Universidad de Valparaíso CHILE
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA CIVIL BIOMÉDICA

Elaboración de red de conocimiento informático que relaciona signos, procedimientos y dispositivos médicos en patologías de las garantías explícitas de salud GES.

HERMAN ALEXANDER CHINGA OLIVARES

Trabajo para optar al Título de

Ingeniero Civil Biomédico

Profesor Guía:

ALEJANDO VELÓZ BAEZA

Julio - 2022

Valparaíso – Chile

Dedicatoria

El presente trabajo se lo dedico a quienes en vida llevaran nombre de Ana del Carmen Ferreira, María Vargas Valladares y María Pardo, siguen estando en mi corazón por siempre, me regocijaron y cuidaron como niño, fueron mujeres fuertes que salieron adelante a pesar de toda la adversidad del norte de Chile y de los malos hombres.

También a mis niñes, Benjamín Ignacio, Raffaela Amaranta y Atenea Belén, han sido la motivación para llegar a la meta de esta etapa y siempre les amaré por ser mi hermano y mis sobrinas <3.

Agradecimientos

Agradezco a quien con su palabra plantó una semilla en mi corazón, a toda persona que me deseó bien y éxito. Sin duda soy vivero de valores y conocimientos, menos o más nobles o profundos que los otros, pero sí genuinamente cuidado por mi familia, profesores y amigos; quienes me educaron de todas las formas, quienes me corrigieron en mis errores y lo seguirán haciendo.

Muchas gracias a mi madre y a mi padre porque nunca me faltaron, muchas gracias a mi hermano por prepararme el camino de la vida y a mis familiares que siempre me tuvieron fe.

Resumen

En el área de la informática biomédica existen recursos y herramientas de Data science e inteligencia artificial que pueden ser aprovechadas para generar aplicaciones de investigación en la salud. Estas herramientas permiten tareas como la manipulación de grandes volúmenes de información o procesamiento de texto con lenguaje no estructurado para el reconocimiento de entidades de interés. **Objetivo**: En el presente trabajo de título se desarrolló una metodología para elaborar una red de información biomédica que relacione problemas de salud del plan AUGE con patologías, signos, procedimientos y dispositivos médicos. **Métodos**: Para lograr el objetivo, se realizó procesamiento a cientos de artículos científicos de la base de datos Pubmed Central para detectar las entidades y generar las relaciones entre estas a través del lenguaje de programación Python. **Resultados**: Finalmente, se obtuvo una red de información con 2097 nodos y 10267 lazos que fueron analizados en el software Gephi.

Palabras Clave: *knowledge graph, Natural language processing, named entity recognition, pubmed central*.

TABLA DE CONTENIDO

Elaboración de red de conocimiento informático que relaciona signos, procedimientos y dispositivos médicos en las patologías de las garantías explícitas de salud GES.

Herman Alexander Chinga Olivares
Escuela de Ingeniería Civil Biomédica
Facultad de Ingeniería, Universidad de Valparaíso, Chile

Palabras clave: *knowledge graph, Natural language processing, named entity recognition, pubmed central.*

1. Introducción

En nuestro país, la estructuración del sistema de salud actual comenzó con la promulgación de la ley de las garantías explícitas en salud (GES) el 2003, en la cual se comenzó a trabajar para que las dimensiones de la atención en salud se conformaran por cuatro garantías: el acceso, la calidad, la oportunidad y la protección financiera y que hoy en día garantiza la atención de 85 problemas de salud, cubiertas bajo el plan de acceso universal a garantías explícitas o plan AUGE. Asimismo, luego de la reforma estructural del sistema el año 2004 se elaboraron estrategias que consideraban el uso de la tecnología para obtener información, ya que a la fecha no había sistemas de información para el seguimiento de las garantías, lo que tenía como consecuencia la poca información disponible para establecer demandas o proyecciones [1] [2]. A partir de ese entonces, la estrategia de informatización de redes asistenciales (SIDRA) comenzó a elaborarse con el objetivo de aumentar la digitalización de los datos en todos los niveles de información y de atención, estableciéndose metas como el registro clínico electrónico compartido o el repositorio nacional de datos en salud, que aún se encuentran en desarrollo [3]. Hoy en día la digitalización de la información en salud está establecida, con ella se genera la comunicación entre distintas instituciones u organizaciones de la salud, se hacen estadísticas, se coordinan atenciones médicas o pagos de prestaciones, entre muchas acciones más. Sin embargo, el uso de información y datos en salud ha sido marco de regulación, porque involucra el manejo de datos catalogados como "sensibles" en la legislación, los cuales se rigen y se deben modelar principalmente bajo el contexto de la ley 20.584 "Derechos y deberes de los pacientes" y la ley 19.628 sobre "Protección de datos de carácter personal", así como también de normativas que se deben cumplir para la acreditación de prestadores públicos y privados [4] [5] [6].

Actualmente, la digitalización en salud es un hecho del cual se han reconocido los éxitos y beneficios que abarcan distintas áreas de conocimiento biomédico, en donde se generan distintos tipos de datos de información que pueden ser utilizados para diversos propósitos, como por ejemplo imágenes con las cuales se pueden entrenar redes neuronales que identifiquen anomalías de una manera muy precisa, o datos estadísticos recogidos de todas las instituciones de salud para elaborar estrategias epidemiológicas a nivel nacional [7] [8]. En este sentido, la cantidad de información que se genera día a día es un campo de interés para la ingeniería biomédica, la cual se suscita en la investigación y aplicación de tecnologías orientadas en mejorar la salud de las personas, por este motivo, la llamada ciencia de datos o *Data science* toma un rol importante para trabajar con un gran volumen de datos y mucha variedad de información en un área ampliamente demandante de desarrollo de tecnologías, y también a la vez sumamente nutritiva en cuanto a la información que se genera día a día. La ciencia de datos abarca desde científicos, educadores, hasta profesionales de la salud que reúnen información muy importante, quienes se encargan de organizar, analizar y convertir los datos en conocimiento, para luego usar tecnología que aplique ese conocimiento en,

por ejemplo, sistemas de apoyo a la decisión clínica [9]. Para estos propósitos, se utilizan distintas técnicas como la inteligencia artificial, *big data* o *machine learning.*

Con la manipulación de los datos se pueden generar modelos complejos de redes de conocimiento -*knowledge graphs- o* grafos de información, las cuales permiten aprender representaciones de conceptos biomédicos (nodos) y las relaciones que tienen éstos entre sí (lazos) [10]. Este aprendizaje puede ser aplicado para distintas dimensiones biomédicas, que van desde la generación de redes para predecir propiedades moleculares como el método propuesto por Gilmer *y col.* (2017) [11], el análisis de la genética humana como por ejemplo el grafo de Li *y col.* (2017) [12], o también para el aprendizaje de comportamientos farmacéuticos y sus riesgos como el método de predicción de comportamiento entre fármacos y objetivos que Zong *y col.* propusieron el año 2017 [13]. Por otro lado, podemos encontrar grafos que permiten ayudar en las decisiones clínicas, como el grafo que relaciona enfermedades y tratamientos de Rotmensch *y col*. (2017) [14]. Los grafos de conocimiento biomédico han sido llamados a lo largo del tiempo como grafos semánticos, redes, bases de conocimiento u ontologías, estos extraen conocimiento desde un largo volumen de información de documentos y bases de datos de distinta naturaleza, y vinculan esas relaciones como una red [10]. Éstos son una estructura importante para la siguiente generación de inteligencia artificial, y como lo visto anteriormente, tienen muchas aplicaciones en el área de la informática biomédica.

En este contexto, dentro del marco de los problemas de salud del plan AUGE, con las herramientas entregadas por el sistema de lenguaje médico unificado (UMLS) -detallada en la siguiente sección- y con el lenguaje de programación Python, se elaboró un algoritmo con el que se pudo analizar cientos de artículos científicos para extraer la coocurrencia de palabras claves o entidades, permitiendo así generar distintos nodos que se fueron relacionando mediante lazos cada vez que las entidades se encontraban en un artículo. De esta manera, se obtuvieron distintas redes de información que relacionan cada problema GES con distintas patologías, las cuales a su vez se relacionan con distintos signos, procedimientos y dispositivos médicos.

OBJETIVOS

1.1.1 Objetivo general:

El objetivo general de este trabajo de título es elaborar una red de conocimiento informático a través del lenguaje de programación Python con las librerías Pandas, Networkx y los recursos entregados por la *National Library of Medicine* (UMLS). Para construir esta red, se designaron las entidades **patologías**, **signos**, **procedimientos** y **dispositivos médicos**, que se identificaron en texto mediante el uso de técnicas de procesamiento de lenguaje natural en literatura científica de la base de datos Pubmed Central en el contexto los 85 problemas de salud GES.

1.1.2 Objetivos específicos:

1. Adquirir y adecuar el Metathesauro del sistema de lenguaje médico unificado y el listado específico de prestaciones del AUGE, para posteriormente generar el léxico, establecer una clasificación para cada uno de los 85 problemas y las expresiones para la búsqueda de artículos científicos.
2. Hacer distintas búsquedas de artículos científicos para cada problema del AUGE y almacenar cada corpus obtenido en distintos directorios, con el objetivo de aplicar técnicas de reconocimiento de entidades con procesamiento de lenguaje natural en cada artículo y así obtener las patologías relacionadas a cada problema.
3. Hacer búsquedas de artículos relacionados con cada una de las patologías asociadas a los problemas ges dentro de cada grupo, para posteriormente realizar procesamiento de lenguaje natural para encontrar signos, procedimientos y dispositivos médicos asociados a cada patología.
4. Generar nodos y lazos para formar la red de información.

2. Marco teórico

2.1 UNIFIED MEDICAL LANGUAGE SYSTEM (UMLS)

El sistema de lenguaje médico unificado es una herramienta que facilita el desarrollo de sistemas informáticos para "comprender" el lenguaje biomédico, con el objetivo de que investigadores puedan construir o mejorar sistemas para crear, procesar, recuperar e integrar información en salud o datos biomédicos. Este sistema tiene tres fuentes de conocimientos: Metathesaurus, red semántica, y un léxico llamado specialist. Metathesaurus es la base de datos que contiene la información de todos los conceptos biomédicos, por otro lado, la red semántica permite caracterizar los conceptos con sus relaciones, y finalmente el léxico specialist entrega herramientas con técnicas de procesamiento de lenguaje natural. Sin embargo, las fuentes de conocimiento red semántica y specialist están desarrolladas en entornos de programación distintos de Python, por lo que, para efectos de cumplir con el objetivo general de este trabajo, se trabajará solamente con el Metathesaurus de UMLS [15].

Metathesaurus

Metathesaurus es una base de datos multi propósito que contiene una gran cantidad de vocabulario de distintos idiomas acerca de conceptos biomédicos o relacionados a la salud. Este conjunto de información se construyó a partir de varios sets de códigos, clasificaciones, o vocabularios controlados de servicios de salud, estadísticas públicas y literatura biomédica. Se organiza por conceptos o significados, es decir, conecta todos los nombres alternativos, o también las diferentes "visiones" acerca de un mismo concepto e identifica relaciones útiles entre conceptos diferentes. Debe ser configurado para darle un uso efectivo en las distintas aplicaciones, esto significa que se debe crear un subconjunto de datos que tenga utilidad según sea el propósito, toda la información se encuentra etiquetada en cuanto al vocabulario de origen, por lo tanto, es posible determinar cuáles vocabularios se seleccionan para la construcción del Metathesaurus [15].

Los datos que conforman al Metathesaurus se encuentran en distintos archivos formato de texto enriquecido con la extensión ".RFF" por sus siglas en inglés *rich release format*. se encuentra dispuesta en una serie de líneas de texto con delimitadores de barra vertical "|" para separar datos dentro de cada línea, esto quiere decir que cada línea, según sea el archivo ".RFF" que se revise, puede contener distintos datos o valores. De manera general el Metathesaurus contiene conceptos, nombres de conceptos y otros atributos que se recogen de una cantidad mayor a 100 terminologías distintas, de las cuales pueden integrar, modificar o eliminar conceptos que el Metathesaurus seguirá almacenando en distintos archivos, como por ejemplo el archivo "MRSAB.RRF", que almacena la versión a la que la terminología modificó algún parámetro o valor para seguir teniendo acceso al dato sin actualizar. Por otro lado, la estructura del Metathesaurus se puede acomodar a traducciones de muchos lenguajes, por lo que tiene disponibles terminologías en distintos idiomas [15].

Como se mencionó anteriormente el Metathesaurus se organiza por conceptos, donde el principal propósito es poder conectar diferentes nombres del mismo concepto desde distintos vocabularios. Para eso, se asignan varios tipos de identificadores únicos, los cuales pueden incluir nombres de conceptos, identificadores o características claves de esos nombres (por ejemplo, el lenguaje, la terminología o el tipo de identificador), en lo concreto, la estructura completa de todos los conceptos aparece en un solo archivo llamado "MRCONSO.RRF". Cada concepto es un significado, y este significado puede tener muchos nombres, para cada concepto se otorga un **identificador de concepto único (CUI)**, un CUI puede ser removido del Metathesaurus cuando se descubre que dos CUIs nombran al mismo concepto (son sinónimos). A cada nombre de concepto o "*string*", en de todos los lenguajes del Metathesaurus le asigna un **identificador único de *string* (SUI),** al mismo tiempo, a cada ocurrencia de un *string*, dependiendo del vocabulario del

que provenga, se le asigna un **identificador único de átomo (AUI),** por lo mismo, distintos AUI se pueden conectar al mismo SUI, como se puede observar en la *ilustración 2* [15].

Concept (CUI)	Terms (LUIs)	Strings (SUIs)	Atoms (AUIs) * RRF Only
C0004238 Atrial Fibrillation (preferred) Atrial Fibrillations Auricular Fibrillation Auricular Fibrillations	**L0004238** Atrial Fibrillation (preferred) Atrial Fibrillations	**S0016668** Atrial Fibrillation (preferred)	**A0027665** Atrial Fibrillation (from MSH) **A0027667** Atrial Fibrillation (from PSY)
		S0016669 (plural variant) Atrial Fibrillations	**A0027668** Atrial Fibrillations (from MSH)
	L0004327 (synonym) Auricular Fibrillation Auricular Fibrillations	**S0016899** Auricular Fibrillation (preferred)	**A0027930** Auricular Fibrillation (from PSY)
		S0016900 (plural variant) Auricular Fibrillations	**A0027932** Auricular Fibrillations (from MSH)

Ilustración 1: Estructura de datos del Metathesaurus, se puede observar las diferencias entre CUI, SUI y AUIs [15]..

Otro archivo presente en el Metathesauro que es de interés para este trabajo es el llamado "MRSTY.RFF", en el cual se entrega la clasificación de tipo semántica para cada uno de los **CUI** que están presentes en el archivo "MRCONSO.RFF", esta clasificación se entrega mediante un identificador único de tipo (**TUI**) y el tipo semántico al que corresponde el identificador (**STY**). Los grupos semánticos que se identifican en el presente trabajo son:

TUI	STY	TUI	STY
T047	Síndrome o enfermedad	T060	Procedimiento de diagnóstico
T046	Función patológica	T059	Procedimiento de laboratorio
T191	Proceso neoplásico	T063	Técnica de investigación molecular
T048	Comportamiento o disfunción mental	T061	Procedimiento terapéutico o preventivo
T037	Daño o envenamiento	T203	Dispositivo de suministro de fármaco
T033	Hallazgo de anomalía	T074	Dispositivo médico
T019	Anormalidad congénita	T075	Dispositivo de investigación
T190	Anormalidad anatómica	T020	Anormalidad adquirida

Tabla 1: grupos semánticos y significados utilizados en este trabajo [15].

2.2 Extracción de Información biomédica

Vocabularios, estándares y ontologías médicas

Si bien el ámbito de vocabularios controlados para el uso de tecnologías en salud está siendo cada vez más desarrollado, aún existe heterogeneidad en los estándares que se utilizan alrededor de todo el mundo. En Chile, los estándares más utilizados son SNOMED-CT para las terminologías médicas, LOINC para laboratorio clínico, DICOM para imagenología, CIE-9/10, CPT, entre otros [2]. Donde en el caso de los vocabularios terminológicos como SNOMED-CT asocian un código a un término de manera que se unifiquen las diferentes expresiones que puedan corresponder, por ejemplo, a la misma enfermedad.

Extracción de información

Una de las aplicaciones que permite *deep learning*, según Shickel y col. (2018) [16], es el procesamiento de la información presente en los registros clínico electrónico. Dentro de las notas clínicas que poseen texto libre, existe mucha información asociada a las visitas en las consultas médicas, observaciones del profesional, tratamientos realizados, órdenes de referencia, etcétera. Es en este espacio en donde existen los mayores desafíos de extraer datos, esto porque la información presente se encuentra de una manera no estructurada, habiendo diferentes estrategias para realizar esta tarea: (1) De las estrategias más generales, es la **extracción de conceptos únicos** como lo pueden ser enfermedades, tratamientos, o conceptos médicos en general, utilizando **la técnica de procesamiento de lenguaje natural (NLP), reconocimiento de entidades nombradas (NER).** (2) Una tarea más compleja es la **extracción de eventos temporales**, como, por ejemplo, "*en los últimos meses* presentó(..)". (3) La **extracción de relaciones** entre conceptos médicos como "el examen X reveló el problema médico Y". (4) Finalmente, otra tarea de la extracción de información es la de **unificar todas las abreviaturas médicas** que manejan los profesionales de la salud, dado que ejemplos como *TAC* o *CT* pueden, en términos prácticos, significar lo mismo para el procesamiento de los datos.

Representación de conceptos médicos

El primer objetivo de esta área es generar representaciones vectoriales desde códigos médicos dispersos, de manera que los conceptos similares estén cercanos en un espacio vectorial de bajas dimensiones, en este sentido, se pueden analizar códigos de fuentes de datos heterogéneas que pueden ser agrupados con distintas técnicas [16]. Algunas veces estas representaciones son llamadas ***word embbedings*** [17] que, como se explicará en el punto 2.3 de esta sección, representan un punto en el espacio de baja dimensión, dando como resultado que palabras con significados parecidos sean vecinos. Es importante señalar que hay redes neuronales que pueden definir embeddings en otros campos de aplicación como, por ejemplo, en capas ocultas de redes convolucionales para proveer embeddings de imágenes, aunque sigue siendo un área profundamente utilizada en el procesamiento de lenguaje natural. Por otro lado, fuera del procesamiento de lenguaje natural, para generar representaciones de conceptos de otra forma se utiliza la **autocodificación**, en donde se ha demostrado que existe una mejora en la forma que se hacen las relaciones entre conceptos cuando se utilizan autocodificadores [18].

