada uno encargado de extraer y combinar distinta información:

Capa de embeddings iniciales

Cada nodo o entidad en el grafo se representa inicialmente mediante un vector denso de características. En la implementación, se crea un espacio de embedding donde cada nodo tiene un vector de dimensión fija (64). Esta capa transforma las entidades discretas en vectores continuos, sirviendo de entrada al resto de la red.

Capas GCN (Graph Convolutional Network)

Dos capas GCN que aplican operaciones de convolución sobre el grafo, combinando los embeddings iniciales con la topología de las relaciones. Cada capa GCN toma como entrada los vectores de los nodos y la matriz de adyacencia del grafo, y genera nuevas representaciones para cada nodo. Estas representaciones codifican tanto la estructura local del grafo como las características de los nodos. En concreto, cada capa realiza una agregación de los vectores de los vecinos de un nodo (según las aristas del grafo) y aplica una transformación lineal seguida de una activación ReLU. El resultado es un vector por nodo que captura información combinada de sus vecinos inmediatos y de su propio estado previo.

Módulo de agregación de reglas lógicas

Además de la GCN convencional, el modelo incluye un componente específico para incorporar conocimiento simbólico proveniente de reglas lógicas extraídas con AnyBURL. Este bloque toma las representaciones intermedias de los nodos (salida de las capas GCN) y las combina con información derivada de las reglas. Esta integración se realiza mediante la construcción y utilización de la matriz de agregación de reglas *A_fn*. El proceso es el siguiente:

- Extracción y transformación de reglas**:** previamente (4.3) se aplicó AnyBURL sobre el grafo de conocimiento, obteniendo un conjunto de reglas lógicas. Cada regla descubierta se asocia con una confianza o peso. Estas reglas se cargan y se transforman a un formato matricial. En la implementación, cada regla se representa como una fila de la matriz *A_fn*, y cada entidad/nodo relevante corresponde a una columna. Si la regla *i*-ésima se aplica o es activable para el nodo *j*-ésimo (por ejemplo, si *j* aparece en la premisa o conclusión de la regla según los datos de entrenamiento), entonces *A_fn[i,j] = score de confianza.* En caso contrario, el elemento es cero. Este proceso codifica en *A_fn* cuáles reglas están relacionadas con cada nodo. El resultado es una matriz dispersa de tamaño $R \times N$, donde R es el número de reglas y N el número de nodos (entidades de interés). Conceptualmente, *A_fn* puede interpretarse como un grafo bipartito entre reglas y entidades.

- Integración en la red**:** en el paso de inferencia (forward) del modelo, las representaciones de los nodos obtenidas por la GNN se combinan con las reglas usando *A_fn*. Concretamente, si denotamos por *H* la matriz de embeddings de nodos (filas = nodos, columnas = dimensión de representación) después de las capas GCN, entonces $H\_rule = A\_fn \cdot H$ genera un nuevo conjunto de vectores (uno por regla) que sintetizan la presencia de patrones lógicos. Estos vectores se vuelven a propagar o agregar de vuelta a los nodos. De esta forma, cada nodo final cuenta con información estructural procesada por la GCN y con señales derivadas de las reglas simbólicas.

- Interacción GCN – reglas**:** la GCN actúa sobre la topología original del grafo, mientras que *A_fn* incorpora conexiones basadas en reglas lógicas. En la práctica, el modelo *LungRiskHybridGNN* aplica primero la GCN tradicional para obtener embeddings intermedios y luego añade la información de reglas. Este ordenamiento modular facilita la implementación y permite ver claramente la contribución de cada componente. El modelo aprende pesos internos (en el MLP final) que ponderan adecuadamente la información estadística de la GCN frente al conocimiento simbólico de las reglas.

Perceptrón multicapa final (MLP)

Las salidas de la GCN y del agregador de reglas se combinan en un único vector por nodo. Este vector combinado alimenta un MLP de dos capas ocultas. El MLP aplica transformaciones lineales sucesivas (con activaciones ReLU intermedias) para producir la predicción final. En la última capa, se utiliza una función de activación sigmoide para normalizar los scores.

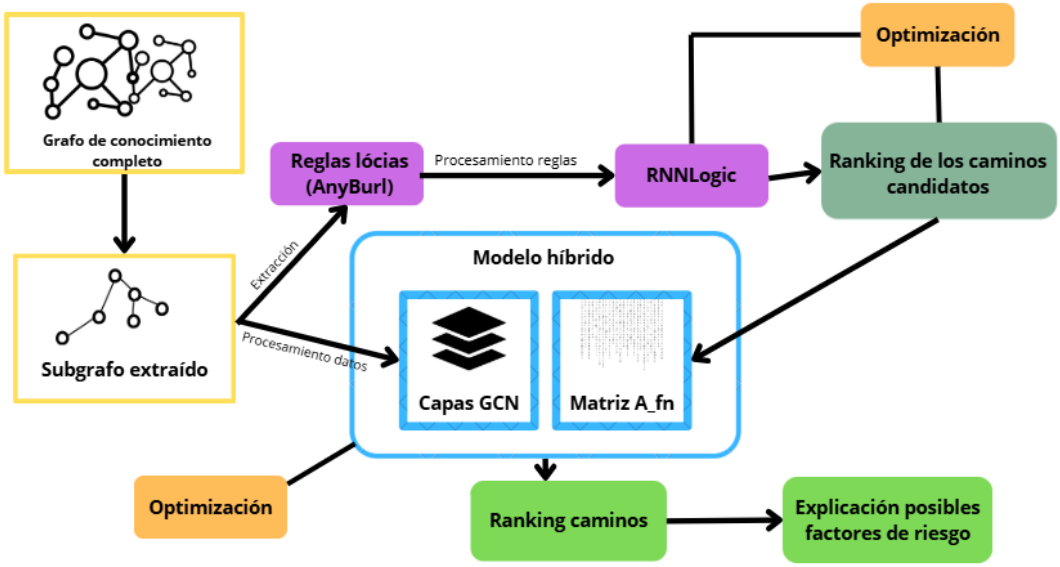


Figura 8: Arquitectura y flujo de procesamiento del modelo

En conjunto, la integración funciona como un modelo neuronal-simbólico: la GCN proporciona generalización a partir de los datos estructurados, y AnyBURL aporta conocimiento deductivo explícito. Este enfoque híbrido sigue la línea de trabajos recientes que destacan el valor de las reglas lógicas para explicar y reforzar las predicciones [35]. Un diagrama detallado del flujo de datos y la interacción de módulos podría ilustrarse en la Figura 8, donde se esquematizan los componentes principales y sus conexiones. Este diagrama clarificaría visualmente la arquitectura híbrida propuesta y el camino de la información hasta la salida del modelo.