2.3 PROCESAMIENTO DE LENGUAJE NATURAL Y RECONOCIMIENTO DE ENTIDADES NOMBRADAS (NER).

Representación de palabras para procesamiento: Aprendizaje de representación o *Representation learning:*

Mucha de la información que se recopila de texto biomédico libre sin estructura, la cual puede contener distintos códigos médicos, pueden ser representados de muchas maneras dependiendo de la complejidad que la tarea de aprendizaje de representación requiera, generalmente, las buenas representaciones son las que permiten facilitar la tarea de aprendizaje. Las representaciones en concreto son vectores numéricos de ***d*** dimensiones que agrupan un conjunto de ***d*** características de un objeto o concepto, se pueden utilizar para diferentes tareas independientemente de cómo se obtuvieron, éstas tratan de abarcar la mayor cantidad de información entrante manteniendo la independencia entre cada una de las representaciones, y también permiten realizar aprendizaje no-supervisado y semi-supervisado con datos que no están lo suficientemente etiquetados [17].

En el texto libre con lenguaje biomédico, se pueden asignar códigos a diferentes términos según su ontología, situación que falla al utilizar conceptos de distintos vocabularios, es por esto que los enfoques más recientes de deep learning utilizan codificación como un espacio de vectores de características para información más detallada como, por ejemplo, diagnósticos, procedimientos o medicamentos [16]. Así es

como se pueden caracterizar a pacientes como una colección ordenada de códigos con eventos médicos para distintas aplicaciones como la detección y predicción de eventos adversos.

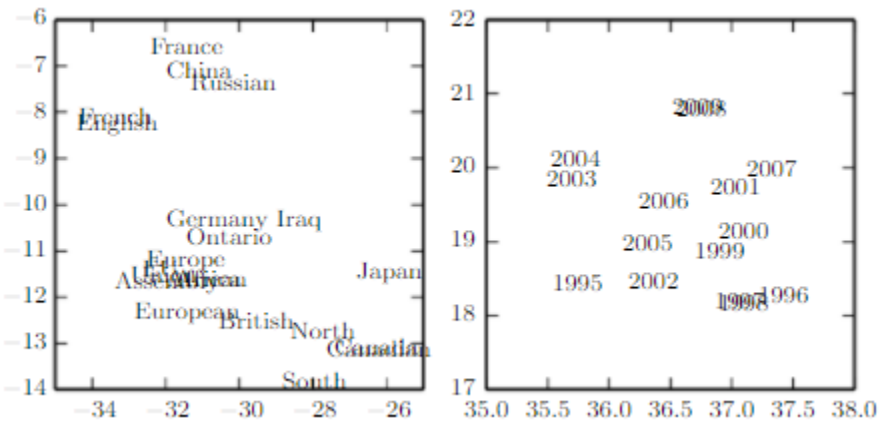


Ilustración 2: Visualización de dos dimensiones de word embeddings obtenidos de un modelo de traducción con una red neuronal. Se observa que hay palabras que están semánticamente relacionadas que están cercanas. Tener en cuenta que es un embeddings de dos dimensiones con propósito de visualizar, en la práctica, estos embeddings tienen una alta dimensionalidad y simultáneamente pueden capturar muchos tipos de similitudes entre palabras. Fuente: Goldfelloy y col. (2016) [17]

A continuación, se repasarán los elementos esenciales para llevar a cabo el análisis de texto biomédico mediante el uso de ***word embeddings*** para hacer representaciones de los conceptos médicos, para esto se detallarán las definiciones realizadas por Aggarwal (2018) [19] en su libro *Machine learning for text*.

Bases para el procesamiento de lenguaje natural:

Para el análisis de texto libre y para así realizar representaciones de los conceptos dentro de este, se debe tener en cuenta que el orden de las palabras no provee un entendimiento del contexto por sí solo, ni tampoco por la frecuencia en la que se repiten las palabras, sin embargo, existen dos enfoques para realizar estas representaciones: (1) **Texto como una bolsa de palabras**, del cual el ordenamiento de las palabras no es usado para el análisis de texto, el set de palabras es convertido en una representación multidimensional dispersa, por lo tanto, el universo de palabras corresponde a las dimensiones (o características). (2) **Texto como set de secuencias**: En este caso, las oraciones en el texto son extraídas como *strings* o secuencias, por lo tanto, el orden de las palabras sí importa en la representación, aunque el orden es localizado dentro de los límites de oraciones o párrafos; este enfoque es utilizado en aplicaciones que requieren de gran interpretación semántica. Un **corpus** es un set de datos que corresponde a una colección de documentos, así mismo, el set de palabras usadas para definir al corpus es referido como *lexicón*, y por último se refiere a las dimensiones como los términos o características, donde es muy importante siempre normalizar estadísticamente la importancia relativa y la cantidad de éstas. Cuando se analizan representaciones multidimensionales de texto, tal como se detalló al comienzo de esta sección, existe una notación consistente, en donde se asume que un corpus **D** con **n** documentos y **d** términos diferentes, puede ser representado como una matriz documento-término dispersa tamaño **n x d.** La fila posición "i" de **D** se representa por el vector d-dimensional $\overline{\overline{X}}_i$, por lo tanto, un corpus puede ser un set de estos vectores d dimensionales denotado por $D = \{\overline{\overline{X}}_1 \ldots \overline{\overline{X}}_n\}$.

Reconocimiento de entidades nombradas:

Ésta es una aplicación del procesamiento de lenguaje natural, consiste en encontrar entidades (como personas, locaciones patologías, etc.) en una oración, a cada palabra o frase encontrada se le asocia una etiqueta con la clasificación correspondiente a la entidad a la que pertenece. Se detallarán las estrategias actuales para lograr esta aplicación en el estado del arte de este trabajo.

2.4 REDES DE CONOCIMIENTO O GRAFOS DE INFORMACIÓN.

Las redes de conocimiento -*knowledge graphs- o* grafos de información, permiten aprender representaciones de entidades de todo tipo (**nodos**) y las relaciones que tienen éstos entre sí (**lazos**), son utilizados en diversas áreas de las ciencias y han permitido la generación de redes importantes que utilizamos día a día, ya sean en algoritmos de búsqueda como Google [20] o en redes sociales como Facebook [21].

Una estructura sencilla para la construcción de un grafo consiste en una tabla con tres columnas, en la cual se pueden añadir filas con dos nodos y un lazo, como se muestra en el siguiente ejemplo:

Sujeto	**Predicado**	**Objeto**
Ingeniería civil biomédica	Carrera	Universidad de Valparaíso
Trabajo de título 2	Asignatura	Ingeniería civil biomédica
Bioinstrumentación 1	Asignatura	Ingeniería civil biomédica

Tabla 2: Ejemplo de estructura para grafo simple.

Con esta estructura se puede graficar una red de información básica, la cual contiene cuatro nodos y tres lazos, la que se puede visualizar la *ilustración 3*, en donde los lazos de color rojo corresponden a "Asignatura" y el de color azul corresponde a "Carrera". Sin embargo, las estructuras utilizadas en *DataScience* o *Machine learning* son más complejas y de mayor número de información.

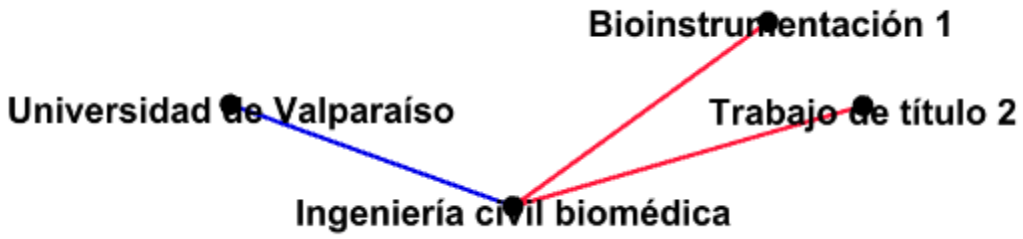


Ilustración 3: Ejemplo de grafo o red de información, corresponde a cuatro nodos formando tres lazos, los lazos de color rojo corresponden a "Asignatura" y el de color azul a "carrera" .

Dentro del área del aprendizaje de relaciones mediante estadística, las representaciones de un objeto pueden contener las relaciones de éste con otros objetos, de esta forma se genera un grafo contenido por nodos y lazos etiquetados. Actualmente existen tecnologías y grafos ampliamente desarrollados que contienen millones de nodos y relaciones, estos se forman con estrategias de estadística y aprendizaje profundo para formar grafos de conocimiento a larga escala, los cuales almacenan información como base de datos con relaciones entre distintas entidades [22].

La precisión y la calidad de los datos almacenados son parámetros importantes para determinar la utilidad de los grafos de conocimientos, y un factor importante para determinar buenos parámetros es la forma en que se construyen estas redes, de las cuales Nickel y col [22]. destacan 4 formas distintas: (1) Con un enfoque de selección, creado manualmente por un grupo cerrado de expertos. (2) Un enfoque colaborativo, en donde se catalogan los nodos y lazos de manera manual con un grupo de voluntarios. (3) En un enfoque automatizado con texto semiestructurado, en el cual se utilizan reglas creadas manualmente, reglas aprendidas o expresiones regulares para generar las entidades. (4) Y finalmente con un enfoque automatizado con entradas sin estructura, donde se crean las entidades a partir de texto sin estructura, con técnicas de *machine learning* o procesamiento de lenguaje natural.

3 Estado del arte

En la siguiente sección se explican las estrategias actuales para abordar el procesamiento de lenguaje natural, primero con el enfoque de Word2vec [23] en donde se analizan secuencias cortas de texto para comprender contextos y semánticas, y luego se explica el uso de Transformadores [24], los cuales utilizan otra estrategia, capaces de analizar párrafos completos de texto en el mismo instante, luego en las siguientes dos secciones, se describen distintos recursos disponibles para Python, algunos utilizados en este trabajo y otros disponibles de acceso abierto.

3.1 ALGORITMOS PARA EL PROCESAMIENTO DE TEXTO: SKIP-GRAM Y TRANSFORMADORES.

Word2vec y algoritmo skip-gram para el procesamiento de lenguaje natural:
Un enfoque para el procesamiento de lenguaje natural y para el aprendizaje de máquina con redes neuronales es el propuesto por Mikolov y col. (2013) [23] llamado word2vec, en el cual se utiliza la técnica de análisis de texto llamada skip-gram. Ésta es una técnica ampliamente utilizada en el campo del procesamiento del lenguaje, en donde se utiliza una selección de frases secuencialmente, de manera que en cada iteración se forman *n-grams* (*bi-grams*, *trigrams*, etc) que reúnen palabras adyacentes que cambian constantemente. Se definen ***k-skip-n-grams*** para las oraciones $\boldsymbol{\omega_1} \ldots \boldsymbol{\omega_m}$ como:

$$\left\{\omega_{i_1}..\omega_{i_2}..\omega_{i_n}\middle|\ \sum_{j=1}^{n} i_J - i_{j-1} < k\right\} \qquad (1)$$

Para ejemplificar esta fórmula, a continuación se demuestra el uso de estos *n-grams* en la frase "Dispositivos médicos fundamentales en salud".

- **Bi-grams** = {Dispositivos médicos, médicos fundamentales, fundamentales en, en salud}
- **2-skip-bi-grams** = {Dispositivos médicos, Dispositivos fundamentales, Dispositivos en, Dispositivos salud, médicos fundamentales, médicos en, médicos salud, fundamentales en, fundamentales salud}
- **Tri-grams** = {Dispositivos médicos fundamentales, médicos fundamentales en, fundamentales en salud}
- **2-skip-tri-grams** = {Dispositivos médicos fundamentales, Dispositivos médicos en, Dispositivos médicos salud, Dispositivos fundamentales en, Dispositivos fundamentales salud, Dispositivos en salud, médicos fundamentales en, médicos fundamentales salud, médicos en salud, fundamentales en salud}

Un modelo de entrenamiento que utiliza la estructura skip-gram, *aprende* representaciones vectoriales de las palabras a partir de la coocurrencia de adyacencias similares, es decir, de n-grams, de manera que los segmentos similares queden cercanos en un espacio vectorial. Para el aprendizaje, se utilizan además arquitecturas de redes neuronales, de manera que las primeras capas se encargan de reconocer las palabras principales en un contexto para finalmente en las capas más internas tener la posibilidad de predecir palabras de un contexto específico [24].

Atención es todo lo que se necesita: Uso de Transformers para procesamiento de lenguaje natural:
En el año 2017, investigadores de Google Brain y Google Research publicaron el artículo "Atención es todo lo que necesitas" [25]. En este artículo se detalló un nuevo modelo para realizar tareas de procesamiento de lenguaje natural (NLP), **el transformador**, el cual mejoró los resultados respecto a los modelos NLP existentes, ya que se entrena más rápido y tiene mejores resultados de evaluación [26].

El modelo original presentado por el artículo de Vaswani y cols [25]. introduce el elemento esencial del transformador, el **mecanismo de atención**; Si bien el modelo original tuvo grandes resultados y revolucionó el estado del arte en esta materia, a la fecha de hoy el transformador original ha sido modificado constantemente para obtener resultados en tareas específicas, es así donde se han quitado partes, añadido

otras, o utilizados entornos distintos de configuración. A continuación, se muestra una ilustración de la arquitectura del transformador original:

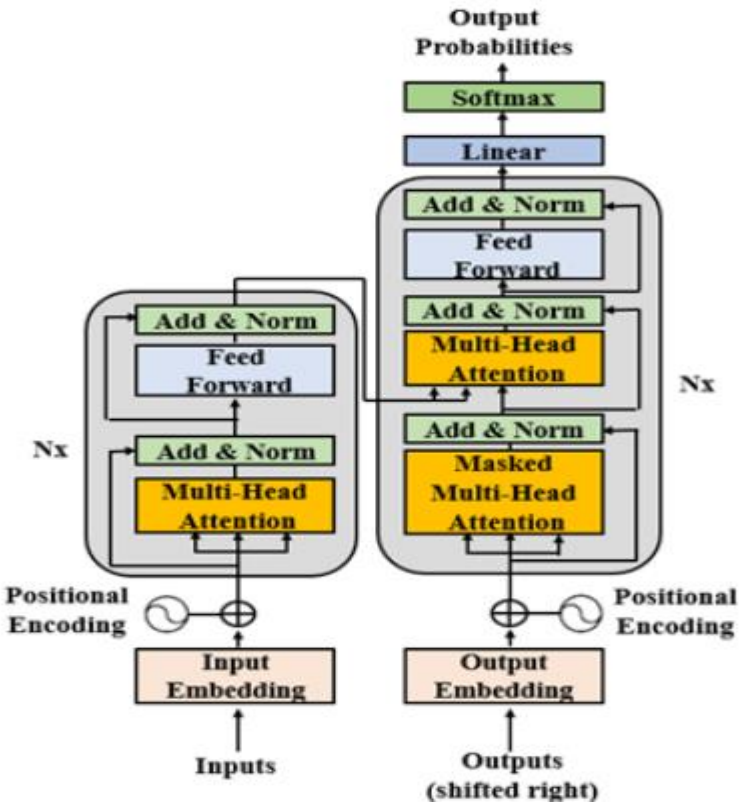


Ilustración 4: Estructura del transformador. En la izquierda se observa que los embeddings de entrada pasan por el mecanismo de atención y por una red feed foward una cantidad de N veces. En la derecha entran los embeddings de entrenamiento [26].

Para profundizar en este mecanismo, en el libro *Transformers for natural language processing* de Rothman se detallan todos los pasos para la construcción de esta arquitectura en Python. De las principales características que observa Rothman del transformador [26], destaca que no utiliza redes neuronales recurrentes, redes LSTM, ni redes convolucionales. Por otro lado, notamos que se abandona la coocurrencia de frases o palabras, la cual es reemplazada por el mecanismo de atención, en donde se requiere una serie de operaciones previas como la distancia entre dos palabras – o su posición relativa en el texto- para poder operar. Este mecanismo opera palabra a palabra, lo que significa que encuentra cómo cada palabra se relaciona con todas las otras palabras de una secuencia, incluyendo la misma palabra analizada.

En síntesis, este mecanismo revolucionó el área del procesamiento de lenguaje, mejorando en tareas de reconocimiento de habla, traducción, utilización de *bots* para preguntas y respuestas, predicción de palabras, entre otras muchas tareas en esta área. A partir de este modelo de procesamiento nacen muchos recursos de acceso abierto en distintos lenguajes de programación, entre los cuales destacan BERT [21], ROBERTA [27], t5-small [28], entre otros. Cada modelo propuesto puede ser entrenado para tareas específicas como las nombradas anteriormente, y hay muchos de acceso abierto para ser utilizadas por cualquier investigador que quiera generar alguna aplicación.

3.2 RECURSOS PARA EL PROCESAMIENTO DE LENGUAJE NATURAL APLICADOS EN LA INFORMÁTICA BIOMÉDICA .

Librería HuggingFace: En el ámbito de la inteligencia artificial, la empresa Huggingface ha sido gran desarrolladora de modelos y datasets de entrenamiento que se utilizan actualmente en otras grandes corporaciones internacionales como Google, Microsoft, Facebook, etc. En su sitio web existen más de cincuenta mil modelos para realizar tareas como clasificación de imágenes, traducción de lenguajes, reconocimiento de habla, clasificación de audio, bots de preguntas y respuestas, entre otras más [29].

Para lenguajes de programación como Python, HuggingFace tiene a disposición la librería llamada **Transformers** con la cual se pueden realizar las tareas nombradas anteriormente. Además, dentro de la librería Transformers se entregan recursos para optimizar tareas de procesamiento de lenguaje natural, algunos de éstos son los **Tokenizadores**, con los cuales se pueden "tokenizar" o identificar palabra a palabra en una secuencia extensa de texto de manera "extremadamente rápida", pudiendo así tokenizar 1GByte de texto en 20 segundos [30]. Por otro lado, esta librería también tiene a disposición la función **Pipeline,** con la cual se pueden utilizar **modelos pre-entrenados** disponibles en la web de HuggingFace en tareas como el reconocimiento de entidades nombradas (NER), tal cual se muestra en las siguientes líneas de código Python.

Utilizar función Pipeline para NER en librería Transformers de HuggingFace

```
!pip install transformers

from transformers import pipeline, AutoModelForTokenClassification
from transformers import AutoTokenizer

tokenizador= AutoTokenizer.from_pretrained("dslim/bert-base-NER")
modelo= AutoModelForTokenClassification.from_pretrained("dslim/bert-base-NER")
ner= pipeline("ner", model=modelo, tokenizer=tokenizador)

ejemplo = "Estudio ingeniería civil biomédica hace seis años"

resultados_ner = ner(ejemplo)

print(resultados_ner)
```

Como se puede observar, utilizando esta librería con al menos 5 líneas de código ya se pueden realizar tareas de procesamiento de lenguaje natural, en donde para este ejemplo se tendría que utilizar un modelo pre-entrenado en el idioma español, ya que el modelo *bert-base-NER* tiene mejor rendimiento con texto en inglés, sin embargo, utilizando un modelo pre-entrenado óptimo para esta tarea tendría hipotéticamente un resultado de la forma: "Estudio" -> Verbo; "ingeniería civil biomédica" -> Carrera universitaria; "hace seis años" -> tiempo pasado.