4.5 Proceso de entrenamiento y validación

El entrenamiento del modelo se lleva a cabo siguiendo el procedimiento clásico de embedding de grafos. El flujo completo es el siguiente:

1. **Preparación de datos**
   A partir del archivo de datos (4.2.3), se generan los conjuntos de entrenamiento y validación. El grafo original se descompone en triples (nodo cabeza, relación, nodo cola) verdaderos. En cada época, para cada triple positivo extraído, se genera un conjunto de triples negativos mediante muestreo aleatorio: se corrompe la cabeza o la cola por una entidad distinta, creando así ejemplos falsos. Este procedimiento de "corrupting" es estándar en aprendizaje de grafos, como se ha explicado anteriormente. Cada lote de entrenamiento contiene tanto triples verdaderos como falsos, con sus respectivas etiquetas (1 para verdadero, 0 para falso).

2. **Iteración por épocas**
   El bucle principal recorre un total de 40 épocas. En cada época, por cada lote de entrenamiento se realiza:

   - Forward propagation: se calculan las representaciones de nodos iniciales y, mediante la GCN seguida del agregador de reglas, se obtienen las puntuaciones predichas para cada triple del lote.

   - Cálculo de la pérdida: se comprueba el rendimiento de diferentes funciones de pérdida: *margin* (margen dinámico basado en la dificultad del ranking), *BPR Loss* (-log(sigmoid(score_pos - score_neg)) y *listwise* (optimiza directamente MRR). La función que obtuvo resultados más eficientes fue esta última.

   - Backpropagation y actualización: se calcula el gradiente de la pérdida respecto a los parámetros del modelo (pesos de la GCN, embeddings, MLP, etc.) y se actualizan mediante un

optimizador (Adam) con tasa de aprendizaje fija (0.001). Se experimentó con tasa de aprendizaje dinámica pero no se obtuvieron los resultados esperados.

3. Validación periódica

En cada época completa se evalúa el rendimiento del modelo sobre el conjunto de validación. Durante la validación, no se actualizan parámetros; en su lugar, se calculan métricas de ranking estándar de tareas de completado de grafos, como MRR (Mean Reciprocal Rank) y Hits@K (para K = 1, 3, 10). Estas métricas se obtienen ordenando las predicciones de cada triple de validación frente a candidatos corruptos y midiendo en qué posición aparece la correcta. Se hicieron varias pruebas para acertar con el número de épocas con el objetivo de evitar un posible sobreajuste.

4. Resultados del entrenamiento

El entrenamiento se ha llevado a cabo sobre el 80% del grafo bidireccional completo, lo que supone que se utilizaron alrededor de 600 mil tripletas. El entrenamiento finalizó en menos de cinco horas en un equipo equipado con una GPU NVIDIA GeForce MX450 (2 GB de VRAM), un procesador Intel Core™ i7-1165G7 a 2,80 GHz, 16 GB de RAM (3 200 MHz).

Durante la última época, la pérdida media de entrenamiento ha descendido por debajo de 0.05, confirmando la convergencia estable del optimizador. En el conjunto de validación, el modelo híbrido ha alcanzado un MRR de 0.17, con Hits@1 = 0.05, Hits@3 = 0.13 y Hits@10 = 0.31. Estas métricas muestran un comportamiento consistente con trabajos previos en grafos de escala similar.

5 Evaluación

En esta sección se detalla la evaluación final del modelo. Se describe la metodología de evaluación, seguido de la presentación de los resultados cuantitativos y cualitativos obtenidos, y finalmente se ofrece una discusión de resultados . El objetivo es analizar tanto el rendimiento predictivo del modelo como la interpretabilidad de las explicaciones generadas por el componente simbólico.

5.1 Metodología de evaluación

Para evaluar el modelo se genera un ranking de entidades candidatas (factores de riesgo) para cada entidad objetivo (enfermedad) del conjunto de validación, basándose en las puntuaciones calculadas por el modelo híbrido. En este proceso se aplican los siguientes pasos: primero se calcula para cada enfermedad la afinidad o probabilidad de conexión con cada una de las 4 clases directamente conectadas; a continuación, se filtran las entidades ya conectadas con la enfermedad en el grafo de entrenamiento para evitar predecir enlaces ya conocidos. Esta técnica de filtrado asegura que sólo se evalúan enlaces “nuevos”, simulando la tarea real de descubrimiento de relaciones.

Luego, al combinar las predicciones de la GNN con las reglas lógicas, se utiliza la matriz *A_fn* derivada de las reglas de AnyBURL. En concreto, las puntuaciones iniciales de la GNN se reponderan o ajustan mediante las evidencias proporcionadas por las reglas lógicas: cada candidato recibe una puntuación compuesta que integra la inferencia numérica de la GNN y la inferencia simbólica de las reglas. De este modo, el método mezcla información estadística y lógica en la clasificación final de los factores de riesgo. Por último, se procesan estas predicciones finales y extrae ***metapaths*** explicativos para cada predicción relevante. Un *metapath* es una cadena de relaciones comprobadas en el grafo que conecta la enfermedad con el factor candidato. Estos *metapaths* actúan como justificaciones lógicas de la predicción: indican qué rutas conocidas en el conocimiento biomédico sustentan la asociación propuesta entre la enfermedad y el factor de riesgo. Así, la metodología asegura no sólo generar rankings de candidatos, sino también proveer rutas interpretables que fundamentan cada predicción sin exponer directamente fragmentos de código.

5.2 Resultados cuantitativos

Los resultados cuantitativos se calcularon sobre el conjunto de test usando las métricas estándar de predicción de enlaces . El modelo obtuvo los siguientes valores:

- **Mean Reciprocal Rank (MRR)** = 0.165.
- **Hits@1** = 0.05.
- **Hits@3** = 0.12.
- **Hits@10** = 0.29.

Estas métricas permiten evaluar la capacidad del modelo para ubicar correctamente las entidades objetivo (factores de riesgo) en los primeros puestos del ranking generado para cada enfermedad. En particular, el Hits@k indica la proporción de veces en las que la entidad correcta (es decir, la verdadera) aparece entre los k primeros elementos del ranking. Por ejemplo, un Hits@10 de 0.29 implica que en el 29 % de los casos, el modelo ha sido capaz de posicionar correctamente la entidad relevante entre las diez primeras posiciones, lo que representa una tasa de recuperación razonable en grafos de conocimiento de gran escala.

Complementariamente, la métrica Mean Reciprocal Rank (MRR) ofrece una visión más precisa de la calidad del ranking completo. Esta métrica calcula el inverso de la posición en la que aparece la primera respuesta correcta para cada predicción, y luego promedia ese valor en todos los casos. Es decir, si la entidad correcta está en la primera posición, se asigna un valor de 1; si está en la segunda, 1/2; en la tercera, 1/3, y así sucesivamente. Un MRR de 0.17 indica que, en promedio, la entidad correcta aparece alrededor de la sexta posición del ranking, lo cual es una señal de que el modelo tiende a asignar puntuaciones altas a las entidades más relevantes, aunque no siempre las ubique en primer lugar.

Aunque estas cifras pudieran parecer relativamente modestas desde una perspectiva superficial, deben interpretarse en el contexto del problema abordado. El modelo fue entrenado sobre un conjunto extenso de más de 600.000 tripletas, lo cual implica una gran cantidad de combinaciones posibles de entidades y relaciones. Este elevado número de tripletas hace que el espacio de búsqueda sea inmenso: si se seleccionaran entidades candidatas de forma puramente aleatoria, la probabilidad de acertar una entidad correcta en el top-10 sería extremadamente baja. El grafo contiene aproximadamente 340.000 entidades únicas, por lo que la probabilidad de que una entidad correcta esté entre las 10 primeras en un ranking aleatorio sería del orden de 10 / 340.000 = 0.00003 (0.003 %), lo cual contrasta notablemente con el 29 % logrado por el modelo. Es decir, el rendimiento observado es más de diez mil veces superior al azar.

Además, la evaluación se llevó a cabo en un entorno computacional limitado, utilizando una GPU NVIDIA GeForce MX450 (2 GB VRAM) junto con un

procesador Intel Core i7-1165G7 y 16 GB de RAM, lo que restringe significativamente la complejidad del modelo y la posibilidad de realizar optimizaciones más profundas. A pesar de estas limitaciones, el sistema logró resultados coherentes con los obtenidos en trabajos previos en tareas de similar dificultad, como se discutirá más adelante. En suma, los valores obtenidos reflejan una capacidad de generalización sólida por parte del modelo híbrido, especialmente si se considera el tamaño y la densidad del grafo, así como la dificultad inherente del dominio biomédico.

En comparación con proyectos recientes de predicción de enlaces en grafos de conocimiento, nuestro enfoque híbrido exhibe un rendimiento consistente con las exigencias de escalabilidad y complejidad del dominio biomédico. En estudios sobre el benchmark YAGO3-10, métodos de factorización clásicos como DistMult y ComplEx informan MRR de aproximadamente 0,34–0,36 con Hits@10 de 0,54–0,55 [36]. Modelos basados en convoluciones, como ConvE, mejoran estos valores hasta MRR ≈0,40 y Hits@10 ≈0,62, mientras que enfoques de rotación en espacio complejo como RotatE alcanzan MRR ≈0,50 y Hits@10 ≈0,67. Variantes más recientes basadas en redes hiper-convolucionales (HypER) reportan MRR ≈0,465 y Hits@10 ≈0,522 [37], y modelos que combinan convolución y espacio complejo como ConEx llegan a MRR ≈0,553 y Hits@10 ≈0,696. Un análisis más realista de Sun et al. (2020) advierte que resultados de ConvE pueden reducirse a MRR ≈0,350 cuando se corrigen sesgos en YAGO3-10 [38]. Estos resultados se obtienen en un dominio de aproximadamente 123 000 entidades y más de un millón de tripletas, lo que contrasta con el entorno del proyecto llevado a cabo en esta memoria, que emplea un grafo bidireccional con más de 340 mil entidades y 782 mil tripletas, en el cual se alcanzó un MRR = 0,165 y Hits@10 = 0,29, con Hits@3 = 0,12 e Hits@1 = 0,05. Para ponerlo en perspectiva, en el benchmark FB15k-237 (≈15 000 entidades) ConvE consigue MRR ≈0,335 y Hits@10 ≈0,501, y RotatE logra MRR ≈0,338 y Hits@10 ≈0,533. Por lo tanto, aunque las cifras absolutas quedan algo lejos de los modelos calibrados para grafos de menor tamaño, se refleja un desempeño sólido y coherente con la complejidad y heterogeneidad del dominio, al combinar la generalización de la GCN con el conocimiento simbólico aportado por AnyBURL.

5.3 Resultados cualitativos

La evaluación cualitativa se centra en analizar e interpretar los metapaths generados, que representan rutas lógicas interpretables en el grafo biomédico. Estos metapaths permiten explicar cómo el modelo conecta enfermedades específicas, en este caso distintos tipos de cáncer de pulmón, con otros tipos de entidades dentro del grafo.

Las cuatro figuras adjuntas muestran claramente cuáles son las siguientes clases más relacionadas tras el primer salto desde las entidades directamente vinculadas a *Disease*. Estas figuras representan visualmente la frecuencia y relevancia de conexiones:

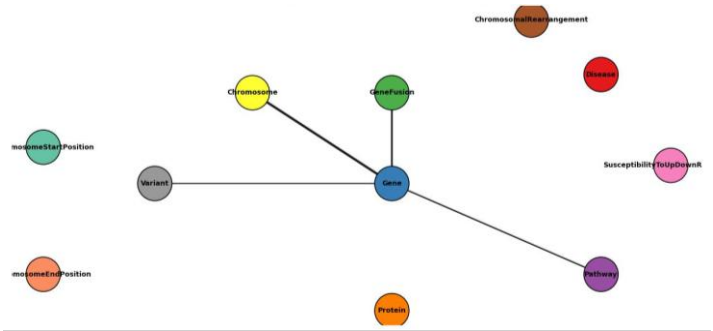


Figura 9: Conexiones clase Gene

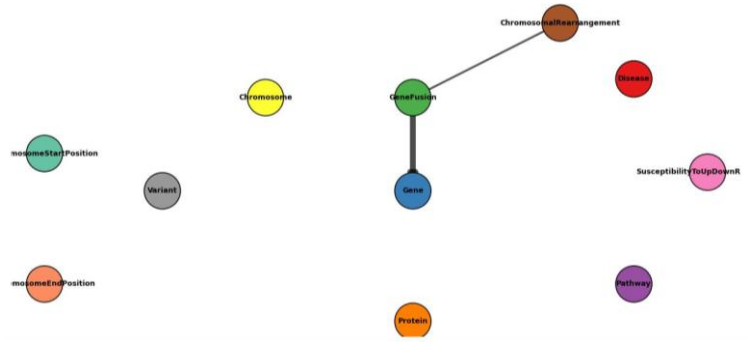


Figura 10: Conexiones GeneFusion

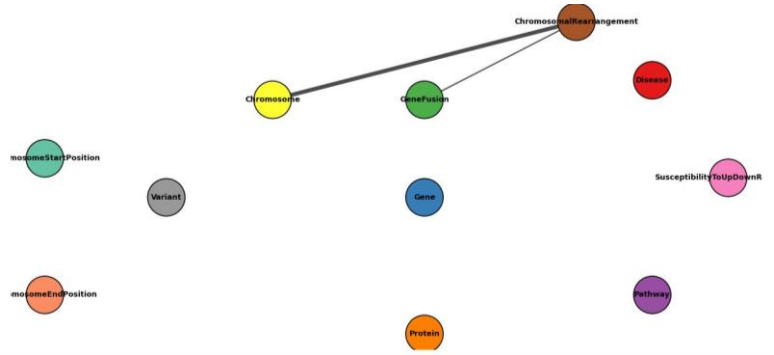


Figura 11: Conexiones clase Rearrangement

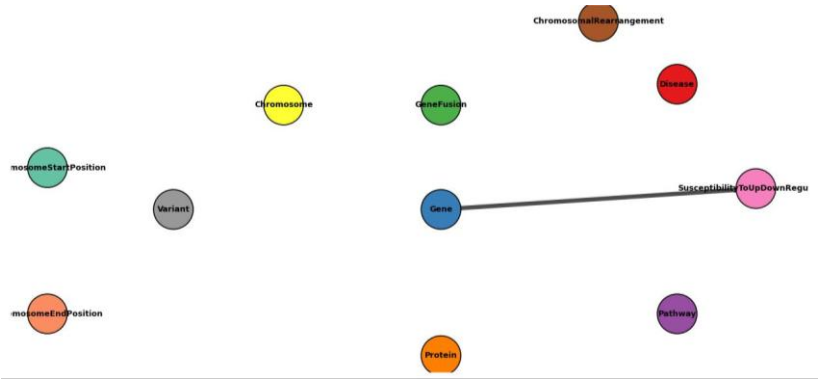


Figura 12: Conexiones clase Susceptibility

Al analizar conjuntamente las cuatro imágenes proporcionadas (Figura 9, Figura 10, Figura 11 y Figura 12), se concluye que las clases *Gene* y *GeneFusion* están profundamente interrelacionadas con las clases *Chromosome* y *ChromosomalRearrangement*. Esta fuerte asociación parece indicar que los eventos moleculares fundamentales en el cáncer de pulmón implican frecuentemente interacciones genéticas directas, fusiones génicas derivadas de reordenamientos cromosómicos, y alteraciones en la regulación genética.

Además, se hizo un conteo de las entidades que más aparecieron en estos metapaths, y se identificaron las siguientes entidades:

- **LUCIA03820dbb-53cb-b201-1149-7cde175b47ab**: fusión génica tipo "TNRC6B::STK11". Esta fusión implica a STK11, un conocido gen supresor tumoral estrechamente vinculado con cánceres pulmonares, especialmente con el cáncer de pulmón de células no pequeñas (NSCLC). STK11 es ampliamente estudiado debido a su rol crucial en la regulación del crecimiento celular y metabolismo energético.

- **ncbigene:7158 (TP53BP1)**: "Tumor protein p53 binding protein 1", proteína que interacciona con el gen TP53, clave en regulación del ciclo celular y apoptosis. Numerosos estudios asocian alteraciones de TP53BP1 con inestabilidad genómica y cáncer pulmonar, respaldando la relevancia del resultado obtenido (National Institutes of Health - PubMed) [3].

- **LUCIA9b54a449-58c3-d956-86cf-a3d379ad63a2**: fusión génica tipo "CTTN::MRGPRF". Aunque menos estudiada específicamente en cáncer de pulmón, esta fusión podría sugerir una vía molecular novedosa a explorar debido a su implicación en procesos de señalización celular y migración celular, características clave en la progresión tumoral.

5.4 Discusión de resultados

La evaluación cuantitativa y cualitativa del modelo muestra un desempeño robusto considerando la complejidad y tamaño del grafo biomédico. Los resultados cuantitativos, aunque inferiores a benchmarks más pequeños como YAGO3-10 o FB15k-237, son razonables dada la alta heterogeneidad y gran escala del grafo utilizado. La capacidad del modelo para colocar entidades correctas entre los primeros diez lugares en el 29 % de los casos, demostrando su potencial en descubrimiento biomédico.

Los resultados cualitativos proporcionan perspectivas valiosas. Los metapaths destacan conexiones coherentes y biológicamente relevantes, validando la capacidad del modelo para generar explicaciones interpretables y significativas.

Las fusiones génicas "TNRC6B::STK11" y "CTTN::MRGPRF" emergen como potenciales candidatos novedosos o poco explorados en la literatura científica, abriendo líneas futuras de investigación biomédica y experimental. Asimismo, la identificación de TP53BP1 fortalece la validez biológica del modelo, dado el amplio reconocimiento de este gen en estudios oncológicos pulmonares.

En conclusión, el modelo híbrido integra eficazmente el razonamiento estadístico de la red neuronal gráfica con la lógica simbólica de AnyBURL, generando predicciones sólidas acompañadas por rutas explicativas claras y biológicamente coherentes. Estos hallazgos demuestran su utilidad potencial para investigaciones biomédicas, destacando tanto factores bien establecidos como nuevos candidatos prometedores en la investigación del cáncer pulmonar.

6 Conclusiones y trabajo futuro

6.1 Conclusiones

Este proyecto ha abordado el objetivo principal de desarrollar un modelo basado en Graph Neural Networks (GNN) para analizar grafos de conocimiento del cáncer de pulmón y extraer información relevante sobre posibles factores de riesgo. El uso de GNN ha demostrado ser eficaz en la identificación y predicción de relaciones complejas dentro del dominio biomédico, destacando su potencial para aportar valiosa información en la prevención y tratamiento del cáncer de pulmón.

En relación con los objetivos específicos planteados:

1. Se ha realizado un estudio exhaustivo sobre Graph Neural Networks, grafos de conocimiento, y la terminología biomédica específica relativa al cáncer de pulmón. Esto ha permitido seleccionar adecuadamente las tecnologías más apropiadas y entender en profundidad las estructuras de datos implicadas.

2. El proyecto ha alcanzado un alto grado de familiarización con los datos del grafo, reconociendo claramente las entidades biomédicas implicadas (genes, proteínas, fusiones génicas, variantes, entre otras) y sus relaciones. Esta comprensión ha resultado crucial para el correcto diseño del modelo de aprendizaje.

3. Se han implementado estrategias robustas para la transformación del grafo en formatos adecuados para su integración efectiva con modelos GNN. Dichas estrategias han garantizado la coherencia semántica y la calidad de los datos introducidos al modelo.

4. El entrenamiento y la optimización del modelo híbrido (combinación de técnicas simbólicas y neuronales mediante AnyBURL y PyTorch Geometric) han demostrado ser efectivos, alcanzando resultados cuantitativos y cualitativos satisfactorios en la tarea de descubrimiento de información sobre factores de riesgo.

5. La evaluación exhaustiva del modelo ha permitido validar su eficacia en la identificación de factores de riesgo potencialmente significativos. Se han obtenido hallazgos coherentes con el conocimiento biomédico establecido, demostrando la capacidad del modelo para contribuir con información novedosa a la investigación del cáncer de pulmón.

A título personal, este proyecto ha supuesto un reto interesante y enriquecedor que ha ampliado considerablemente mi conocimiento sobre aplicaciones prácticas de inteligencia artificial en el ámbito biomédico. No solo he aprendido profundamente sobre técnicas avanzadas de inteligencia artificial, como llevo haciendo durante gran parte de la carrera, sino también acerca del mundo biomédico, comprendiendo cómo la intersección inteligente entre estos dos ámbitos puede producir importantes avances y beneficios reales para la sociedad. Este proceso de aprendizaje continuo durante el desarrollo del trabajo ha resultado enormemente gratificante y formativo.

6.2 Trabajo futuro

El desarrollo llevado a cabo en este proyecto presenta diversas posibilidades de ampliación y mejoras, así como la oportunidad de extensión a otras enfermedades o dominios clínicos.

Ampliaciones o mejoras del modelo

Una posible mejora para trabajos futuros es el desarrollo de un modelo significativamente más complejo y profundo, capaz de explotar plenamente las capacidades de representación de las Graph Neural Networks. En este proyecto, ciertas limitaciones computacionales han restringido la complejidad del modelo. Con mayores recursos computacionales sería posible explorar arquitecturas más sofisticadas, capas adicionales, o técnicas avanzadas de regularización y optimización que podrían aumentar considerablemente la precisión y capacidad predictiva del modelo.