Modelos: Biobert, Clinical Transformer NER y MedCat.
Con la utilización de los recursos explicados en la sección anterior se han desarrollado modelos complejos y entrenados en el área de la informática biomédica, éstos son algunos ejemplos dentro de una cantidad mayores de recursos de acceso abierto disponibles en sitios como GitHub o el sitio web de MedCat.

- Biobert [31] es un modelo de representación de lenguaje pre-entrenado con texto biomédico de artículos científicos de la base de datos PubMed para así realizar tareas de minería de texto como NER, extracción de relaciones entre entidades, preguntas y respuestas, entre otras. Utiliza transformadores en base al modelo de HuggingFace llamado BERT [32].

- ClinicalBert [33] es un modelo de word embeddings contextuales al área biomédica pre-entrenado con la base de datos de notas clínicas llamada MIMIC-III [34] para realizar tareas de procesamiento de lenguaje natural.
- MedCat [35] es un modelo con herramientas disponibles para procesamiento de lenguaje natural que se apoyan con las terminologías del sistema de lenguaje médico unificado (UMLS), utiliza transformadores y skip-gram para reconocer entidades biomédicas y asignarle un código CUI.

3.3 RECURSOS PARA LA GENERACIÓN DE GRAFOS DE INFORMACIÓN EN PYTHON.

Existen recursos disponibles en Python para la generación de las redes o grafos que se explicaron en la sección 2.4 de este trabajo, a continuación, se describirán brevemente alguno de ellos.

- Pykg2vec [36] es una librería de Python para aprender representaciones de entidades y relaciones en un grafo de conocimiento basados en Pytorch 1.5 [37].
- Librería KGTK: Knowledge Graph Toolkit [38] es una librería de Python para la manipulación de grafos de conocimiento. Permite operaciones como la extracción de subgrafos, filtrado, embeddings, entre otras [39].
- Librería NetworkX [40] es una librería de Python para la creación, manipulación y estudio de estructuras de grafos dinámicos y otras estructuras más complejas [41].

Para el presente trabajo se hizo uso de la librería Networkx, por lo que se hará una breve explicación de la sintaxis utilizada para la creación de la red, con sus nodos y lazos.

Utilizar librería Networkx para creación de red de información.

```
!pip install network

Import network as nx

Red = nx.Graph()
Nodos=[("hipertensión",{'Entidad':'Patología'}),("Esfigmomanómetro",{'Entidad'
:'Dispositivo'})]
Lazo=('hipertensión','Esfigmomanómetro')

Red.add_nodes_from(Nodos)
Red.add_edges_from(Lazo)

import matplotlib.pyplot as plt
nx.draw(Red)

nx.write_gexf(Red, "Grafo de ejemplo.gexf")
```

En este código Python se puede observar que para crear nodos y lazos se deben **estructurar los datos como tuplas**, para el caso de nodos, estas tuplas deben conformarse por el nombre que recibirá el nodo y las características asociadas al nodo, en donde estas últimas deben estar dentro de un diccionario. Los lazos, de forma similar, deben estar conformados por dos nodos en estructura de tupla, y puede agregarse un tercer elemento que describa las características del lazo. Además, esta librería tiene sentencias para graficar un grafo de manera sencilla con la librería matplotlib.

Finalmente, para procesar y visualizar grafos con mayores volúmenes de datos, existen software como Gephi [42] con en los que se puede importar un grafo formato ".gexf".

3.4 ARTÍCULOS DE REFERENCIA

En el área de la informática biomédica, existen grafos a distintas escalas de conocimiento que pueden variar entre las relaciones moleculares de fármacos y tejidos [43], hasta las que se generan entre enfermedades y tratamientos como el grafo KEGG [44]. Para visualizar la estructura de los grafos de conocimiento biomédicos, en la *ilustración 6* se muestran las dimensiones que abarcan las representaciones de los conceptos.

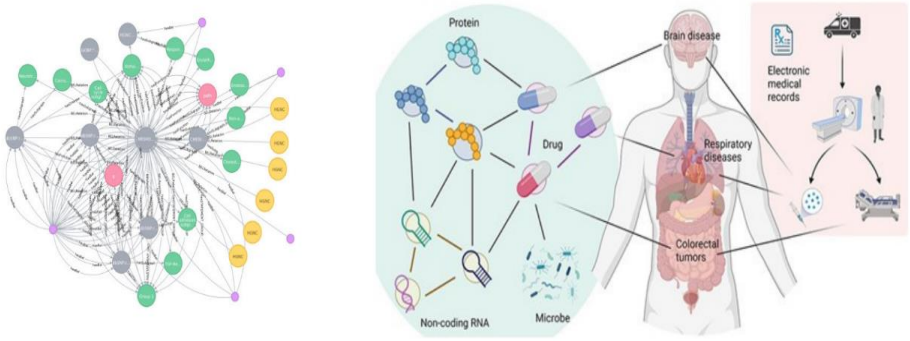


Ilustración 5: Dimensiones de los grafos de conocimiento biomédicos, a la izquierda se observa cómo se unen los distintos nodos y lazos de un grafo asociado a la genética del Alzheimer [45]. En la derecha se ilustran las dimensiones en las que se pueden relacionar conceptos, que van desde las interacciones moleculares, comportamientos fisiológicos, hasta las relaciones entre entidades de los sistemas de salud [10].

Para este trabajo, se utilizaron prácticas realizadas en el artículo "Construyendo el grafo de la medicina a partir de millones de narrativas clínicas" por Finlayson y cols. (2014) [46], en el cual se analizaron veinte millones de narrativas clínicas, abarcando diecinueve años de recolección de información. Con este volumen de información, se extrajeron un millón de conceptos clínicos, con los que se generaron matrices de co-ocurrencia para cuantificar la relación entre todos los conceptos utilizando métodos estadísticos.

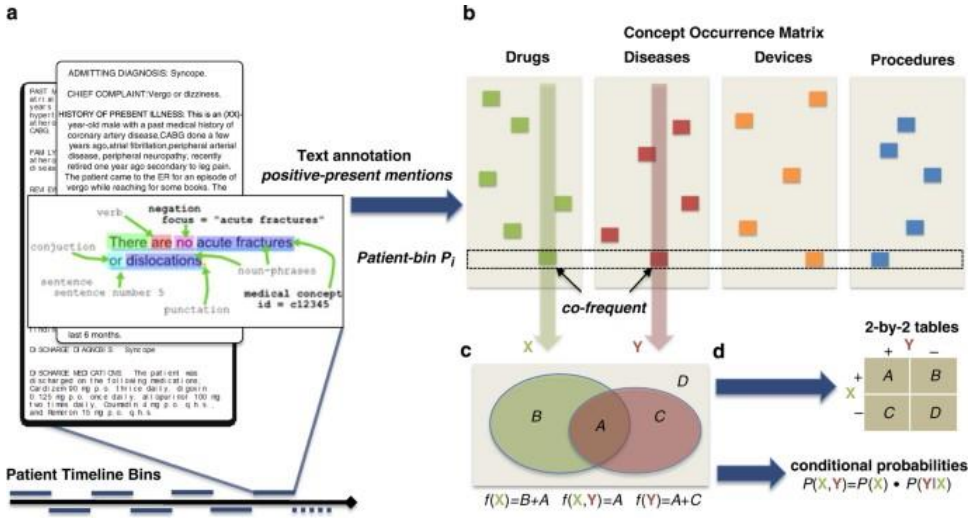


Ilustración 6: Arquitectura de trabajo para generar las matrices de co-ocurrencia, en la parte izquierda se observa que se utilizó reconocimiento de entidades en notas clínicas(a) para luego, en la izquierda, contar las co-ocurrencia y catalogar en grupos distintos (b). En c y d, se utilizan métodos estadísticos para cuantificar las relacione.

4 Metodología e Implementación

A continuación, se hará una descripción paso a paso de la metodología utilizada para llevar a cabo el objetivo general. Esta sección se divide en la enumeración de las actividades para realizar cada objetivo específico. Además, los distintos pasos enumerados van en concordancia con el cuaderno de programación adjunto en el anexo 1 de este trabajo, de manera que solo se acotará a describir las etapas sin mayor detalle de la sintáctica del lenguaje de programación.

En primer lugar, antes de comenzar con el detalle de cada actividad se esquematizarán en la siguiente tabla las actividades principales desarrolladas en cada objetivo específico.

• **Objetivo general: Elaborar una red de conocimiento informático a través del lenguaje de programación Python con las librerías Pandas, Networkx y los recursos entregados por la *National Library of Medicine* (UMLS) en el marco de patologías del plan AUGE.**	
Objetivo específico 1**: Adquirir y adecuar el Metathesauro del sistema de lenguaje médico unificado y el listado específico de prestaciones del auge, para posteriormente generar el léxico, establecer una clasificación para cada uno de los 85 problemas y las expresiones para la búsqueda de artículos científicos.**	• Obtención de licencia para trabajar con datos del sistema de lenguaje médico unificado (UMLS). • Adquisición de terminologías y codificación biomédica. • Descarga de listado específico de prestaciones disponibles ne la web del plan AUGE. • Selección y clasificación de problemas de salud. • Establecer expresiones para la búsqueda de artículos relacionadas a los problemas seleccionados.
Objetivo específico 2: Hacer distintas búsquedas de artículos científicos para cada problema del AUGE y almacenar cada corpus obtenido en distintos directorios, con el objetivo de aplicar técnicas de reconocimiento de entidades con procesamiento de lenguaje natural en cada artículo y así obtener las patologías relacionadas a cada problema.	• Realizar la búsqueda de artículos en relación con las expresiones generadas del objetivo anterior • Almacenar artículos en archivos de texto con estructura estándar. • Realizar procesamiento de lenguaje natural para la detección de patologías • Almacenamiento de patologías en archivos csv • Establecer expresiones de búsqueda referente a las patologías encontradas
Objetivo específico 3: Hacer búsquedas de artículos relacionados con cada una de las patologías asociadas a los problemas ges dentro de cada grupo, para posteriormente realizar procesamiento de lenguaje natural para encontrar signos, procedimientos y dispositivos médicos asociados a cada patología.	• Realizar la búsqueda de artículos en relación con las expresiones de cada patología • Almacenar artículos en archivos de texto con estructura estándar. • Realizar procesamiento de lenguaje natural para la detección de signos, procedimientos y dispositivos médicos • Almacenamiento de entidades en archivo formato csv
Objetivo específico 4: **Generar nodos y lazos para formar la red de información.**	• Normalizar y acondicionar las entidades encontradas en idioma español. • Generar nodos y lazos • Guardar red en formato de grafo

Tabla 3: Resumen de actividades realizadas en metodología de investigación.

4.1 OBJETIVO ESPECÍFICO 1: ADQUIRIR Y ADECUAR EL METATHESAURO DEL SISTEMA DE LENGUAJE MÉDICO UNIFICADO Y EL LISTADO ESPECÍFICO DE PRESTACIONES DEL AUGE, PARA POSTERIORMENTE GENERAR EL LÉXICO, ESTABLECER UNA CLASIFICACIÓN PARA CADA UNO DE LOS 85 PROBLEMAS Y LAS EXPRESIONES PARA LA BÚSQUEDA DE ARTÍCULOS CIENTÍFICOS.

1 Para cumplir con el objetivo específico número 1, en primer lugar, se procede a hacer registro en la página de la *National Library of Medicine (US)* con el objetivo de adquirir la licencia que permite descargar los datos del UMLS, una vez hecho el registro, luego de un plazo de 3 días se aprueba la licencia y se habilita la opción de descargar un archivo de 33,4 GB.

2 El segundo paso, luego de la descarga del archivo del UMLS, es hacer la instalación del Metathesaurus para obtener la codificación de las terminologías, para ello se ejecuta el archivo llamado *run64* (ver anexo 1) que abre al software metamorphosys, una herramienta para instalar los diferentes componentes del UMLS como Metathesaurus, red semántica y Lexicon. Al abrir Metamorphosys, se entrega la opción de instalar los tres componentes y seleccionamos solamente el primero, luego se abre la ventana de configuraciones iniciales donde seleccionamos las carpetas de destino de los archivos y también el formato de éstos (*Rich release format*). En el siguiente paso se seleccionan las terminologías que conformarán al Metathesauro anexo 2, una vez terminado este proceso se selecciona *done* y se procede a instalar el Metathesaurus

3 Procesar los archivos del Metathesauro como DataFrame de la librería Pandas, esto se realizó programando un código Python que iteró en todas las líneas de los archivos formato *rich release format* del Metathesaurus llamados CONSO.RFF y STY.RFF, en donde a partir de cada salto de línea (‘\n’) se distinguieron las filas que contenían los elementos de las columnas separados por la barra vertical “|”. Una vez procesados, se almacenaron dos archivos formato csv llamados “**umls_conso.csv**” y “**umls_sty.csv**” para utilizarlos nuevamente.

4 Descargar el archivo excel del listado específico de prestaciones GES disponible en la web AUGE, una vez descargado se procede a acondicionar los datos para extraer la información necesaria. Para esto se eliminaron todas las filas vacías y se seleccionaron los 85 problemas de salud presentes, los que se almacenaron en un archivo csv.

5 Una vez reunidos los 85 problemas de las GES, se hizo un excel para categorizarlos en 6 criterios a partir del número y nombre de cada problema, el primer criterio utilizado fue la asociación de un CUI de la UMLS que engloba el concepto biomédico principal de cada problema de salud, esto se realizó mediante la búsqueda manual de CUIs en el buscador de términos del portal web del Metathesauro UMLS, la segunda categorización fue asignarle el nombre del CUI asociado, en tercer lugar se categorizó a cada problema con una población objetivo a la que abarca, de manera que hayan problemas que abarcan a todas las edades, solo niños, adultos mayores, etc. Luego en cuarto lugar se asoció una expresión booleana en inglés a los problemas que aplicaban a una población objetivo específica, como por ejemplo a niños o adultos mayores, con el objetivo de hacer expresiones de búsqueda para descargar a artículos o más similares al nombre de cada problema. Finalmente, se categorizó a cada uno de los 85 problemas dentro de 13 categorías según las similitudes que tienen desde el punto de vista fisiológico y biomédico, con el objetivo de realizar el procesamiento de la información para generar un grafo de información por grupos.

6 Luego de clasificar los problemas, se hizo una selección de solo un grupo de los problemas del plan AUGE para el desarrollo de los siguientes objetivos de este trabajo, , pertenecientes al grupos

"problemas cardiovasculares" Una vez seleccionados, con el archivo excel creado en el punto anterior se generó un archivo csv con las expresiones de búsqueda para cada término de los problemas de las ges y todas sus expresiones en inglés entregadas por el UMLS, es decir, se realizaron búsquedas con todos los términos LUIs asociados al CUI que corresponde al problema de las GES más la expresión "AND (related diseases)", con el objetivo de buscar artículos de las patologías relacionadas a esos problemas. Adicionalmente, se consideraron las expresiones booleanas en caso de ser necesarias para los problemas que englobaban una población objetivo.

7 Para finalizar con este objetivo específico, se organizaron directorios para almacenar los artículos a descargar en dos grupos distintos, diferenciándose en la clasificación de cada problema de las GES seleccionadas en el punto anterior, es decir, se hicieron dos carpetas para almacenar en cada una dos corpus, uno para obtener patologías asociadas (Objetivo N°2) y otro corpus para identificar las otras entidades relacionadas a cada patología (Objetivo N°3).

4.2 OBJETIVO ESPECÍFICO 2: HACER DISTINTAS BÚSQUEDAS DE ARTÍCULOS CIENTÍFICOS PARA CADA PROBLEMA DEL AUGE Y ALMACENAR CADA CORPUS OBTENIDO EN DISTINTOS DIRECTORIOS, CON EL OBJETIVO DE APLICAR TÉCNICAS DE RECONOCIMIENTO DE ENTIDADES CON PROCESAMIENTO DE LENGUAJE NATURAL EN CADA ARTÍCULO Y ASÍ OBTENER LAS PATOLOGÍAS RELACIONADAS A CADA PROBLEMA.

Antes de comenzar con la metodología de almacenamiento de texto de los artículos científicos, se buscaron recursos de otros programadores en GitHub que tuvieran acceso libre para poder obtener los cuerpos de los artículos, búsqueda que concluyó en los recursos entregados por cyclecycle [47] dentro del repositorio llamado parse-pubmed. Este recurso se apoya de la librería BioPython [48] y de los recursos Entrez de Pubmed [49], consiste en obtener y estructurar artículos obtenidos de distintas bases de datos asociadas a Pubmed en forma de clases y funciones, metodología que simplifica la obtención del título, cuerpo, palabras claves, códigos de identificación de manera simple en líneas cortas de programación. Sin embargo, el código fue modificado y adaptado para estructurar los distintos artículos en archivos formato de texto almacenados en el directorio principal de este trabajo.

1 Una vez que se clonan e instalan los recursos y librerías descritas en el punto anterior, se procede a programar la metodología de adquisición de los artículos para ser almacenados en formato de **string** con separadores de forma **"||\n"** en archivos de texto, para la búsqueda de los artículos, éstos se obtienen a partir de una expresión de búsqueda en la base de datos Pubmed Central (PMC). Cada archivo contiene los identificadores, el título, resumen, cuerpo y palabras claves. Además, el nombre de cada archivo corresponde a una codificación con separadores de la forma **"_"** que contiene el número del problema ges con otros identificadores para evitar la sobreposición de los archivos**.** Por otro lado, es necesario destacar que en el proceso previo de la descarga de artículos se obtenían archivos que se almacenaban sin cuerpo del artículo mismo, motivo por el cual se seleccionó a éstos para almacenarse en otro directorio llamado **"papers sin cuerpo".**

2 Una vez establecida la metodología de adquisición de artículos, se procede a realizar una iteración de las expresiones de búsqueda obtenidas en el objetivo específico anterior en la función descrita en el punto anterior para almacenar **diez artículos por cada término de búsqueda** relacionado con las patologías de las GES seleccionadas, en la carpeta llamada "*corpus patologías*".

3 Luego de almacenar los artículos científicos en las carpetas correspondientes, se procede a realizar una limpieza de artículos repetidos mediante la comparación de cada ID almacenado en el archivo, y otra con relación al tamaño de cada archivo, en donde se eliminaron todos aquellos artículos de mayor tamaño a 150Kbytes.

4 Ya establecidos los corpus de los dos grupos seleccionados se procede a realizar el procesamiento de lenguaje natural para el reconocimiento de las entidades **patologías.** Esto se realizó en primer lugar cargando los modelos adjuntando los directorios correspondientes en los que se encuentran Biobert [31] y MedCat [35].

5 Después de cargar los recursos para hacer procesamiento de lenguaje natural, la metodología de procesamiento constó de tres etapas: La primera, consistió en separar el *string* correspondiente al cuerpo en una lista, en donde cada elemento de esta lista corresponde a un párrafo del cuerpo del artículo. En la segunda etapa, se hizo una selección de los párrafos con una cantidad mayore a diez palabras. Finalmente, en la tercera etapa se eliminaron las "palabras comunes" o *stopwords* presentes en cada párrafo.