Otra vía relevante para mejorar el modelo sería la incorporación de clases adicionales presentes en el grafo original que no fueron consideradas en el subgrafo extraído. Incluir el resto de entidades podría permitir al modelo capturar una gama más amplia de relaciones y fenómenos (seleccionándolas de manera inteligente, si no podría añadir ruido al modelo), enriqueciendo así su capacidad para descubrir patrones significativos sobre factores de riesgo.

Asimismo, una mejora práctica muy valiosa sería el desarrollo de una interfaz gráfica de usuario (GUI), que permita la interacción directa con el paciente o usuario final. A través de esta interfaz, los pacientes podrían ingresar datos clínicos y personales, facilitando que el modelo evalúe internamente su perfil y prediga su nivel de riesgo de padecer cáncer de pulmón. Esta herramienta tendría un impacto significativo en el ámbito clínico, proporcionando una manera amigable e intuitiva de interpretar resultados complejos generados por el modelo.

Extensión a otras enfermedades o datos clínicos

La metodología desarrollada en este proyecto es generalizable y podría ser aplicada en estudios sobre otras enfermedades complejas, tales como diferentes tipos de cáncer, enfermedades autoinmunes o trastornos neurológicos. Este enfoque podría ayudar a descubrir patrones desconocidos y facilitar nuevas hipótesis clínicas, acelerando potencialmente la investigación y mejorando la prevención y tratamiento de múltiples enfermedades.

Además, sería interesante extender el modelo para integrar otras fuentes de datos clínicos, tales como registros electrónicos de salud, datos genómicos o farmacológicos, ampliando así significativamente la profundidad y aplicabilidad de los resultados obtenidos. Esta extensión facilitaría el desarrollo de herramientas integradas de apoyo a la decisión clínica, mejorando significativamente su impacto práctico en el ámbito médico.

7 Análisis de Impacto

En este apartado se examinan las implicaciones globales del proyecto en diferentes ámbitos. Se describen las posibles aplicaciones en la investigación biomédica (7.1) y los efectos esperados en los planos personal, empresarial, social, económico, medioambiental y cultural (7.2). El análisis considera tanto los beneficios potenciales como los riesgos o impactos adversos, así como la alineación con los Objetivos de Desarrollo Sostenible (ODS) relevantes (ODS 3, 9 y 10).

7.1 Aplicaciones Potenciales en la Investigación Biomédica

La aplicación de redes neuronales de grafos (GNN) al grafo de conocimiento sobre cáncer de pulmón permite integrar y analizar relaciones complejas entre factores de riesgo y distintos tipos de cáncer de pulmón. Las GNN están diseñadas para procesar datos estructurados en forma de grafo, capturando tanto la topología de la red como atributos de los nodos mediante transformaciones no lineales. Esta capacidad facilita la detección de patrones ocultos y asociaciones de alto nivel en el grafo, lo que puede conducir a la identificación de nuevos biomarcadores o variables de riesgo no evidentes con métodos convencionales.

Por ejemplo, gracias a la representación por grafos se pueden generar hipótesis automáticas sobre asociaciones entre genes y el cáncer de pulmón. El modelo puede priorizar factores de riesgo candidatos al identificar nodos muy conectados o subredes relevantes, acelerando así el descubrimiento de relaciones causales entre biología molecular y enfermedad. De esta forma se apoyaría la búsqueda de tratamientos personalizados basados en el perfil genético del paciente. En general, el uso de aprendizaje de representación en grafos ha demostrado mejorar el estado del arte en tareas biomédicas: ha permitido identificar variantes genéticas implicadas en rasgos complejos, caracterizar comportamientos celulares y mejorar diagnósticos clínicos. Estas capacidades son especialmente valiosas en la investigación biomédica, donde la combinación de datos genómicos, epigenómicos y clínicos puede enriquecer los análisis.

Asimismo, en el ámbito farmacéutico el modelo podría revelar nuevas dianas terapéuticas. Al procesar grandes volúmenes de datos heterogéneos, las GNN pueden descubrir patrones de interacción moleculas o rutas de señalización asociadas con el cáncer de pulmón. Esto favorece el diseño de medicamentos más seguros y eficaces, alineando el proyecto con las metas de salud globales. En concreto, estas herramientas pueden reducir el tiempo de desarrollo de fármacos al sugerir compuestos candidatos o combinaciones de terapias basadas en la estructura del grafo. Además, la capacidad de generalizar a casos no vistos permite aplicar los resultados a poblaciones diversas, apoyando la medicina de precisión.

En conclusión, se esperan aplicaciones como la detección temprana de factores de riesgo, la priorización de biomarcadores y la ayuda en el desarrollo de tratamientos, contribuyendo a un avance en el conocimiento de la enfermedad. Estos beneficios potencian el impacto positivo en la salud y la innovación científica, tal como evidencian revisiones recientes sobre la eficacia de las GNN en biomedicina.

7.2 Impacto Social, Medioambiental y Cultural

El proyecto puede generar impactos relevantes más allá del ámbito estrictamente técnico. A continuación se analizan los efectos esperados en distintos ámbitos, así como su alineación con los ODS aplicables.

- **Ámbito personal y sanitario (ODS 3)**: A nivel individual, se anticipa una mejora en la prevención y el diagnóstico. La IA médica, como en este proyecto, puede aumentar la cobertura y calidad de la atención sanitaria permitiendo diagnósticos más tempranos y precisos. Esto contribuye directamente al ODS 3 (Salud y bienestar), pues puede mejorar el acceso a servicios de salud y reducir disparidades sanitarias [39]. No obstante, también existen riesgos personales: las predicciones de un modelo imperfecto podrían generar falsos positivos o negativos, provocando ansiedad en pacientes o diagnósticos erróneos. Asimismo, se debe gestionar cuidadosamente la privacidad de los datos de salud: el uso de información biomédica sensible exige garantizar confidencialidad y consentimiento informado.

- **Ámbito empresarial e innovación (ODS 9)**: Tecnologías avanzadas como las GNN impulsan la innovación en el sector salud. El desarrollo de este modelo abre oportunidades de negocio en software médico y análisis de datos, estimulando la industria de biotecnología y salud digital. Al optimizar procesos de I+D (por ejemplo, acortar ciclos de investigación traslacional), puede aumentar la eficiencia económica y reducir costes en el largo plazo. Estas mejoras contribuyen al ODS 9 (Industria, innovación e infraestructura), pues promueven la infraestructura tecnológica y la investigación propia. Sin embargo, existe el riesgo de que solo grandes entidades con recursos masivos lideren el uso de IA, lo que podría marginalizar a empresas más pequeñas. Para evitarlo, se han privilegiado herramientas de código abierto que facilitan la adopción amplia del proyecto.