6 En la última etapa para realizar el reconocimiento de las entidades (NER) en el cuerpo de los artículos ya procesados, se utilizó el modelo pre-entrenado Biobert [31] de la forma descrita en el estado del arte de este trabajo, seleccionando así todas aquellas entidades (palabras o conjunto de palabras) en la que el modelo las clasificaba como "DISEASE". Para finalizar se almacenan estas entidades en un DataFrame de la librería Pandas en conjunto con el número del problema AUGE y título.

7 El siguiente paso es hacer la iteración por las carpetas de los corpus de artículos recolectados para hacer reconocimiento de patologías relacionados con los problemas de las ges seleccionadas. Cada vez que se realizó NER en un artículo las entidades reconocidas se fueron almacenando en un archivo csv llamado "**pre-patologias.csv**".

8 Luego del reconocimiento de las entidades se utilizó la herramienta MedCat [35] en el archivo "**pre-patologias.csv**" para así obtener un código CUI asociado a cada entidad, seleccionando los códigos de los grupos semánticos *STY* ['T047'], ['T191'], ['T050'], pertenecientes a los grupos semánticos síndrome o enfermedad, proceso neoplásico y modelo experimental de enfermedad. Una vez relacionada la codificación del UMLS en las entidades encontradas por Biobert, se almacenan estos en un archivo *csv* llamado "**patologias encontradas.csv".**

9 Para finalizar este objetivo se ordenaron las entidades reconocidas según las co-ocurrencias detectadas en los distintos artículos, ordenándolas así de mayor a menor co.ocurrencia. Luego, para cada nombre de las distintas patologías se añadieron 3 expresiones de búsquedas distintas con operadores booleanos para conducir la búsqueda de otro corpus de artículos. Las expresiones de búsqueda añadidas son "AND (signs OR diagnosis)", " AND (tratment* OR therapeutic*)" y " AND ('medical device' OR 'medical devices' OR device*", las cuales se almacenaron en un archivo *csv* llamado "**expresiones de búsqueda-entidades.csv".**

4.3 OBJETIVO ESPECÍFICO 3: HACER BÚSQUEDAS DE ARTÍCULOS RELACIONADOS CON CADA UNA DE LAS PATOLOGÍAS ASOCIADAS A LOS PROBLEMAS GES DENTRO DE CADA GRUPO, PARA POSTERIORMENTE REALIZAR PROCESAMIENTO DE LENGUAJE NATURAL PARA ENCONTRAR SIGNOS, PROCEDIMIENTOS Y DISPOSITIVOS MÉDICOS ASOCIADOS A CADA PATOLOGÍA.

Para este objetivo específico, se utilizó una estrategia similar a la del objetivo específico anterior, diferenciándose principalmente en la codificación del nombre de cada archivo de texto, de manera que en esta etapa se integró el término de la patología con el que se hizo la búsqueda de artículos. La segunda diferencia que hay está en el reconocimiento de entidades, dado que para este objetivo se reconocieron las entidades signo, procedimiento y dispositivo médico.

1. Con los términos de búsqueda que contienen el nombre de la patología más las 3 expresiones complementarias, se comienza con la búsqueda haciendo una iteración por cada uno, haciendo una selección de diez artículos por cada iteración y almacenándolos en una carpeta llamada "*corpus signos procedimientos y dispositivos médicos*".

2. Después de la adquisición del corpus para entidades, se eliminaron los artículos duplicados y los de tamaño mayor a 100Kbytes.

3. Para el procesamiento de cada artículo, se utilizaron los mismos recursos del objetivo anterior, pero diferenciándose en la selección de los grupos semánticos seleccionados, tal que para signos se utilizaron los STY ['T033'], ['T046'], ['T020'], ['T190'], ['T049'], ['T019'], ['T184'], ['T046'], ['T201'], ['T043'], ['T037']. Para el reconocimiento de procedimientos, se seleccionaron los STY ['T060'], ['T058'], ['T063'], ['T062'], ['T061']. Y finalmente para seleccionar dispositivos médicos se utilizaron los STY ['T203'], ['T074'] y ['T075']. A cada entidad que se fue reconociendo se fue guardando su CUI en un archivo csv llamado "**pre-entidades.csv**".

4. El archivo csv del punto anterior se procesó para obtener las co-ocurrencias de los distintos CUIs y la información quedó almacenada en el archivo **"Entidades – signos procedimientos y dispositivos.csv**".

4.4 OBJETIVO ESPECÍFICO 4: GENERAR NODOS Y LAZOS PARA FORMAR LA RED DE INFORMACIÓN.

Una vez realizada la adquisición de los léxicos de cada corpus, estos se acondicionaron para posteriormente generar los distintos nodos y lazos. Dentro de este objetivo se propuso primero que los nombres de los distintos nodos sean los términos de las 4 distintas entidades en idioma español, y que estén complementados por información como el STY, CUI y la co-ocurrencia.

1. El primer paso de este objetivo fue normalizar la clasificación de las entidades, para esto se programó una función que clasifica los distintos STY de los CUI en 4 entidades relacionadas a patología, signo, procedimiento y dispositivo médico.

2. El segundo procedimiento fue unir el archivo "Entidades – signos procedimientos y dispositivos.csv" con la información en español de "mrconso.csv" mediante cada CUI que corresponde a las distintas entidades, seguido de este paso, se unió la información de "sty.csv" para obtener los grupos semántico de cada CUI. Una vez que se unió la información, se procede a añadir la clasificación entregada por la función del punto anterior para clasificar todo el léxico de palabras dentro las 4 entidades.

3. Una vez que se organizó la información, se procedió a generar los distintos nodos con la estructura de diccionario que soporta la librería Nertowrkx, esto se realizó seleccionando el nombre de cada entidad en español como nombre para el nodo y también la frecuencia, clasificación de entidad, CUI y STY como atributos correspondientes al nodo. Luego, en la misma línea se crearon los nodos referentes a los cinco problemas de salud del grupo seleccionado, quedando con el nombre del problema de salud y población objetivo, con sus respectivos atributos como el número del problema del plan AUGE, el CUI asociado y el grupo al que pertenecen.

4. Para la creación de los lazos entre los distintos nodos se realizaron dos iteraciones en los archivos 'Entidades-patologías.csv' y 'Entidades- signos procedimientos y dispositivos.csv', la primera tuvo el objetivo de encontrar lazos entre las patologías y los problemas de salud del plan AUGE cada vez que se encontraban en las filas del DataFrame. Por otro lado, con la segunda iteración se recogieron los lazos

al coincidir cada entidad (signo, procedimiento o dispositivo) con las patologías, como atributo de los distintos lazos, se utilizó el título de artículo correspondiente.

5. Una vez creada la red de información con todos sus nodos, lazos y atributos, se procedió a exportar el archivo **".gexf"** para ser almacenado en la memoria. Una vez almacenado, se importó en el software Gephi [42] para realizar visualizaciones y análisis.

6. En el software, los análisis realizados consistieron en hacer visualizaciones de la red seleccionando nodos específicos para ver con qué términos y entidades tenían relación, es así como se compararon relaciones entre **problemas-patologías**, **patologías-signos**, **patologías-procedimientos**, **patologías-dispositivos**.

4 Resultados

En esta sección se repasarán los resultados obtenidos en cada objetivo específico, cada análisis se hará conforme a volúmenes de datos recogidos y tipos de resultados obtenidos. , se analizará la red elaborada con algunos ejemplos con muestras de nodos. En anexos se encuentra una clasificación a los 85 problemas del plan AUGE en 13 grupos distintos, se seleccionó el grupo relacionado con patologías cardiacas, los problemas seleccionados fueron:

- 2. Cardiopatías congénitas operables en personas menores de 15 años.
- 5. Infarto agudo del miocardio.
- 21. Hipertensión arterial primaria o esencial en personas de 15 años y más.
- 25. Trastornos de generación del impulso y conducción en personas de 15 años y más, que requieren marcapaso.
- 74. Tratamiento quirúrgico de lesiones crónicas de la válvula aórtica en personas de 15 años y más.
- 79. Tratamiento quirúrgico de lesiones crónicas de las válvulas mitral y tricúspide en personas de 15 años y más.

4.1 Objetivo específico 1: Base de datos y codificación

Para cumplir con este objetivo se adquirió la base de datos del UMLS y se almacenó en el disco duro la información respectiva.

Resultado	Observación
Licencia UMLS	ANEXO 1
Instalación de UMLS	ANEXO 2
Directorio UMLS	ANEXO 3 2.108 Archivos Tamaño: 19.2 GB
Metathesauro UMLS	24 archivos formato ".RRF" Tamaño total: 13,3 GB • "MRCONSO.RRF" : 653 MBytes • "MRSTY.RRF": 81,3 MBytes

Tabla 4: Resultados de adquisición de UMLS.

Los problemas seleccionados para este trabajo fueron los pertenecientes al grupo uno, a los que se le asoció un CUI, criterio de búsqueda y población objetivo, cada problema se identifica con la columna 'N°'.

N°	CUI	Problema de salud	Criterio de búsqueda	Población objetivo	Grupo	Nombre
2	C0152021	CardiopatÃas congÃ©nitas menores de 15	Kids OR kid OR children OR child	NiÃ±ez	1	Problemas Cardiovasculares
5	C0155626	Infarto agudo de miocardio	NaN	Todas las edades	1	Problemas Cardiovasculares
21	C0085580	HipertensiÃ³n primaria	NaN	Adolescencia y adultez	1	Problemas Cardiovasculares
25	C0340914	alteraciÃ³n del sistema de marcapasos cardÃ-ac...	NaN	Adolescencia y adultez	1	Problemas Cardiovasculares
74	C1260873	valvulopatÃa aÃ³rtica	NaN	Adolescencia y adultez	1	Problemas Cardiovasculares
79	C0018824	valvulopatÃa	NaN	Adolescencia y adultez	1	Problemas Cardiovasculares

Tabla 5: Problemas seleccionados para realizar procesamiento de información.

Una vez que se seleccionaron los problemas, se obtuvieron ochenta y dos expresiones para la búsqueda de artículos y el directorio en los que se guardará el corpus, como se muestra en la tabla de a continuación.

Grupo	N°	cod	expresion	Directorio
1	2	_0	"Congenital heart disease" AND (Kids OR kid OR...	/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db...
1	2	_0_d	"Congenital heart disease" AND (Kids OR kid OR...	/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db...
1	2	_6	"Congenital heart disease NOS" AND (Kids OR ki...	/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db...
1	2	_6_d	"Congenital heart disease NOS" AND (Kids OR ki...	/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db...
1	2	_8	"Heart Diseases, Congenital" AND (Kids OR kid ...	/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db...
...	...	...	...	...
1	79	_158	"Cardiac valvulopathy"	/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db...
1	79	_158_d	"Cardiac valvulopathy" AND "related diseases"	/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db...
1	79	_160	"Heart valve disorder (disorder)"	/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db...
1	79	_160_d	"Heart valve disorder (disorder)" AND "related...	/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db.

Tabla 6: Expresiones de búsqueda para la adquisición de artículos relacionados con los problemas de las GES.

4.2 OBJETIVO ESPECÍFICO 2: OBTENCIÓN DE PATOLOGÍAS

Para el cumplimiento de este objetivo específico se realizó la adquisición de artículos mediante la iteración de las expresiones de búsqueda, para luego aplicar procesamiento de lenguaje natural a cada uno, obteniendo los siguientes resultados:

Resultado	Observación
Corpus patologías	Tamaño de carpeta: 7 MBytes Cantidad de artículos seleccionados: 238 Artículos eliminados por tamaño: 64 Artículos duplicados eliminados: 351
Pre-patologías.csv	Tamaño: 4,82 MBytes Cantidad de elementos: 36.586 filas
Entidades-patologías.csv	Tamaño: 795 KBytes Cantidad de elementos: 3.845 filas

Tabla 7: Resultados obtenidos de objetivo 2, corpus de patologías y archivos csv con información procesada.

A modo de clarificar el resultado Entidades-patologías.csv de este objetivo, se muestra una tabla con una muestra de la información reunida.

N°	title	PAT	CUI	freq	LAT	STR	STY	TUI	Entidad
5	Diagnostic value of copeptin combined with hypersensitive cardiac troponin T detection in early acute myocardial infarction	Acute myocardial infarction	C0155626	918.0	ENG	Acute myocardial infarction	Disease or Syndrome	T047	PatologÃa
79	Survival of people with valvular heart disease in a large, English community-based cohort study	hypertension	C0020538	894.0	ENG	Hypertension	Disease or Syndrome	T047	PatologÃa
79	Survival of people with valvular heart disease in a large, English community-based cohort study	Myocardial infarction	C0027051	284.0	ENG	Myocardial Infarction	Disease or Syndrome	T047	PatologÃa
…	…	…	…	…	…	…	…	…	…
79	COVID-19 increases the risk for the onset of atrial fibrillation in hospitalized patients	myocardial infarction	C0027051	284.0	ENG	Myocardial Infarction	Disease or Syndrome	T047	PatologÃa
79	Risk of cardiac valvulopathy with use of bisphosphonates: a population-based, multi-country case-control study	aortic stenosis	C0003507	118.0	ENG	Aortic Valve Stenosis	Disease or Syndrome	T047	PatologÃa
79	Andrographolide, a New Hope in the Prevention and Treatment of Metabolic Syndrome	atherosclerosis	C0004153	103.0	ENG	Atherosclerosis	Disease or Syndrome	T047	PatologÃa
2	Cardiovascular risk in children: a burden for future generations	atherosclerosis	C0004153	103.0	ENG	Atherosclerosis	Disease or Syndrome	T047	PatologÃa
5	Abacavir Use and Risk for Myocardial Infarction and Cardiovascular Events: Pooled Analysis of Data From Clinical Trials	dyslipidemia	C0242339	40.0	ENG	Dyslipidemias	Disease or Syndrome	T047	PatologÃa
79	Incidence and Predictors of Heart Failure in Patients With Atrial Fibrillation	peripheral - artery disease	C1704436	38.0	ENG	Peripheral Arterial Disease	Disease or Syndrome	T047	PatologÃa

Tabla 8: Muestra de patologías encontradas en procesamiento de artículos relacionados con problemas de las GES, PAT representa al término detectado por método de NER y STR representa el nombre dado por UMLS al código CUI asociado..

Finalmente, este objetivo específico se concluye al obtener las expresiones para la búsqueda de artículos relacionados a cada patología, almacenadas en el archivo Expresiones de búsqueda-patologías.csv. A continuación se ejemplifica una muestra de este archivo.

STR	CUI	ExpresiÃ³n
Acute myocardial infarction	C0155626	'Acute myocardial infarction' AND (signs OR diagnosis)
Acute myocardial infarction	C0155626	'Acute myocardial infarction' AND (tratment* OR therapeutic*)
Acute myocardial infarction	C0155626	'Acute myocardial infarction' AND ('medical device' OR 'medical devices' OR device*
hypertension	C0020538	'hypertension' AND (signs OR diagnosis)
hypertension	C0020538	'hypertension' AND (tratment* OR therapeutic*)
...	...	...
obesity	C0028754	'obesity' AND (signs OR diagnosis)
obesity	C0028754	'obesity' AND (tratment* OR therapeutic*)
obesity	C0028754	'obesity' AND ('medical device' OR 'medical devices' OR device*

Tabla 9: Muestra de expresiones para búsqueda de artículos relacionados a las distintas patologías detectadas.

4.3 OBJETIVO ESPECÍFICO 3: OBTENCIÓN DE SIGNOS, PROCEDIMIENTOS Y DISPOSITIVOS

En este objetivo específico se realizó la adquisición de corpus y procesamiento de lenguaje natural para el reconocimiento de signos, procedimientos y dispositivos médicos.

Resultado	Observaciones
Corpus signos, procedimientos y dispositivos	Tamaño de carpeta: 7 MBytes Cantidad de artículos seleccionados: 266 Artículos eliminados por tamaño: 188 Artículos duplicados eliminados: 591
Pre-entidades.csv	Tamaño: 10,5 MBytes Cantidad de elementos: 78.458 filas
Entidades signos, procedimientos y dispositivos.csv	Tamaño: 3,71 MBytes Cantidad de elementos: 18.004 filas

Tabla 10: Resultados obtenidos de objetivo 2, un corpus de artículos para encontrar signos, procedimientos y dispositivos, los que se fueron almacenando en 'pre-entidades.csv'. En el archivo 'Entidades signos procedimientos y dispositivos médicos.csv' se almacenaron las co-ocurrencias de cada entidad.

Las entidades recolectadas a partir del procesamiento del corpus se expresaron con la siguiente estructura de datos:

PAT	title	CUI	freq	LAT	STR	STY	TUI	Entidad
atherosclerosis	Aurora Borealis in dentistry: The applications of cold plasma in biomedicine	C0332461	7	ENG	Plaque	Finding	T033	Signo
cardiovascular diseases	Retinal Vascular Signs as Screening and Prognostic Factors for Chronic Kidney Disease: A Systematic Review and Meta-Analysis of Current Evidence	C3854333	16	ENG	Narrowing	Anatomical Abnormality	T190	Signo
heart failure	The role of informal carers in the diagnostic process of heart failure: a secondary qualitative analysis	C0543467	74	ENG	Surgical Procedures, Operative	Therapeutic or Preventive Procedure	T061	Procedimiento
...	...	...	...	...	...	...	...	...
atherosclerosis	Aurora Borealis in dentistry: The applications of cold plasma in biomedicine	C0220825	79	ENG	Evaluation	Health Care Activity	T058	Procedimiento
valvular stenosis regurgitation	Application of Internet of Medical/Health Things to Decentralized Clinical Trials: Development Status and Regulatory Considerations	C0162589	38	ENG	Defibrillators, Implantable	Medical Device	T074	Dispositivo mÃ©dico
valvular stenosis regurgitation	Application of Internet of Medical/Health Things to Decentralized Clinical Trials: Development Status and Regulatory Considerations	C0030163	2	ENG	Pacemaker, Artificial	Medical Device	T074	Dispositivo mÃ©dico

Tabla 11: Muestra de resultados finales de reconocimiento de signos, procedimientos y dispositivos en corpus, se observa que PAT es la patología con la que se buscaron artículos, freq es la co-ocurrencia del término, STR representa a cada término y STY, TUI y Entidad representan a la clasificación semántica de cada palabra.

4.4 OBJETIVO ESPECÍFICO 4: RED DE INFORMACIÓN.

Finalmente, una vez que se reconocieron y obtuvieron las entidades, se organizó la información para la creación del grafo, teniendo como resultados una red con los siguientes parámetros.

Red de información no dirigida • **Cantidad de nodos totales: 2097** • **Cantidad de lazos totales: 10267**	
Problemas de salud	6 nodos
Patologías	124 nodos
Signos	721 nodos
Procedimientos	1126 nodos
Dispositivos	103 nodos

Tabla 12: Características de la red configurada.

Luego de ser exportado al software Gephi, se logró la visualización completa de la red. Es así como, con una distribución tipo Yifan Hu (seleccionada dentro del software), se obtuvo la siguiente imagen:

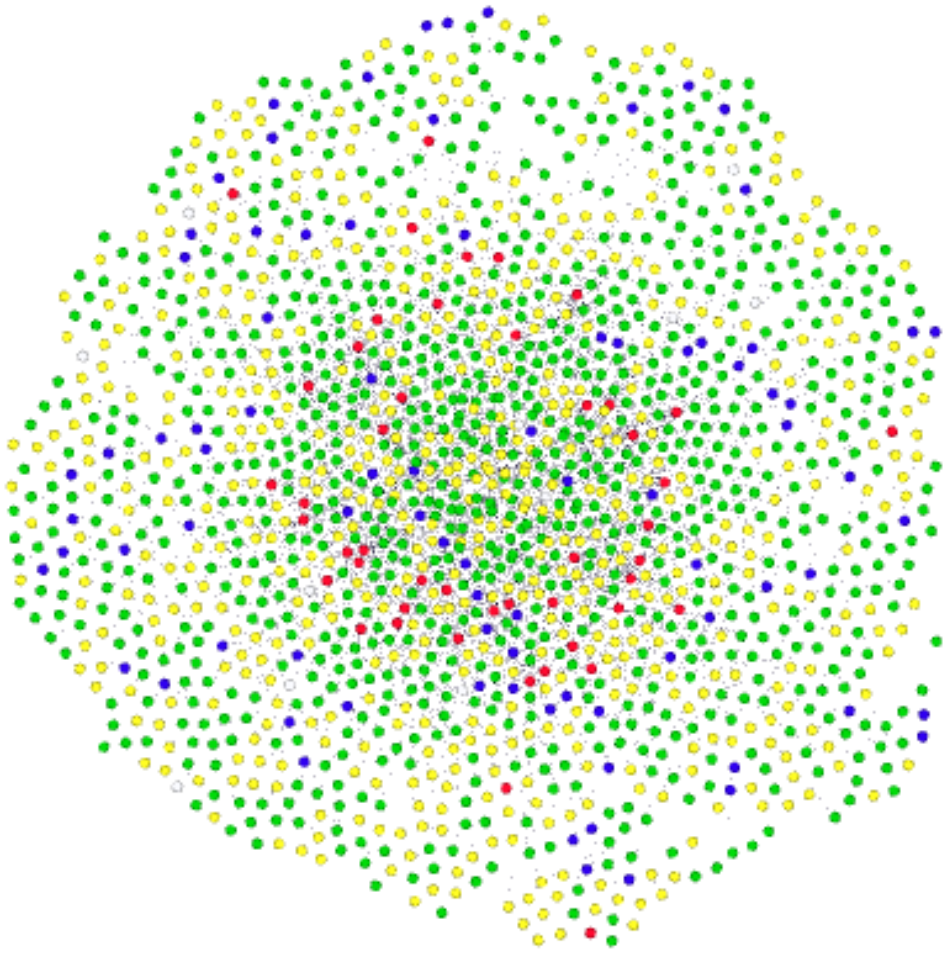


Ilustración 7: Visualización completa del grafo de información, en donde se pueden visualizar nodos de color rojo que corresponden a patologías, nodos de color amarillo que corresponden a signos, procedimientos de color verde y dispositivos de color azul.

Los análisis realizados en la red con el software se basaron principalmente en observar cómo los nodos se relacionan con sus pares. Bajo esta lógica, se analizaron primero como los nodos respectivos a los 5 problemas de salud del plan AUGE se relacionan con las patologías asociadas.

Los problemas seleccionados para este trabajo fueron:

- 2. Cardiopatías congénitas operables en personas menores de 15 años.
- 5. Infarto agudo del miocardio.
- 21. Hipertensión arterial primaria o esencial en personas de 15 años y más.
- 25. Trastornos de generación del impulso y conducción en personas de 15 años y más, que requieren marcapaso.
- 74. Tratamiento quirúrgico de lesiones crónicas de la válvula aórtica en personas de 15 años y más.
- 79. Tratamiento quirúrgico de lesiones crónicas de las válvulas mitral y tricúspide en personas de 15 años y más.

Por lo que aplicados los filtros correspondientes en el software, se generaron imágenes con los nombres de las entidades relacionadas y con el color de la entidad correspondiente, para este caso, serán grises las patologías y de color los problemas de salud del plan AUGE.

Ilustración 8: Visualización de patologías y los distintos problemas de las ges seleccionados.

Para obtener una mejor visualización, a continuación se muestra una captura de los nodos relacionados entre el problema de salud 21-Hipertensión arterial primaria o esencial en personas de 15 años y más y patologías.

Ilustración 9: Visualización de patologías y problema 21 de salud.

Por último, se mostrarán capturas de la red en las que se seleccionaron relaciones entre patologías y las tres otras entidades, signos, procedimientos y dispositivos.

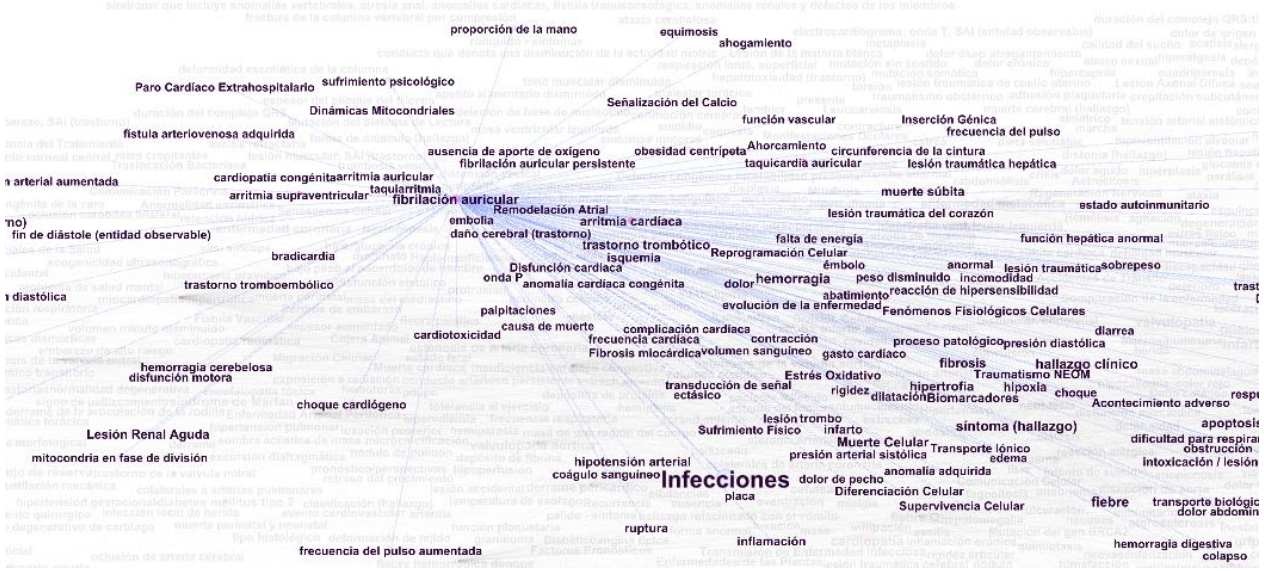


Ilustración 10:Visualización de red con la selección de patología Fibrilación Auricular y los signos relacionados

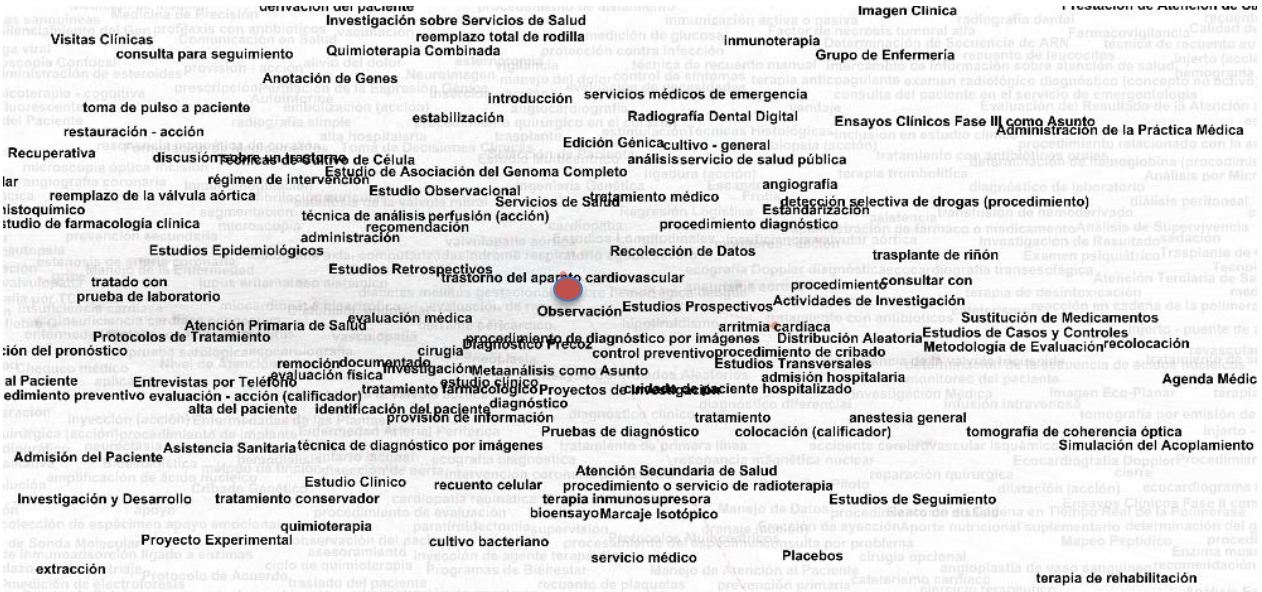


Ilustración 11: Visualización de red al seleccionar la patología Trastorno del aparato cardiovascular y los procedimientos relacionados

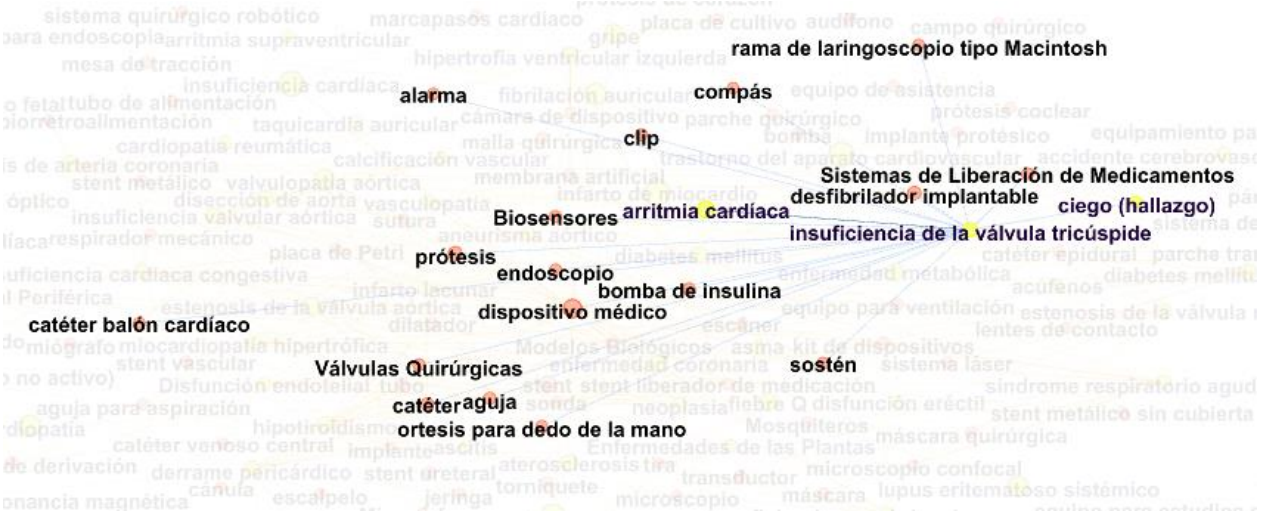


Ilustración 12: Visualización de patología Insuficiencia de válvula tricúspide y Dispositivos médicos asociados.

5 Discusión

En las secciones anteriores se realizó una metodología en la que se utilizaron distintas estrategias de "*DataScience",* soportándose principalmente mediante el uso de DataFrame y las distintas funciones de la librería Pandas. No obstante, en el transcurso del desarrollo e investigación de este trabajo, se fueron necesitando más herramientas para procesar información, ya sea para realizar la adquisición de los cuerpos de los artículos científicos o para el procesamiento de lenguaje natural, es por esto que se tuvo que recurrir a modelos existentes que optimizaran las tareas a realizar. En este sentido la búsqueda de modelos pre-entrenados del sitio HuggingFace culminó con el modelo BIOBERT, con el cual se pudo lograr realizar tareas de reconocimiento de entidades sin tener la necesidad de entrenar y diseñar métodos de inteligencia artificial. Sin embargo, para el reconocimiento de entidades hubo limitaciones para cumplir con los objetivos, puesto que los modelos para NER disponibles en HuggingFace solo permitían el reconocimiento de enfermedades y patologías, por este motivo, se tuvo que seguir buscando herramientas en la web para reconocer signos, procedimientos y dispositivos, resultando la selección del recurso MedCat porque, al reconocer entidades en texto, entrega codificación del UMLS.

Otro aspecto importante es que el entorno de hardware utilizado permitió realizar estas tareas de programación de forma óptima, mas no fue un entorno preparado para la eficiencia de las actividades, esto en el sentido de que para procesar grandes volúmenes de información hoy en día se utilizan GPUs que son capaces de procesar matrices completas de información y tarjetas SSD para el almacenamiento de datos, en cambio, para este trabajo se utilizó un procesador Intel I7 9na generación, SO Ubuntu Linux almacenado en tarjeta SSD de 100GB y almacenamiento de información en disco duro de 1T.

Para finalizar, es importante destacar que los resultados obtenidos no se abstienen de la existencia de ruido--información que no aporta o no fue deseada en la visualización de los nodos de la red-dado que hubo aspectos de la metodología que no fueron eficaces. El primer aspecto está en el almacenamiento de los artículos y eliminación de duplicados ya que, si bien volver a procesar dos veces la misma información no es eficiente, tampoco se aplicaron estrategias para compensar el sesgo que se generó al eliminar artículos que generaban relaciones entre las entidades detectadas en su contenido y la entidad que se registraba en el nombre del archivo del artículo. Es decir, por ejemplo al adquirir 10 artículos del problema 1, los archivos quedan grabados con el 1 en su nombre, si en la búsqueda del problema 4 se encontraron otros 10 artículos y habían algunos repetidos, al eliminarlos no se pudieron detectar relaciones entre las entidades de los artículos y el problema 4. Una forma de solución hubiese sido detectar duplicados y reunir los términos con los que se encontró el mismo artículo, para luego detectar entidades y relacionarlas a este conjunto. Por otro lado, en la selección de entidades para generar los nodos de la red se descartaron aquellas que tenían una co-ocurrencia menor al primer cuartil de coocurrencias, esto para disminuir el volumen de información manejado en el grafo, sin embargo, en otra aplicación este criterio puede afectar a la tarea que se quiera realizar, como por ejemplo para la detección de patologías raras o poco frecuentes.

Esta metodología tiene desafíos que pueden ser mejorables y que sin duda pueden seguir desarrollándose, esto considerando que el DataScience e inteligencia artificial están en la vanguardia en desarrollos informáticos. Para la salud pública, por ejemplo, se podrían desarrollar mejores métodos de reconocimiento de entidades en español para crear grafos a partir del procesamiento de millones registros clínicos electrónicos, pudiéndose investigar de esta forma relaciones entre patologías o tener controles de existencias de dispositivos médicos.

6 Conclusión

Los análisis de las redes de información permiten concluir que para el manejo, visualización y acondicionamiento de estas es necesario utilizar software de soporte externos al lenguaje de programación o método con el que se obtuvo, ya que, al menos Gephi, presenta una interfaz bastante intuitiva para disponer de colores, etiquetas, establecer distribuciones o para filtrar elementos. Las herramientas entregadas por Networkx, en cambio, si bien ofrecen vínculo con matplotlib y permiten generar gráficas de manera sencilla, esta tarea se vuelve más compleja al manipular grandes volúmenes de información .

En la visualización y análisis de la red ejemplificadas en la sección final de los resultados se puede notar que existe ruido presente como nodos relacionados como entidades, este ruido se hace más presente en las visualizaciones con mayores cantidad de nodos, como es el caso de la ilustración 10 y 11, ya que muestran relaciones entre nodos que quizás para fines prácticos no tienen relación alguna, como por ejemplo la relación entre la patología Trastorno del aparato cardiovascular y el procedimiento Radiografía dental digital.

El análisis de los problemas de salud del plan AUGE y las patologías relacionadas mediante la búsqueda de artículos científicos sí entrega una visión clara de lo que puede ser un "contexto" o "entorno" biomédico del mismo problema, en la ilustración 9 se puede notar directa relación entre el nombre del problema "Hipertensión(..)" y la patología con más co-ocurrencias "Hipertensión", y así mismo analizar los distintos nodos. Este resultado se vio favorecido también por haber agrupado los distintos problemas según su relación biomédica, así no se rellenó de sobremanera la red con nodos que no tenían relación entre sí.

Para finalizar, se concluye que la metodología desarrollada permitió crear una red con entidades coherentes y legibles, la utilización del UMLS como base de datos y terminologías permitió tener control de las entidades detectadas, permitiendo hasta cambiar de idioma para mejorar la visualización. El método de procesamiento de lenguaje natural permitió el reconocimiento de entidades coherentes y con existencia de ruido y, finalmente, utilizar estructura de redes abre paso a la investigación y desarrollo de aplicaciones más complejas para la biomédica. Además, la utilización de redes puede aportar en áreas gubernamentales, epidemiológicas o como soporte para las decisiones clínicas.

REFERENCIAS

[1] D. G. B. S. y D. G. Valdivia, «REFORMA DE SALUD EN CHILE; EL PLAN AUGE O RÉGIMEN DE GARANTÍAS EXPLÍCITAS EN SALUD (GES). SU ORIGEN Y EVOLUCIÓN,» *Boletín escuela de medicina UC,* 2007.

[2] M. d. salud, Plan Estratégico de tecnologías de información en salud [e-salud], 2011-2020.

[3] M. d. salud, «Estrategia e-Salud,» 2014.

[4] M. D. SALUD y S. D. S. PÚBLICA, *REGULA LOS DERECHOS Y DEBERES QUE TIENEN LAS PERSONAS EN RELACIÓN CON ACCIONES VINCULADAS A SU ATENCIÓN EN SALUD,* Congreso nacional de Chile, 2021.

[5] C. Carmona Santander, *Protección de datos personales Ley No. 19.628,* Congreso nacional de Chile, 2000.

[6] G. L. Adamo, «salud-e.cl,» 03 08 2016. [En línea]. Available: http://www.salud-e.cl/wp-content/uploads/2016/08/RCE-y-Acreditaci%C3%B3n.pdf. [Último acceso: 01 06 2021].