- **Ámbito social y equidad (ODS 10)**: A nivel colectivo, el proyecto mejora el conocimiento público sobre el cáncer de pulmón al integrar datos epidemiológicos y clínicos. Esto puede empoderar a comunidades y sistemas de salud para identificar factores de riesgo locales, generando campañas preventivas basadas en evidencia. La IA, en escenarios ideales,

tiene potencial para reducir desigualdades en salud (ODS 10), al hacer posible diagnósticos avanzados en áreas remotas o con pocos recursos. No obstante, también puede exacerbar brechas existentes si no se implementa equitativamente. Por ejemplo, los modelos deben entrenarse con datos representativos; de lo contrario, podrían sesgarse contra minorías étnicas o grupos desatendidos, empeorando desigualdades.

- **Ámbito económico**: Desde una perspectiva macroeconómica, el uso de IA en salud puede derivar en ahorros significativos a través de la prevención y eficiencia clínica. Detectar riesgos tempranos reduce la carga hospitalaria y puede disminuir gastos en tratamientos avanzados. Además, la generación de datos y hallazgos favorece la economía del conocimiento, creando activos que empresas farmacéuticas o tecnológicas pueden aprovechar. Por otro lado, la implementación del proyecto implica inversión en infraestructura de datos y capacitación de personal, lo que representa un coste inicial. Sin embargo, el balance neto suele ser positivo si se traduce en mejoras sanitarias y productivas. En caso de que los beneficios económicos se concentren solo en ciertas organizaciones, existiría un efecto adverso en la redistribución de recursos; por ello es importante fomentar políticas de acceso y formación para diversos actores.

- **Ámbito medioambiental**: La implementación de modelos de IA conlleva un consumo energético notable, asociado al entrenamiento continuo de redes neuronales y al mantenimiento de centros de datos. Estudios recientes advierten que el desarrollo acelerado de IA en salud genera costes medioambientales significativos, especialmente por las emisiones de carbono de la infraestructura tecnológica [41]. Este impacto no suele ser asumido directamente por los usuarios de la tecnología, sino que repercute en la sociedad en general. Para mitigarlo, el proyecto ha considerado decisiones como utilizar recursos computacionales eficientes y formatos de datos que no requieran procesamientos redundantes. Además, fomentar la sensibilización sobre la huella de carbono de la IA puede encaminar a buscar fuentes de energía renovable para los centros de cómputo.

- **Ámbito cultural y ético**: El proyecto promueve una cultura de investigación abierta y colaborativa. Se ha optado por herramientas de código abierto (bibliotecas comunes de aprendizaje automático) y estándares de datos interoperables (CSV, JSON, formatos RDF semánticos), de modo que otros investigadores puedan acceder, reproducir y ampliar los resultados. Esto refuerza valores de transparencia y democratización del conocimiento, alineados con la ciencia abierta. Al mismo tiempo, introduce desafíos culturales: el uso de IA en medicina requiere la confianza de profesionales y pacientes. Para ello es esencial documentar bien los algoritmos y exponer de manera accesible las ventajas y limitaciones del modelo. También se deben tener

en cuenta los aspectos éticos, como la responsabilidad en caso de error del sistema y el equilibrio entre automatización y juicio médico humano.

Bibliografía

[1] World Health Organization, “Lung cancer,” WHO Fact Sheet. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/lung-cancer. Accessed: Mar. 3, 2025.

[2] F.-T. Meng, J.-R. Jhuang, Y.-T. Peng, C.-J. Chiang, Y.-W. Yang, C.-Y. Huang et al., “Predicting lung cancer survival to the future: population-based cancer survival modeling study,” JMIR Public Health Surveill., vol. 10, p. e46737, May 2024. [Online]. Available: https://publichealth.jmir.org/2024/1/e46737. Accessed: Mar. 15, 2025.

[3] H. Li, Y. Yang, C. Tang, B. He, Q. Liu, F. Li et al., “CGMega: explainable graph neural network framework with attention mechanisms for cancer gene module dissection,” Nat. Commun., vol. 15, no. 1, p. 5997, Jul. 17, 2024, doi: 10.1038/s41467-024-50426-6. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC11252405/. Accessed: Mar. 28, 2025.

[4] M. Contreiras Silva, P. Eugénio, D. Faria, and C. Pesquita, “Ontologies and knowledge graphs in oncology research,” Cancers (Basel), vol. 14, no. 8, p. 1906, Apr. 10, 2022, doi: 10.3390/cancers14081906. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC9029532/. Accessed: Apr. 4, 2025.

[5] Pan American Health Organization, “Objetivos de Desarrollo Sostenible,” PAHO. [Online]. Available: https://www.paho.org/es/temas/objetivos-desarrollo-sostenible. Accessed: Apr. 10, 2025.

[6] B. Keogh, “Why we must act now to shape a lung cancer revolution,” World Economic Forum, Aug. 11, 2022. [Online]. Available: https://www.weforum.org/stories/2022/08/why-we-must-act-now-to-shape-a-lung-cancer-revolution/. Accessed: Apr. 18, 2025.

[7] American Lung Association, “New Report: Lung cancer survival rate improves, but gaps in biomarker testing and lack of screening hinder progress,” Press Release, Chicago, IL, Nov. 19, 2024. [Online]. Available: https://www.lung.org/media/press-releases/state-of-lung-cancer-2024. Accessed: Apr. 27, 2025.

[8] D. N. Nicholson and C. S. Greene, “Constructing knowledge graphs and their biomedical applications,” Comput. Struct. Biotechnol. J., vol. 18, pp. 1414–1428, Jun. 2, 2020, doi: 10.1016/j.csbj.2020.05.017. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC7327409/. Accessed: May 2, 2025.

[9] IBM, “Knowledge graph,” IBM Think. [Online]. Available: https://www.ibm.com/es-es/think/topics/knowledge-graph. Accessed: May 8, 2025.

[10] D. N. Nicholson and C. S. Greene, “Constructing knowledge graphs and their biomedical applications,” Comput. Struct. Biotechnol. J., vol. 18, pp. 1414–1428, Jun. 2, 2020, doi: 10.1016/j.csbj.2020.05.017. [Online]. Available:

https://pmc.ncbi.nlm.nih.gov/articles/PMC7327409/. Accessed: May 14, 2025.
[11] W. L. Hamilton, “Graph Neural Networks,” in Graph Representation Learning, San Rafael, CA, USA: Morgan & Claypool, 2020. [Online]. Available: https://www.cs.mcgill.ca/~wlh/grl_book/files/GRL_Book-Chapter_5-GNNs.pdf. Accessed: May 6, 2025.
[12] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, and M. Sun, “Graph Neural Networks: A Review of Methods and Applications,” arXiv:1812.08434, 2018. [Online]. Available: https://arxiv.org/pdf/1812.08434. Accessed: May 11, 2025.
[13] L. Wu, P. Cui, J. Pei, and L. Zhao, Eds., Graph Neural Networks: Foundations, Frontiers, and Applications, Singapore: Springer Nature Singapore, 2022, doi: 10.1007/978-981-16-6054-2.
[14] F. Aisopos and G. Paliouras, “Comparing methods for drug–gene interaction prediction on the biomedical literature knowledge graph: performance versus explainability,” BMC Bioinformatics, vol. 24, no. 1, p. 272, 2023, doi: 10.1186/s12859-023-00876-4.
[15] D. Bang, S. Lim, S. Lee, and S. Kim, “Biomedical knowledge graph learning for drug repurposing by extending guilt-by-association to multiple layers,” Nat. Commun., vol. 14, no. 1, p. 3570, Dec. 2023, doi: 10.1038/s41467-023-39301-y. [Online]. Available: https://doi.org/10.1038/s41467-023-39301-y. Accessed: May 16, 2025.
[16] M. Zitnik, M. Agrawal, and J. Leskovec, “Modeling polypharmacy side effects with graph convolutional networks,” Bioinformatics, vol. 34, no. 13, pp. i457–i466, Jul. 2018, doi: 10.1093/bioinformatics/bty294.
[17] Z. Zeng, Q. Cheng, and Y. Si, “Logical Rule-Based Knowledge Graph Reasoning: A Comprehensive Survey,” Mathematics, vol. 11, no. 21, art. no. 4486, Oct. 30, 2023, doi: 10.3390/math11214486. [Online]. Available: https://www.mdpi.com/2227-7390/11/21/4486. Accessed: May 22, 2025.
[18] A. Jiménez, M. J. Merino, J. Parras, and S. Zazo, “Explainable drug repurposing via path based knowledge graph completion,” Sci. Rep., vol. 14, art. no. 16587, Jul. 18, 2024, doi: 10.1038/s41598-024-67163-x. [Online]. Available: https://www.nature.com/articles/s41598-024-67163-x. Accessed: May 29, 2025.
[19] A. Gogleva, D. Polychronopoulos, M. Pfeifer, V. Poroshin, M. Ughetto, M. J. Martin, H. Thorpe, A. Bornot, P. D. Smith, B. Sidders, J. R. Dry, M. Ahdesmäki, U. McDermott, E. Papa, and K. C. Bulusu, “Knowledge graph-based recommendation framework identifies drivers of resistance in EGFR mutant non-small cell lung cancer,” Nat. Commun., vol. 13, no. 1, p. 1667, Mar. 29, 2022, doi: 10.1038/s41467-022-29292-7. [Online]. Available: https://www.nature.com/articles/s41467-022-29292-7. Accessed: May 3, 2025.
[20] C. Chen, S. R. Haupert, L. Zimmermann, X. Shi, L. G. Fritsche, and B. Mukherjee, “Global prevalence of post-coronavirus disease 2019 (COVID-19) condition or long COVID: a meta-analysis and systematic

review," J. Infect. Dis., vol. 226, no. 9, pp. 1593–1607, Nov. 1, 2022, doi: 10.1093/infdis/jiac136.
[21] M. Vaida, J. Wu, E. Himdiat, J.-F. Haince, R. A. Bux, G. Huang, P. S. Tappia, B. Ramjiawan, and W. R. Ford, "M-GNN: A graph neural network framework for lung cancer detection using metabolomics and heterogeneous graph modeling," Int. J. Mol. Sci., vol. 26, no. 10, art. no. 4655, May 13, 2025, doi: 10.3390/ijms26104655. [Online]. Available: https://www.mdpi.com/1422-0067/26/10/4655. Accessed: May 30, 2025.
[22] Authors Unknown, "Title Unknown," in Proc. 12th Language Resources and Evaluation Conference (LREC 2024), Marseille, France, 2024. [Online]. Available: https://aclanthology.org/2024.lrec-main.792.pdf. Accessed: May 7, 2025.
[23] Authors Unknown, "Title Unknown," OpenReview, 2024. [Online]. Available: https://openreview.net/pdf?id=DEsIX_D_vR. Accessed: May 13, 2025.
[24] Python Software Foundation, "Python," [Online]. Available: https://www.python.org/. Accessed: May 1, 2025.
[25] B. Khemani, S. Patil, K. Kotecha, and S. Tanwar, "A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions," J. Big Data, vol. 11, no. 1, art. no. 18, Jan. 16, 2024. [Online]. Available: https://doi.org/10.1186/s40537-023-00876-4. Accessed: May 21, 2025. SpringerOpen
[26] D. Hellmann, The Python Standard Library by Example. Boston, MA, USA: Addison-Wesley, 2007. [Online]. Available: https://doughellmann.com/books/the-python-standard-library-by-example/. Accessed: May 4, 2025.
[27] A. Sweigart, Automate the Boring Stuff with Python. [2nd ed.]. Glen Park, CA, USA: No Starch Press, 2019. Chapter 9. [Online]. Available: https://automatetheboringstuff.com/2e/chapter9/. Accessed: May 9, 2025.
[28] W. McKinney, Python for Data Analysis: Data Wrangling with pandas, NumPy, and Jupyter, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2018.
[29] "PyTorch," [Online]. Available: https://pytorch.org/. Accessed: May 17, 2025.
[30] "PyTorch Geometric documentation," [Online]. Available: https://pytorch-geometric.readthedocs.io/en/latest/. Accessed: May 23, 2025.
[31] "NetworkX," [Online]. Available: https://networkx.org/. Accessed: May 24, 2025.
[32] C. Meilicke, M. W. Chekol, D. Ruffinelli, and H. Stuckenschmidt, "An introduction to AnyBURL," in KI 2019: Advances in Artificial Intelligence: 42nd German Conference on AI, Kassel, Germany, Sept. 23–26, 2019, Proc., LNCS 11704, Springer, 2019, pp. 244–248.
[33] E. Prud'hommeaux and A. Seaborne, "SPARQL 1.1 Query Language," W3C Recommendation, Mar. 21, 2013. [Online]. Available: https://www.w3.org/TR/sparql11-query/. Accessed: May 18, 2025.

[34] P. Chandak, K. Huang, and M. Zitnik, “Building a knowledge graph to enable precision medicine,” Sci. Data, vol. 10, art. no. 67, Feb. 2, 2023, doi: 10.1038/s41597-023-01960-3. [Online]. Available: https://doi.org/10.1038/s41597-023-01960-3. Accessed: May 26, 2025. Nature

[35] M. Qu, J. Chen, L.-P. Xhonneux, Y. Bengio, and J. Tang, “RNNLogic: Learning logic rules for reasoning on knowledge graphs,” arXiv:2010.04029 [cs.AI], Jul. 16, 2021. [Online]. Available: https://arxiv.org/abs/2010.04029. Accessed: May 10, 2025.

[36] “Convolutional-Complex-Knowledge-Graph-Embeddings,” GitHub repository, 2020. [Online]. Available: https://github.com/dice-group/Convolutional-Complex-Knowledge-Graph-Embeddings. Accessed: May 2, 2025.

[37] I. Balažević, C. Allen, and T. M. Hospedales, “Hypernetwork Knowledge Graph Embeddings,” arXiv:1808.07018 [cs.LG], Jul. 15, 2019. [Online]. Available: https://arxiv.org/abs/1808.07018. Accessed: May 5, 2025.