[7] H. Kang, L. Xia, F. Yan, Z. Wan, F. Shi, H. Yuan, H. Jiang, D. Wu, H. Sui, C. Zhang y D. Shen, «Diagnosis of Coronavirus Disease 2019 (COVID-19) With Structured Latent Multi-View

Representation Learning,» *IEEE TRANSACTIONS ON MEDICAL IMAGING,* vol. 39, nº 8, pp. 2606-2614, 2020.

[8] «Departamento de epidemiología - minsal,» [En línea]. Available: http://epi.minsal.cl/objetivos-estrategicos-y-funciones/. [Último acceso: 04 12 2021].

[9] D. B. Fridsma, «Data Sciences and Informatics: What's in a name?,» *Journal of the American Medical Informatics Association,* vol. 25, nº 1, p. 109, 2018.

[10] X. Yue, Z. Wang, J. Huang, S. Parthsarathy, Y. H. Soheil Moosavinasab, Y. Huang, S. M. Lin, W. Zhang y P. Zhang, «Graph embedding on biomedical networks: methods, aplications and evaluations,» *Bioinformatics,* vol. 36, nº 4, pp. 1241-1251, 2020.

[11] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals y G. E. Dahl, «Neural Message Passing for Quantum Chemistry,» de *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[12] H. P, Y. P, Z. P y e. al, «GCN-MF: disese-gene association identification by graph convolutional networks and matrix factorization,» de *proceedings of de 25th ACM SIGKDD international conferene on knowledge discovery and data mining association for computing machiney*, Anchorage, AK, USA, 2019.

[13] Z. N, W. RSN, Y. Y y e. al, «Deep mining heterogeneous networks of biomedical linked data to predict novel drug.target associations,» *Bioinfomatics,* vol. 33, pp. 2337- 2344, 2021.

[14] M. Rotmensch, Y. Halpern, A. Tilmat y e. al, «Leveranging a health knowledge graph from medical records,» *Sci reports,* vol. 7, p. 5994, 2017.

[15] N. l. o. m. (US), *UMLS reference manual,* 2021.

[16] B. Shickel, P. J. Tighe, A. Bihorac y P. Rashidi, «Deep EHR: A Survey of Recent Advances in Deep Learning Techniques for Electronic Health Record (EHR) Analysis,» *IEEE Journal of Biomedical and Health Informatics,* vol. 22, nº 5, pp. 1589-1604, 2018.

[17] I. Goodfellow, Y. Bengio y A. Courville, Deep Learning, MIT Press, 2016.

[18] X. Lv, Y. Guan, J. Yang y J. Wu, «Clinical Relation Extraction with Deep Learning,» *International Journal of Hybrid Information Technology,* vol. 9, nº 7, pp. 237-248, 2016.

[19] C. C. Aggarwal, Machine Learning for Text, Springer, Cham, 2018.

[20] A. Singhal, «Introducing the Knowledge Graph: things, not strings,» 16 Mayp 2012. [En línea]. Available: https://blog.google/products/search/introducing-knowledge-graph-things-not/. [Último acceso: 01 Julio 2022].

[21] Meta, «Meta for developers,» [En línea]. Available: https://developers.facebook.com/docs/graph-api. [Último acceso: 1 Julio 2022].

[22] M. Nickel, K. Murphy, V. Tresp y E. Gabrilovich, «A Review of Relational Machine Learning for Knowledge Graphs,» *Proceedings of the IEEE,* vol. 104, nº 1, pp. 11-33, 2016.

[23] T. Mikolov, K. Chen, G. Corrado y J. Dean, «Efficient estimation of word representations in vector space,» *Proceedings of the International Conference on Learning Representation,* pp. 1-12, 2013.

[24] M. Leimeister y B. J. Wilson, «Skip-gram word embeddings in hyperbolic space,» *Arxiv,* 2018.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser y I. Polosukhin, «Attention is All you Need,» *Advances in Neural Information Processing Systems,* vol. 30, pp. 5998-6008, 2017.

[26] D. Rothman, Transformers for Natural Language Processing, Birmingham B3 2PB, UK: Packt Publishing Ltd, 2021.

[27] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer y V. Stoyanov, «RoBERTa: A Robustly Optimized BERT Pretraining Approach,» *Arxiv,* 2019.

[28] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li y P. J. Liu, «Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer,» *Arxiv,* 2019.

[29] H. corporation. [En línea]. Available: https://huggingface.co/.

[30] HuggingFace, «Tokenizers documentation,» [En línea]. Available: https://huggingface.co/docs/tokenizers/index.

[31] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So y J. Kang, «BioBERT: a pre-trained biomedical language representation model for biomedical text mining,» *Bioinformatics,* vol. 36, nº 4, p. 1234–1240, 2020.

[32] J. Devlin, M.-W. Chang, K. Lee y K. Toutanova, «BERT: pre-training of deep bidirectional transformers for language understanding,» *Proceedings of NAACL-HLT,* pp. 4171-4186, 2019.

[33] E. Alsentzer, J. Murphy, W. Boag, W.-H. Weng, D. Jindi, T. Naumann y M. McDermott, «Publicly Available Clinical BERT Embeddings,» *Proceedings of the 2nd Clinical Natural Language Processing Workshop,* pp. 72-78, 2019.

[34] A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi y R. G. Mark, «MIMIC-III, a freely accessible critical care database,» *Scientific Data,* vol. 3, nº 160035, 2016.

[35] Z. Kraljevic, T. Searle, A. Shek, L. Roguski, K. Noor, D. Bean, A. Mascio, L. Zhu, A. A. Folarin, A. Roberts, R. Bendayan, M. P. Richardson y R. Stewart, «Multi-domain clinical natural language processing with {MedCAT}: The Medical Concept Annotation Toolkit,» *Artif. Intell. Med.,* vol. 117, p. 102083, 2021.

[36] S. Y. Yu, S. Rokka Chhetri, A. Canedo, P. Goyal y M. A. A. Faruque, «Pykg2vec: A Python Library for Knowledge Graph Embedding,» *arXiv preprint arXiv:1906.04239,* 2019.

[37] «Pykg2vec: Python Library for KGE Methods,» [En línea]. Available: https://pypi.org/project/pykg2vec/.

[38] F. Ilievski, D. Garijo, H. Chalupsky, N. T. Divvala, Y. Yao, C. Rogers, R. Li, J. Liu, A. Singh, D. Schwabe y P. Szekely, «KGTK: A Toolkit for Large Knowledge Graph Manipulation and Analysis,» *International Semantic Web Conference,* pp. 278--293, 2020.

[39] «KGTK documentation,» [En línea]. Available: https://kgtk.readthedocs.io/en/latest/.

[40] A. Hagberg, D. Schult y P. Swart, NetworkX Reference, 2022.

[41] N. developers, «Network analysis in python,» 2022. [En línea]. Available: https://networkx.org/.

[42] G. software, «The Open Graph Viz Platform,» NetBeans, [En línea]. Available: https://gephi.org/.

[43] H. E. Manoochehri y M. Nourani, «Proceedings of the 16th Annual MCBIOS Conference: bioinformatics,» *Proceedings of the 16th Annual MCBIOS Conference: bioinformatics,* vol. 21, nº 248, 2020.

[44] M. Kanehisa, M. Araki, S. Goto, M. Hattori, M. Hirakawa, M. Itoh, T. Katayama, S. Kawashima, S. Okuda, T. Tokimatsu y Y. Yamanishi, «KEGG for linking genomes to life and the environment,» *Nucleic Acids Research,* vol. 36, p. D480–D484, 2008.

[45] J. Dörpinghaus, A. Stefan, B. Schultz y M. Jacobs, «Towards context in large scale biomedical knowledge graphs,» *Arxiv,* 2020.

[46] S. G. Finlayson, P. LePendu y N. H. Shah, «Building the graph of medicine from millions of clinical narratives,» *Scientific Data,* vol. 1, nº 140032, 2014.

[47] N. Morley, «https://github.com/,» [En línea]. Available: https://github.com/cyclecycle/parse-pubmed. [Último acceso: 22 07 2022].

[48] B. v. 1.79, «Biopython library Documentation,» [En línea]. Available: https://biopython.org/. [Último acceso: 22 07 2022].

[49] G. Gibney y A. D. Baxevanis, «Searching NCBI Databases Using Entrez,» *Curr Protoc Hum Genet*, vol. 6, 2011.

[50] G. d. Chile, *Mensaje Presidencial, 21 de Mayo de 2000,* 2000.

[51] Y.-P. Chen, Y.-H. Lo, F. Lai y C.-H. Huang, « Disease Concept-Embedding Based on the Self-Supervised Method for Medical Information Extraction from Electronic Health Records and Disease Retrieval: Algorithm Development and Validation Study,» *J Med Internet Res* , vol. 23, 2021.

[52] T. Wu, Y. Wang, Y. Wang, E. Zhao y Y. Yuan, «Leveraging graph-based hierarchical medical entity embedding for healthcare applications,» *Scientific Reports volume*, vol. 11, nº 5858, 2021.

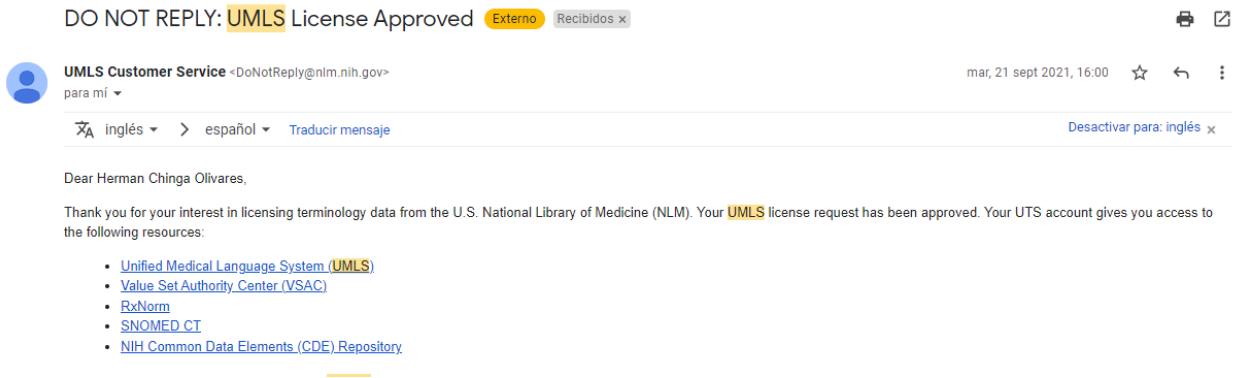
[53] L. Campillos-Llanos, «First Steps towards a Medical Lexicon for Spanish with Linguistic and Semantic Information,» *Proceedings of the 18th BioNLP Workshop and Shared Task*, pp. 152--164, Agosto 2019.

[54] «UMLS Database Query Diagrams: How to find all information associated with a particular atom (AUI value),» National library of medicine, [En línea]. Available: https://www.nlm.nih.gov/research/umls/implementation_resources/query_diagrams/er2.html. [Último acceso: 12 12 2021].

[55] «Auge 85,» Ministerio de salud, [En línea]. Available: https://auge.minsal.cl/problemasdesalud/lep. [Último acceso: 12 12 2021].

7 Anexos

ANEXO: LICENCIA DE NATIONAL LIBRARY OF MEDICINE PARA ADQUIRIR INFORMACIÓN E UMLS


DO NOT REPLY: UMLS License Approved Externo Recibidos ×

UMLS Customer Service <DoNotReply@nlm.nih.gov> mar, 21 sept 2021, 16:00

para mí

inglés › español Traducir mensaje Desactivar para: inglés ×

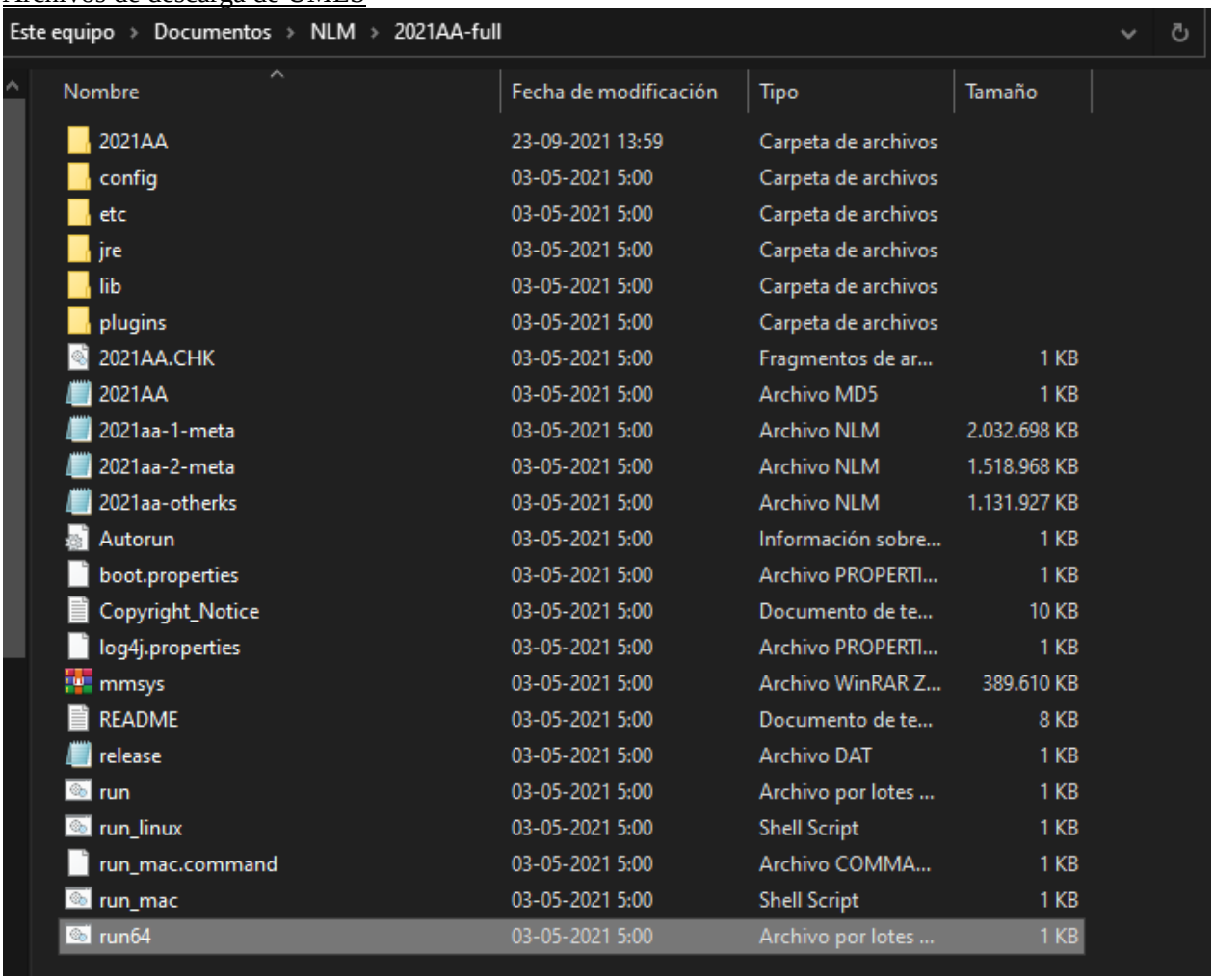
Dear Herman Chinga Olivares,

Thank you for your interest in licensing terminology data from the U.S. National Library of Medicine (NLM). Your UMLS license request has been approved. Your UTS account gives you access to the following resources:

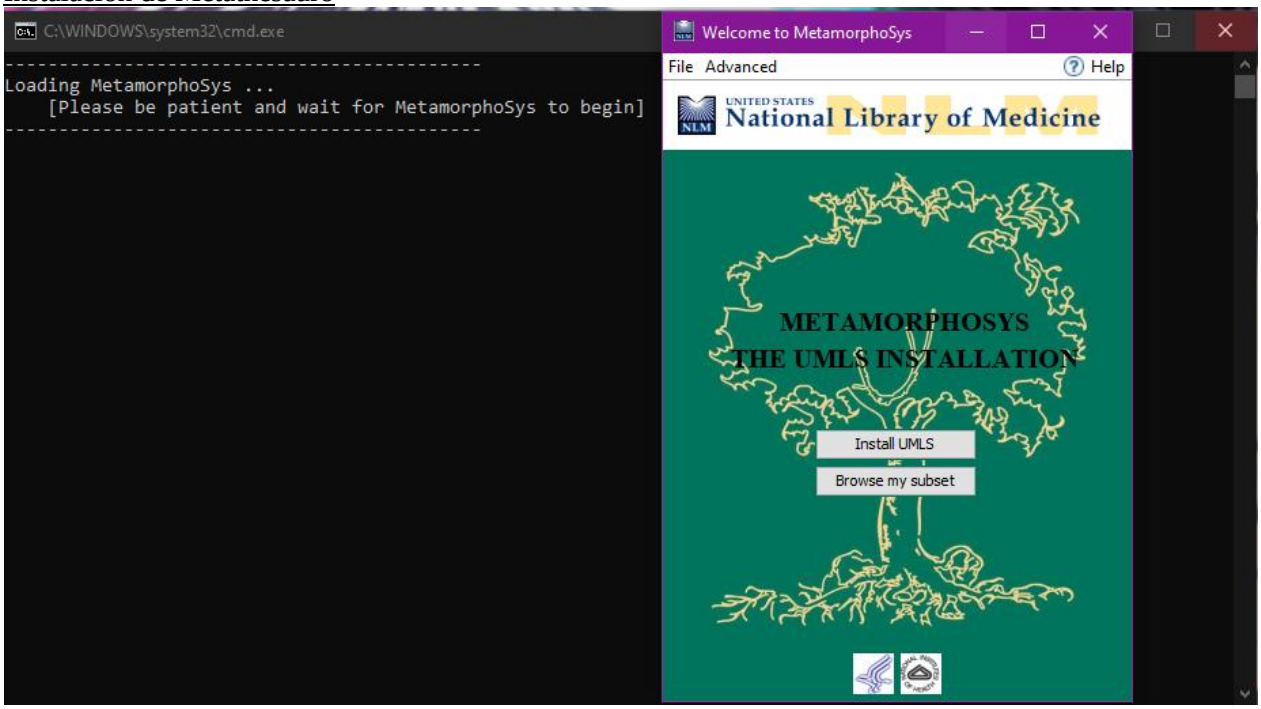
- Unified Medical Language System (UMLS)
- Value Set Authority Center (VSAC)
- RxNorm
- SNOMED CT
- NIH Common Data Elements (CDE) Repository