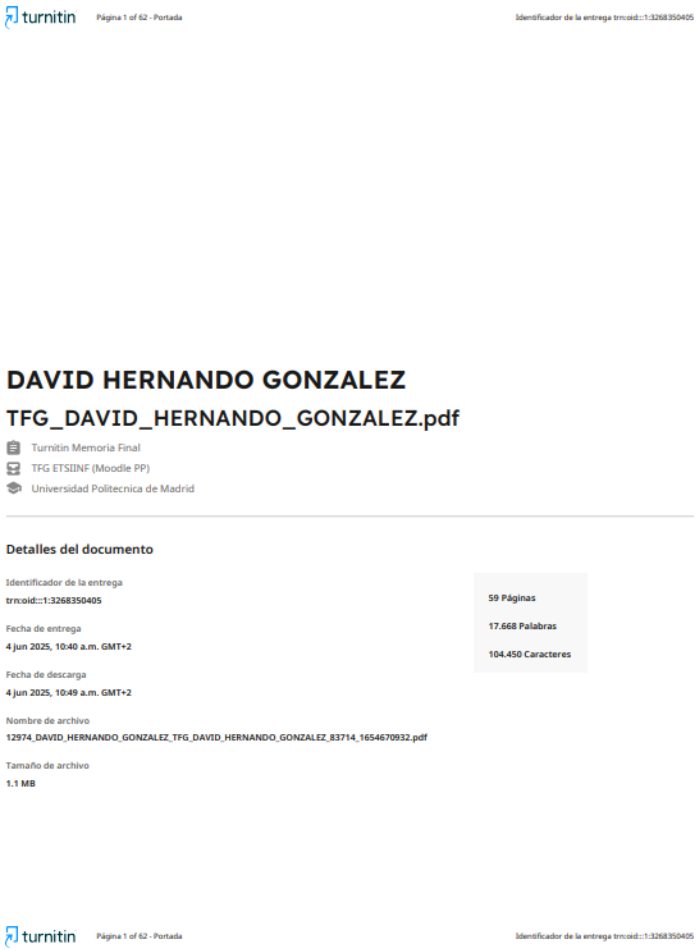
[38] F. Akrami, M. S. Saeef, Q. Zhang, W. Hu, and C. Li, “Realistic Re-evaluation of Knowledge Graph Completion Methods: An Experimental Study,” arXiv:2003.08001 [cs.AI], Mar. 18, 2020. [Online]. Available: https://arxiv.org/abs/2003.08001. Accessed: May 12, 2025.

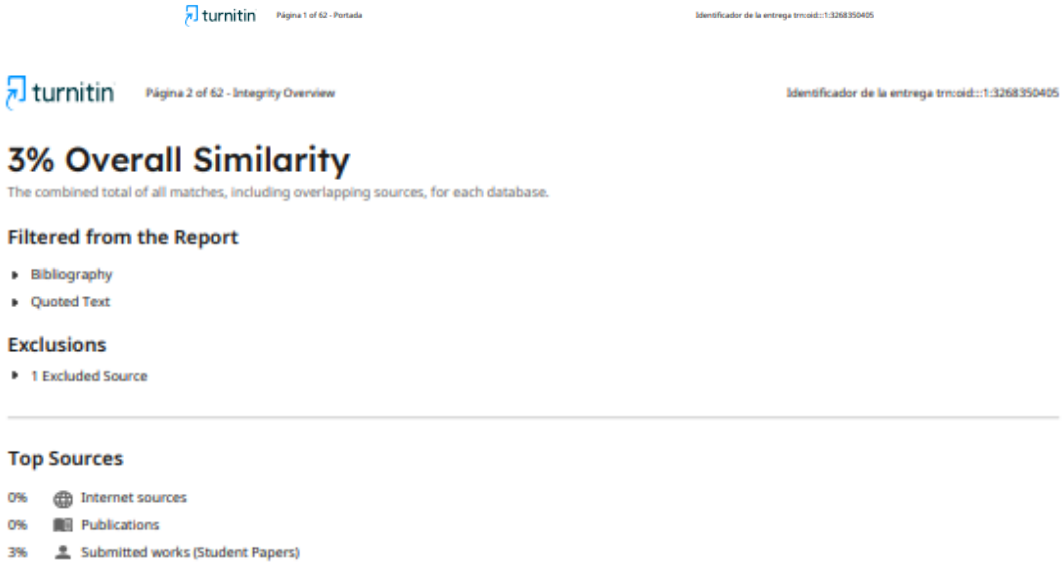
[39] A. Bohr and K. Memarzadeh, “The Rise of Artificial Intelligence in Healthcare Applications,” in Artificial Intelligence in Healthcare, A. Bohr and K. Memarzadeh, Eds., Academic Press, 2020, pp. 25–60, doi: 10.1016/B978-0-12-818438-7.00002-2. Accessed: May 14, 2025

[40] D. A. Moreno Perdomo, P. Tejera Nevado, L. Prieto-Santamaria, G. Vigueras, A. J. Diaz-Honrubia, and A. Rodriguez-Gonzalez, "Lung-CABO: Lung Cancer Concepts Association Biological Ontology," in *Proc. 2025 IEEE 35th Int. Symp. Comput.-Based Med. Syst. (CBMS)*, June 2025.

[41] R. Selvan, N. Bhagwat, L. F. W. Anthony, B. Kanding y E. B. Dam, "Carbon Footprint of Selecting and Training Deep Learning Models for Medical Image Analysis," arXiv preprint arXiv:2203.02202, 2022.

Anexo


turnitin
Página 1 of 62 - Portada
Identificador de la entrega trn:oid:::1:3268350405
DAVID HERNANDO GONZALEZ
TFG_DAVID_HERNANDO_GONZALEZ.pdf
Turnitin Memoria Final
TFG ETSIINF (Moodle PP)
Universidad Politecnica de Madrid
Detalles del documento
Identificador de la entrega
trn:oid:::1:3268350405
Fecha de entrega
4 jun 2025, 10:40 a.m. GMT+2
Fecha de descarga
4 jun 2025, 10:49 a.m. GMT+2
Nombre de archivo
12974_DAVID_HERNANDO_GONZALEZ_TFG_DAVID_HERNANDO_GONZALEZ_83714_1654670932.pdf
Tamaño de archivo
1.1 MB
59 Páginas
17.668 Palabras
104.450 Caracteres


turnitin
Página 2 of 62 - Integrity Overview
Identificador de la entrega trn:oid:::1:3268350405
3% Overall Similarity
The combined total of all matches, including overlapping sources, for each database.
Filtered from the Report
Bibliography
Quoted Text
Exclusions
1 Excluded Source
Top Sources
0% Internet sources
0% Publications
3% Submitted works (Student Papers)

Este documento esta firmado por

	**Firmante**	CN=tfgm.fi.upm.es, OU=CCFI, O=ETS Ingenieros Informaticos - UPM, C=ES
	Fecha/Hora	Wed Jun 04 16:34:05 CEST 2025
	Emisor del Certificado	EMAILADDRESS=camanager@etsiinf.upm.es, CN=CA ETS Ingenieros Informaticos, O=ETS Ingenieros Informaticos - UPM, C=ES
	Numero de Serie	561
	Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)