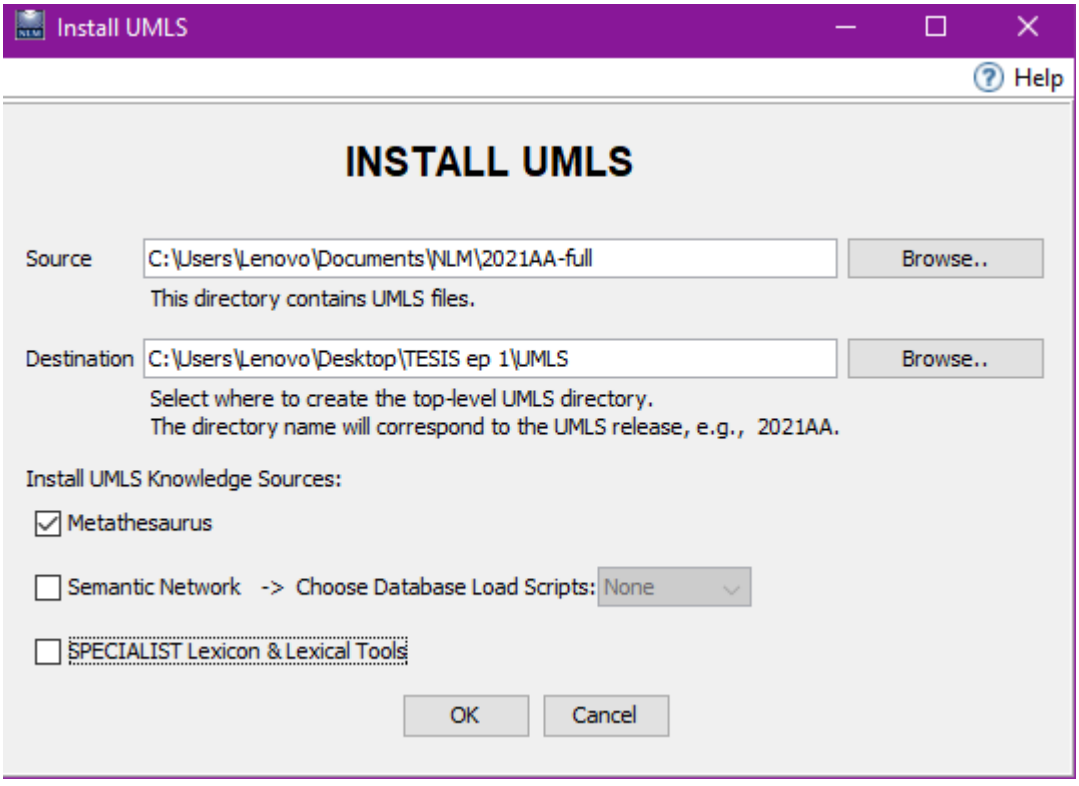
ANEXO: PROCESO DE INSTALACIÓN Y DESCARGA DE UMLS

Archivos de descarga de UMLS

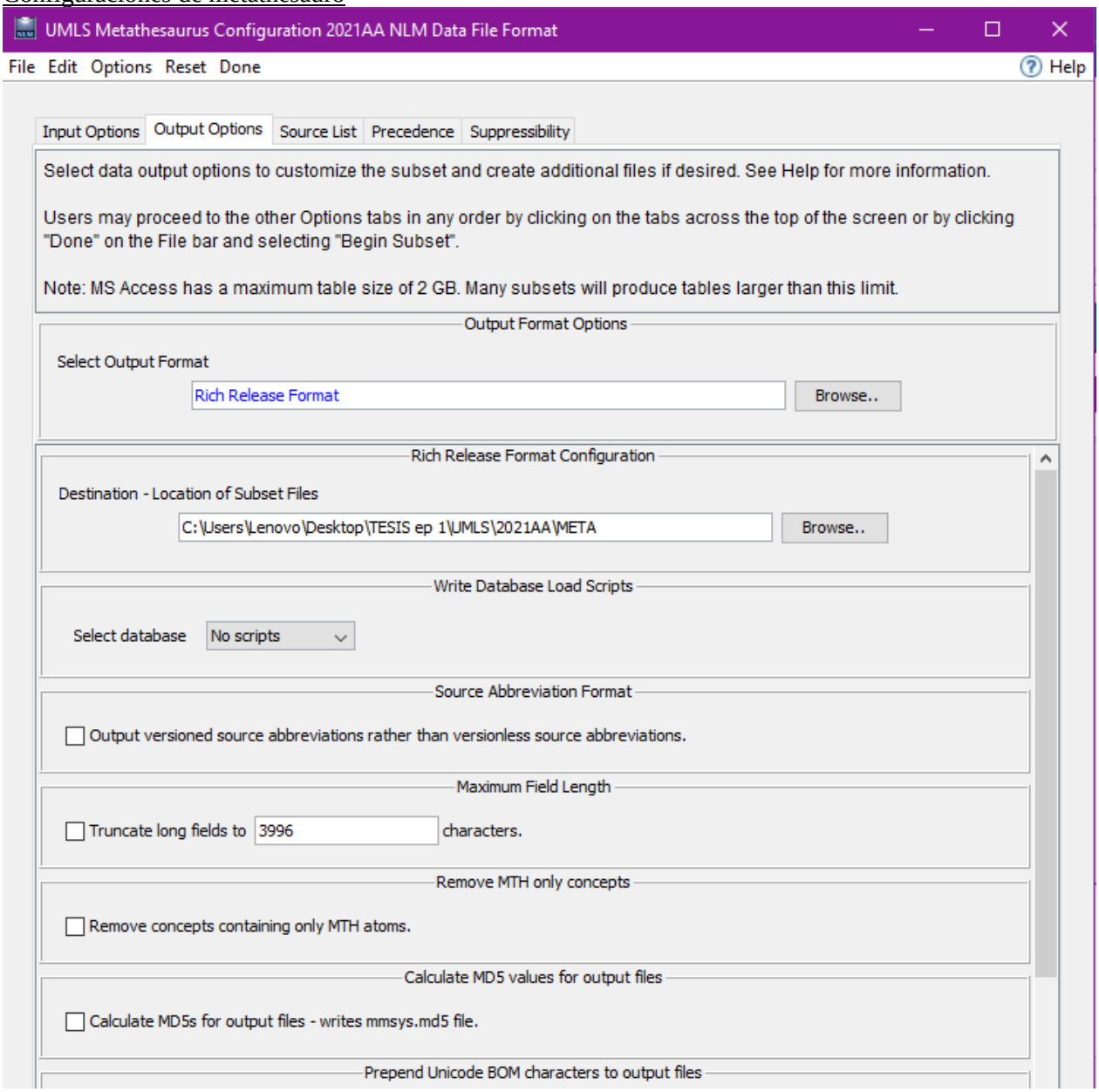


Instalación de Metathesauro

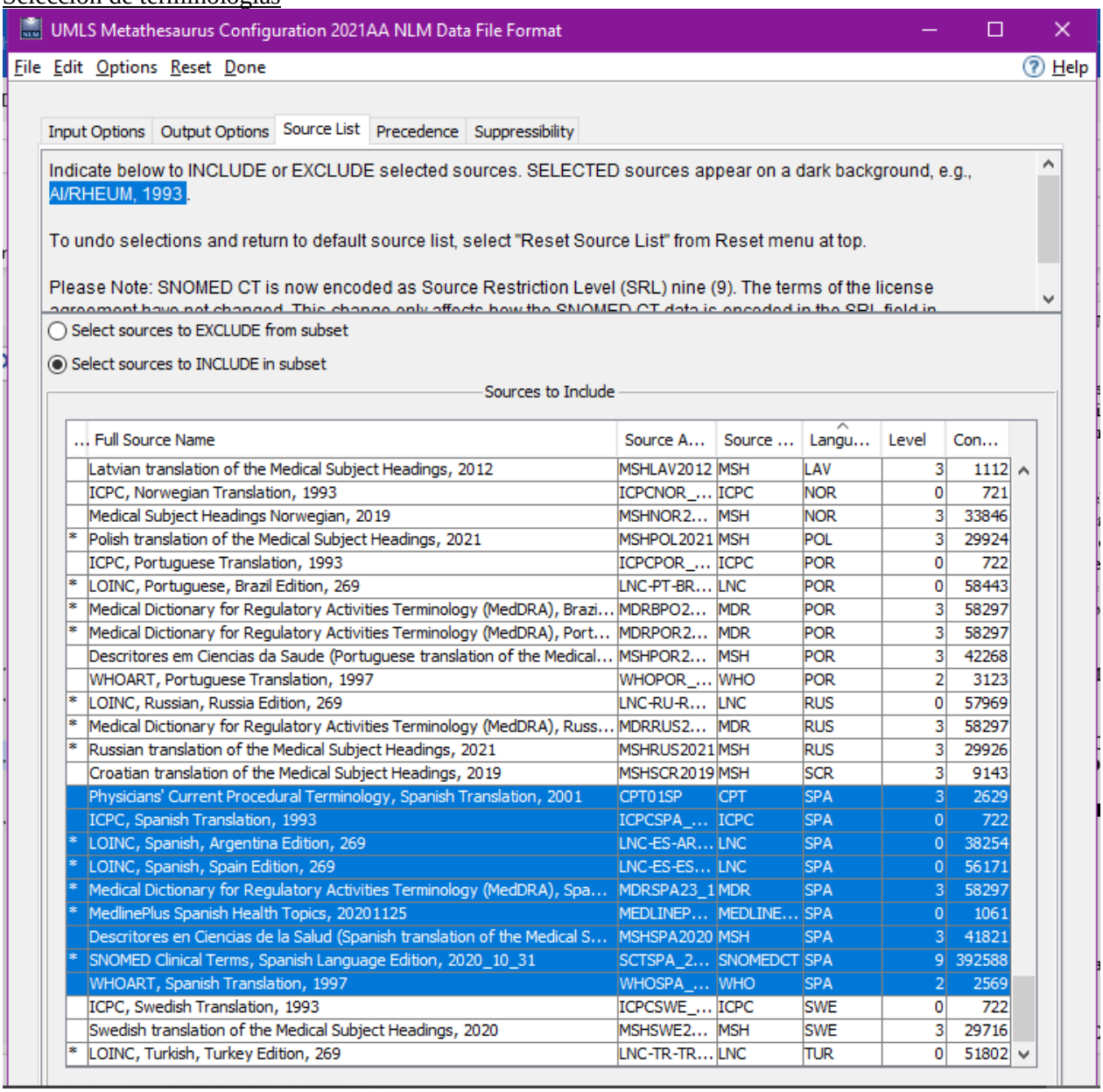




Configuraciones de metathesauro

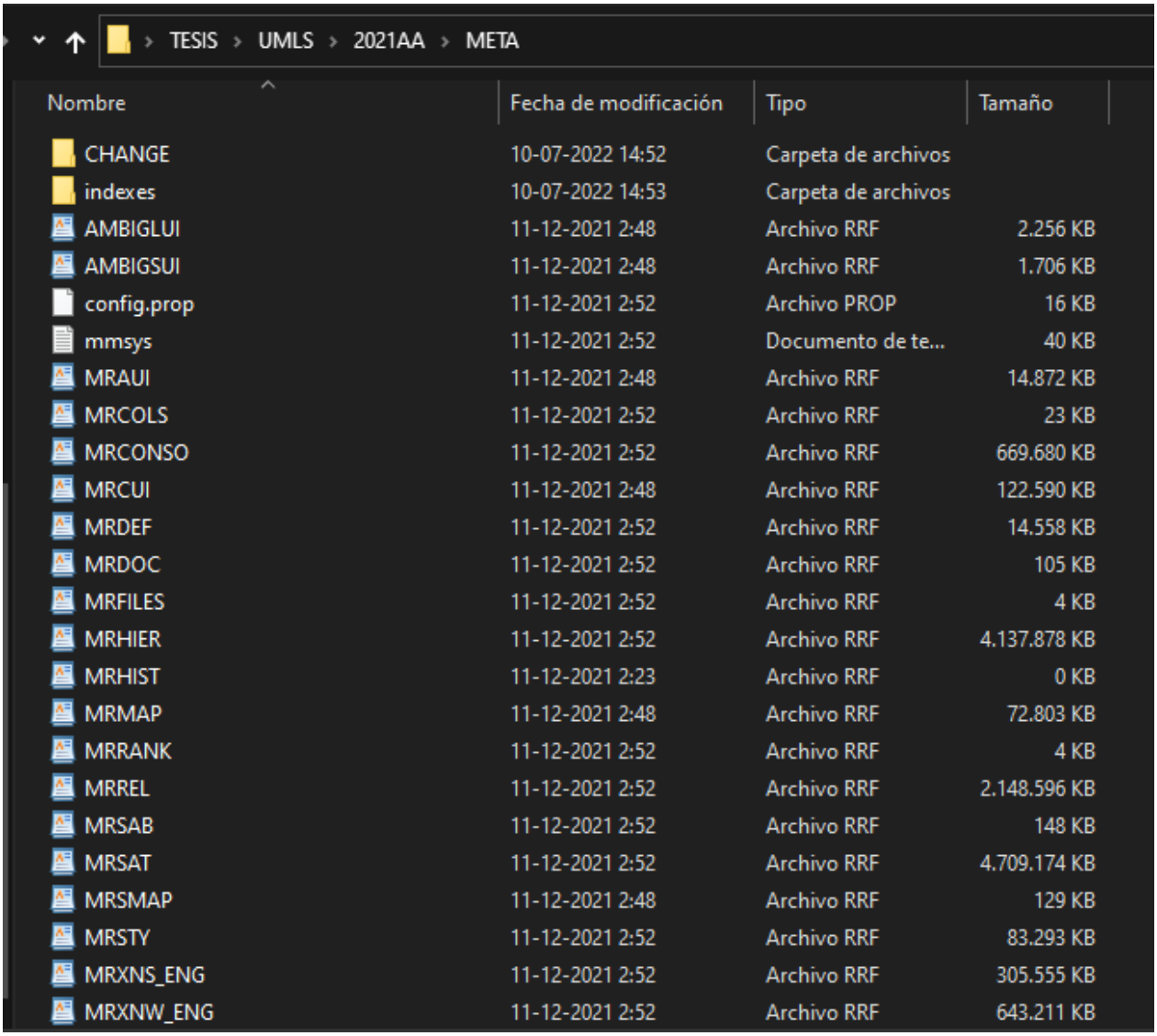


Selección de terminologías



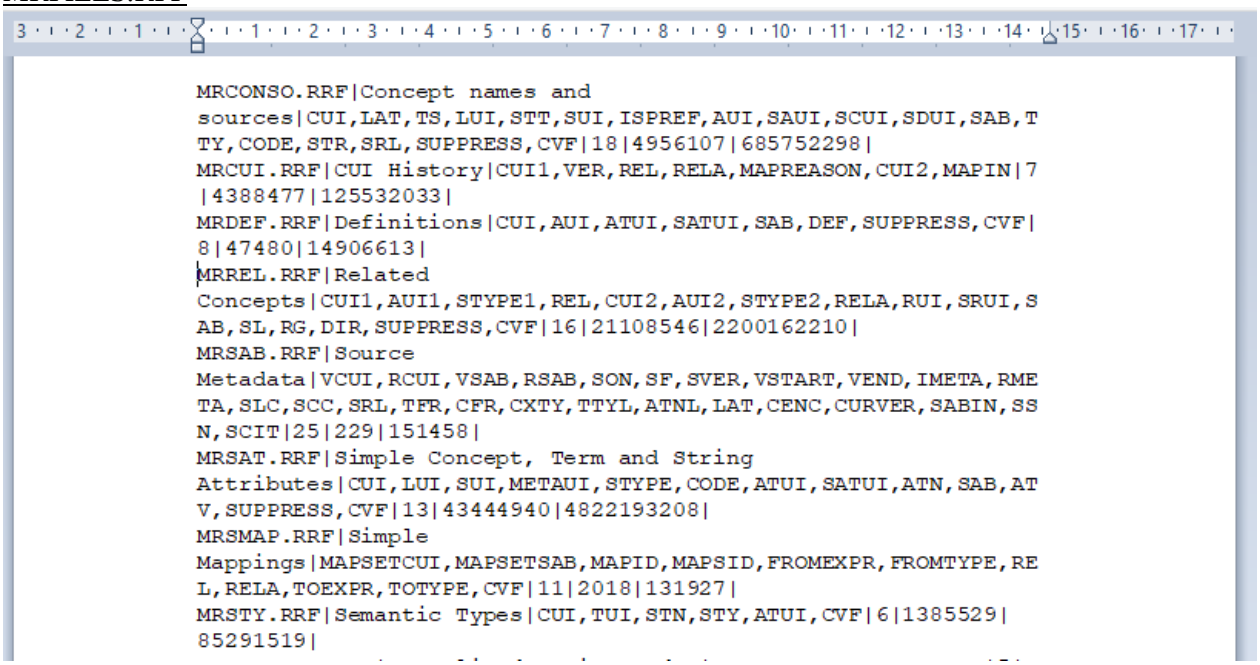
...	Full Source Name	Source A...	Source ...	Langu...	Level	Con...
	Latvian translation of the Medical Subject Headings, 2012	MSHLAV2012	MSH	LAV	3	1112
	ICPC, Norwegian Translation, 1993	ICPCNOR_...	ICPC	NOR	0	721
	Medical Subject Headings Norwegian, 2019	MSHNOR2...	MSH	NOR	3	33846
*	Polish translation of the Medical Subject Headings, 2021	MSHPOL2021	MSH	POL	3	29924
	ICPC, Portuguese Translation, 1993	ICPCPOR_...	ICPC	POR	0	722
*	LOINC, Portuguese, Brazil Edition, 269	LNC-PT-BR...	LNC	POR	0	58443
*	Medical Dictionary for Regulatory Activities Terminology (MedDRA), Brazi...	MDRBPO2...	MDR	POR	3	58297
*	Medical Dictionary for Regulatory Activities Terminology (MedDRA), Port...	MDRPOR2...	MDR	POR	3	58297
	Descritores em Ciencias da Saude (Portuguese translation of the Medical...	MSHPOR2...	MSH	POR	3	42268
	WHOART, Portuguese Translation, 1997	WHOPOR_...	WHO	POR	2	3123
*	LOINC, Russian, Russia Edition, 269	LNC-RU-R...	LNC	RUS	0	57969
*	Medical Dictionary for Regulatory Activities Terminology (MedDRA), Russ...	MDRRUS2...	MDR	RUS	3	58297
*	Russian translation of the Medical Subject Headings, 2021	MSHRUS2021	MSH	RUS	3	29926
	Croatian translation of the Medical Subject Headings, 2019	MSHSCR2019	MSH	SCR	3	9143
	Physicians' Current Procedural Terminology, Spanish Translation, 2001	CPT01SP	CPT	SPA	3	2629
	ICPC, Spanish Translation, 1993	ICPCSPA_...	ICPC	SPA	0	722
*	LOINC, Spanish, Argentina Edition, 269	LNC-ES-AR...	LNC	SPA	0	38254
*	LOINC, Spanish, Spain Edition, 269	LNC-ES-ES...	LNC	SPA	0	56171
*	Medical Dictionary for Regulatory Activities Terminology (MedDRA), Spa...	MDRSPA23_1	MDR	SPA	3	58297
*	MedlinePlus Spanish Health Topics, 20201125	MEDLINEP...	MEDLINE...	SPA	0	1061
	Descritores en Ciencias de la Salud (Spanish translation of the Medical S...	MSHSPA2020	MSH	SPA	3	41821
*	SNOMED Clinical Terms, Spanish Language Edition, 2020_10_31	SCTSPA_2...	SNOMEDCT	SPA	9	392588
	WHOART, Spanish Translation, 1997	WHOSPA_...	WHO	SPA	2	2569
	ICPC, Swedish Translation, 1993	ICPCSWE_...	ICPC	SWE	0	722
	Swedish translation of the Medical Subject Headings, 2020	MSHSWE2...	MSH	SWE	3	29716
*	LOINC, Turkish, Turkey Edition, 269	LNC-TR-TR...	LNC	TUR	0	51802

ANEXO CARPETA DE ALMACENAMIENTO DE UMLS



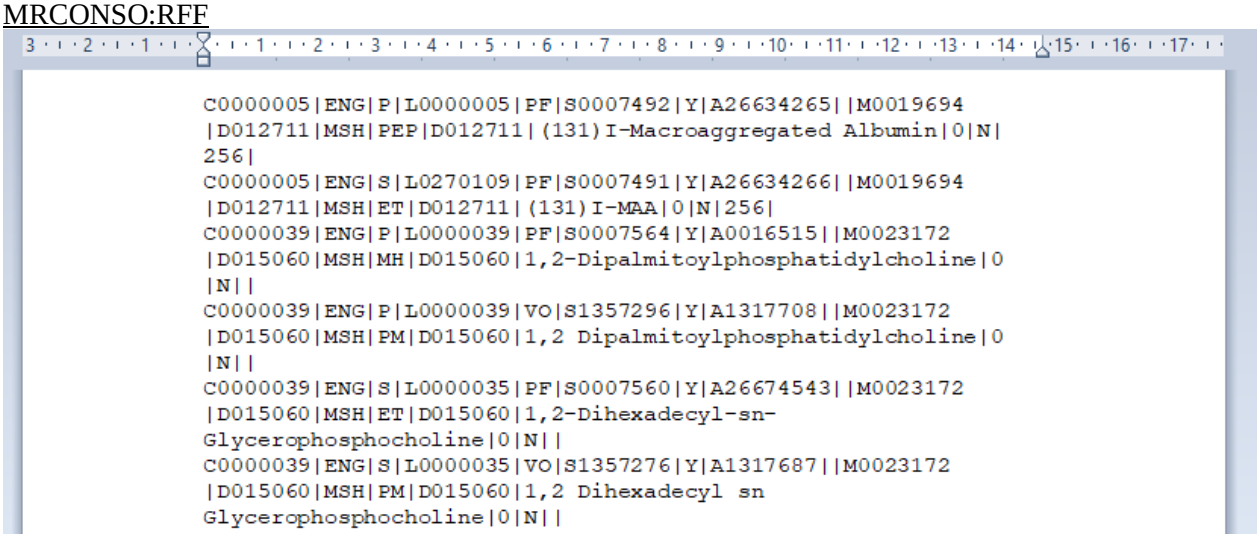
ANEXO ESTRUCTURA DE DATOS DE MRFILES.RFF, MRCONSO.RFF Y MRSTY.RFF

MRFILES.RFF



```
MRCONSO.RRF|Concept names and
sources|CUI,LAT,TS,LUI,STT,SUI,ISPREF,AUI,SAUI,SCUI,SDUI,SAB,T
TY,CODE,STR,SRL,SUPPRESS,CVF|18|4956107|685752298|
MRCUI.RRF|CUI History|CUI1,VER,REL,RELA,MAPREASON,CUI2,MAPIN|7
|4388477|125532033|
MRDEF.RRF|Definitions|CUI,AUI,ATUI,SATUI,SAB,DEF,SUPPRESS,CVF|
8|47480|14906613|
MRREL.RRF|Related
Concepts|CUI1,AUI1,STYPE1,REL,CUI2,AUI2,STYPE2,RELA,RUI,SRUI,S
AB,SL,RG,DIR,SUPPRESS,CVF|16|21108546|2200162210|
MRSAB.RRF|Source
Metadata|VCUI,RCUI,VSAB,RSAB,SON,SF,SVER,VSTART,VEND,IMETA,RME
TA,SLC,SCC,SRL,TFR,CFR,CXTY,TTYL,ATNL,LAT,CENC,CURVER,SABIN,SS
N,SCIT|25|229|151458|
MRSAT.RRF|Simple Concept, Term and String
Attributes|CUI,LUI,SUI,METAUI,STYPE,CODE,ATUI,SATUI,ATN,SAB,AT
V,SUPPRESS,CVF|13|43444940|4822193208|
MRSMAP.RRF|Simple
Mappings|MAPSETCUI,MAPSETSAB,MAPID,MAPSID,FROMEXPR,FROMTYPE,RE
L,RELA,TOEXPR,TOTYPE,CVF|11|2018|131927|
MRSTY.RRF|Semantic Types|CUI,TUI,STN,STY,ATUI,CVF|6|1385529|
85291519|
```

MRCONSO:RFF



```
C0000005|ENG|P|L0000005|PF|S0007492|Y|A26634265||M0019694
|D012711|MSH|PEP|D012711|(131)I-Macroaggregated Albumin|0|N|
256|
C0000005|ENG|S|L0270109|PF|S0007491|Y|A26634266||M0019694
|D012711|MSH|ET|D012711|(131)I-MAA|0|N|256|
C0000039|ENG|P|L0000039|PF|S0007564|Y|A0016515||M0023172
|D015060|MSH|MH|D015060|1,2-Dipalmitoylphosphatidylcholine|0
|N||
C0000039|ENG|P|L0000039|VO|S1357296|Y|A1317708||M0023172
|D015060|MSH|PM|D015060|1,2 Dipalmitoylphosphatidylcholine|0
|N||
C0000039|ENG|S|L0000035|PF|S0007560|Y|A26674543||M0023172
|D015060|MSH|ET|D015060|1,2-Dihexadecyl-sn-
Glycerophosphocholine|0|N||
C0000039|ENG|S|L0000035|VO|S1357276|Y|A1317687||M0023172
|D015060|MSH|PM|D015060|1,2 Dihexadecyl sn
Glycerophosphocholine|0|N||
```

MRSTY.RFF

```
C0000005|T116|A1.4.1.2.1.7|Amino Acid, Peptide, or
Protein|AT17648347|256|
C0000005|T121|A1.4.1.1.1|Pharmacologic Substance|AT17575038|
256|
C0000005|T130|A1.4.1.1.4|Indicator, Reagent, or Diagnostic
Aid|AT17634323|256|
C0000039|T109|A1.4.1.2.1|Organic Chemical|AT45562015|256|
C0000039|T121|A1.4.1.1.1|Pharmacologic Substance|AT17567371|
256|
C0000052|T116|A1.4.1.2.1.7|Amino Acid, Peptide, or
Protein|AT08381079|256|
C0000052|T126|A1.4.1.1.3.3|Enzyme|AT08775334|256|
C0000074|T109|A1.4.1.2.1|Organic Chemical|AT205422471|256|
C0000084|T116|A1.4.1.2.1.7|Amino Acid, Peptide, or
Protein|AT17641823|256|
C0000084|T123|A1.4.1.1.3|Biologically Active
Substance|AT17597318|256|
```

ANEXO EXTRACTO DE EXCEL DE LISTADO ESPECÍFICO DE PRESTACIONES DISPONIBLES EN WEB WWW.AUGE.CL

Problema de salud	Intervención sanitaria	Prestación o grupo de prestaciones	Glosa	Observaciones
Enfermedad renal crónica etapa 4 y 5	Diagnóstico	Confirmación retardo crecimiento óseo	Consulta integral de especialidades en medicina interna y subespecialidades, oftalmología, neurología, oncología (hospital alta complejidad)	
			Perfil bioquímico (determinación automatizada de 12 parámetros)	
			Tiroestimulante (TSH), hormona (adulto, niño o R.N.)	
			Tiroxina o tetrayodotironina (T4)	
			Triyodotironina (T3)	
			IGF1 o somatomedina - C (Insulin like growth factor)	
			IGFBP3, iGFBP1 (Insulin like growth factor binding proteins) c/u	
			Resonancia magnética cráneo encefálica u oídos, bilateral	
		Tratamiento con hormona de crecimiento en personas menores de 15 años	Consulta integral de especialidades en medicina interna y subespecialidades, oftalmología, neurología, oncología (hospital alta complejidad)	
			Perfil bioquímico (determinación automatizada de 12 parámetros)	
			Tiroestimulante (TSH), hormona (adulto, niño o R.N.)	
			IGF1 o somatomedina - C (Insulin like growth factor)	
			Radiografía edad ósea: carpo y mano	1 exposición
			Perfil lipídico (incluye: colesterol total, HDL, LDL, VLDL y triglicéridos)	
			Somatropina	Lápiz prellenado
		Tratamiento citomegalovirus alto riesgo	Ganciclovir	
			PCR en tiempo real	
			Valganciclovir	
		Tratamiento citomegalovirus bajo riesgo	Día cama hospitalización integral medicina, cirugía, pediatría, obstetricia-ginecología y especialidades (sala 3 camas o más) (hospital alta complejidad)	
			Ganciclovir	
			PCR en tiempo real	
			Valganciclovir	
			Consulta integral de especialidades en urología, otorrinolaringología, medicina física y rehabilitación, dermatología, pediatría y subespecialidades (en CDT)	
			Consulta integral de especialidades en cirugía, ginecología y obstetricia,	

LEP 2019 (2)

ANEXO: CLASIFICACIÓN DE PROBLEMAS GES

N°	CUI	Problema de salud	Criterio de búsqueda	Población objetivo	Grupo	Nombre
1	C0022661	insuficiencia renal cronica etapa 4 y 5		Todas las edades	3	Problemas por diabetes mellitus, renales

						y de cirugÃa general
2	C0152021	CardiopatÃas congÃ©nitas menores de 15	Kids OR kid OR children OR child	NiÃ±ez	1	Problemas Cardiovasculares
3	C0302592	Cancer cervicouterino		Personas gestantes	4	Problemas por cÃ¡ncer
4	C1304888	Cuidados paliativos y alivio del dolor		Todas las edades	4	Problemas por cÃ¡ncer
5	C0155626	Infarto agudo de miocardio		Todas las edades	1	Problemas Cardiovasculares
6	C0011854	Diabetes mellitus tipo 1		Todas las edades	3	Problemas por diabetes mellitus, renales y de cirugÃa general
7	C0011860	Diabetes mellitus tipo 2		Todas las edades	3	Problemas por diabetes mellitus, renales y de cirugÃa general
8	C0006142	CÃ¡ncer de mama		Adolescencia y adultez	4	Problemas por cÃ¡ncer
9	C0344479	disrafismo espinal		Todas las edades	5	Problemas por artropatÃas y ortopÃ©dicos
10	C0036439	Escoliosis		NiÃ±ez, adolescencia y adultez	5	Problemas por artropatÃas y ortopÃ©dicos
11	C1504458	CirugÃa de cataratas		Todas las edades	7	Problemas oftalmolÃ³gicos y otorrinonaringolÃ³gicos
12	C0186177	inserciÃ³n de prÃ³tesis de cadera, total (procedimiento)	elderly people OR elderly person OR old person OR old people	Adultos mayores	5	Problemas por artropatÃas y ortopÃ©dicos
13	C0158646	Trastornos por labio y paladar hendidos		Todas las edades	9	Problemas por enfermedades congÃ©nitas
14	C1301605	sospecha de cÃ¡ncer infantil	Kids OR kid OR children OR child	NiÃ±os y niÃ±as	4	Problemas por cÃ¡ncer
15	C0036341	Esquizofrenia		Todas las edades	8	Problemas de salud mental
16	C0153594	CÃ¡ncer testicular		Adolescencia y adultez	4	Problemas por cÃ¡ncer

17	C1302540	linfoma hallazgo		Adolescencia y adultez	4	Problemas por cÃ¡ncer
18	C0001175	SÃndrome de inmunodeficiencia adquirida		Todas las edades	10	Problemas epidemiolÃ³gicos
19	C0339901	infecciones respiratorias agudas	Kids OR kid OR children OR child	NiÃ±ez	2	Problemas respiratorios
20	C0694549	neumonÃa adquirida en la comunidad	(Outpatient OR ambulatory care) AND elderly people OR elderly person OR old person OR old people	Adultos mayores	2	Problemas respiratorios
21	C0085580	HipertensiÃ³n primaria		Adolescencia y adultez	1	Problemas Cardiovasculares
22	C0014544	Epilepsia, todos los tipos	Kids OR kid OR children OR child	NiÃ±ez y adolescencia	8	Problemas de salud mental
23	C0237078	atenciÃ³n mÃ©dica/odontolÃ³gica	Kids OR kid OR children OR child	NiÃ±os y niÃ±as de 6 aÃ±os	6	Problemas dentales
24	C0473390	Amenaza de parto prematuro	Prevention OR preventive care	Personas gestantes	11	Problemas de maternidad y del reciÃ©n nacido
25	C0340914	alteraciÃ³n del sistema de marcapasos cardÃaco (trastorno)		Adolescencia y adultez	1	Problemas Cardiovasculares
26	C0008320	colecistectomÃa	Adults	Adultos y adultos mayores	3	Problemas por diabetes mellitus, renales y de cirugÃa general
27	C0024623	cÃ¡ncer gÃ¡strico		Todas las edades	4	Problemas por cÃ¡ncer
28	C0376358	CÃ¡ncer de prÃ³stata		Adolescencia y adultez	4	Problemas por cÃ¡ncer
29	C0015397	trastorno oftalmolÃ³gico	elderly people OR elderly person OR old person OR old people	Adultos mayores	7	Problemas oftalmolÃ³gicos y otorrinonaringolÃ³gicos
30	C0038379	estrabismo	Kids OR kid OR children OR child	NiÃ±ez	7	Problemas oftalmolÃ³gicos y otorrinonaringolÃ³gicos

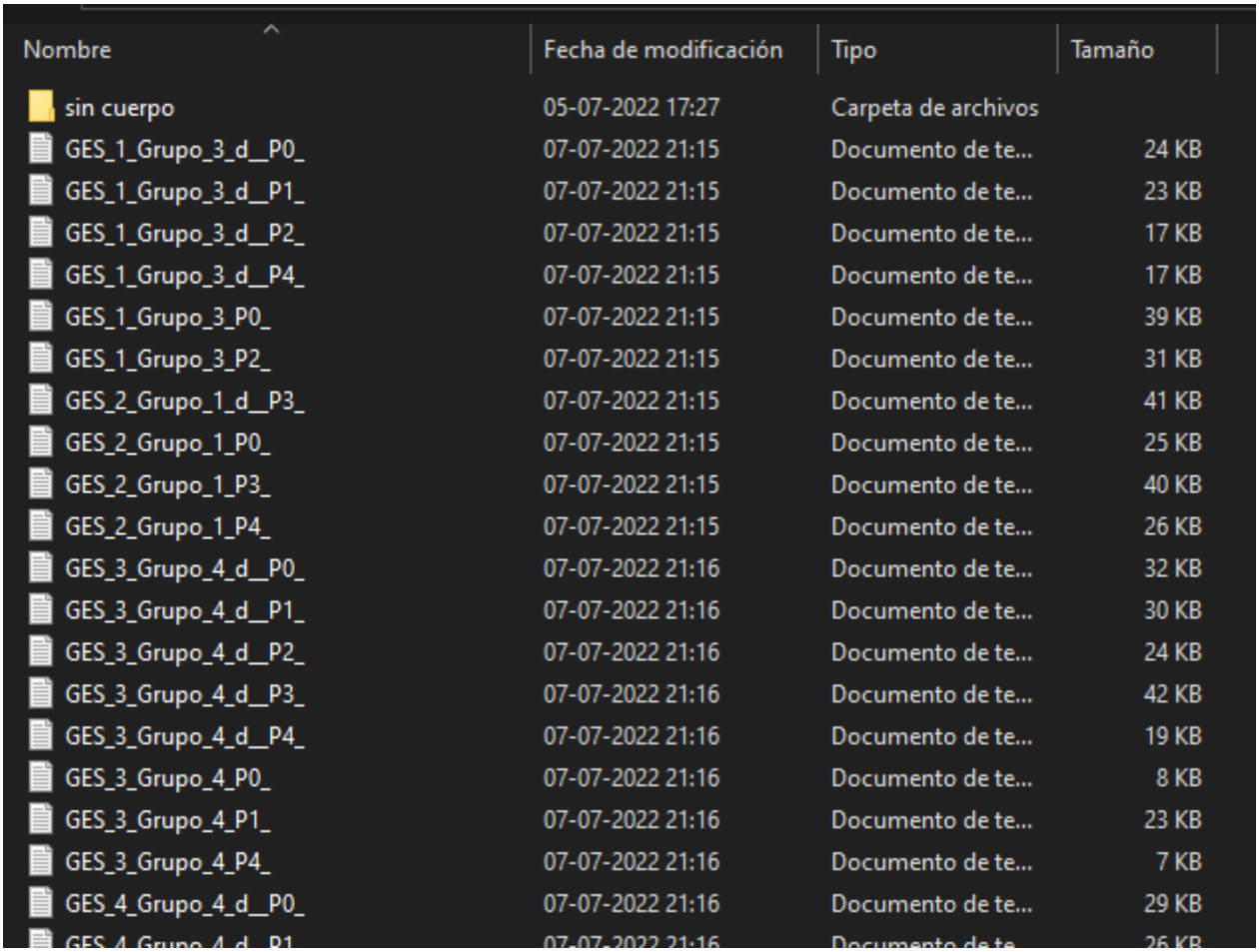
31	C0011884	retinopatÃa debida a diabetes mellitus		Todas las edades	7	Problemas oftalmolÃ³gicos y otorrinonaringolÃ³gicos
32	C0271055	desprendimiento de retina regmatÃ³geno		Todas las edades	7	Problemas oftalmolÃ³gicos y otorrinonaringolÃ³gicos
33	C0684275	hemofilia		Todas las edades	9	Problemas por enfermedades congÃ©nitas
34	C0011581	depresiÃ³n		Adolescencia y adultez	8	Problemas de salud mental
35	C1704272	hiperplasia prostÃ¡tica benigna		Todas las edades	4	Problemas por cÃ¡ncer
36	C0029355	Orthopedics	(technical support) AND elderly people OR elderly person OR old person OR old people	Adultos mayores	5	Problemas por artropatÃas y ortopÃ©dicos
37	C0948008	accidente cerebrovascular isquÃ©mico		Adolescencia y adultez	1	Problemas Cardiovasculares
38	C0024117	enfermedad pulmonar obstructiva crÃ³nica		Todas las edades	2	Problemas respiratorios
39	C0004096	asma bronquial	Kids OR kid OR children OR child	NiÃ±ez y adolescencia	2	Problemas respiratorios
40	C0035220	sÃndrome de dificultad respiratoria en el reciÃ©n nacido	neonatorum	Neonatos	11	Problemas de maternidad y del reciÃ©n nacido
41	C0022408	artropatÃa		Adultos y adultos mayores	5	Problemas por artropatÃas y ortopÃ©dicos
42	C0751003	Aneurisma del Cerebro		Todas las edades	1	Problemas Cardiovasculares
43	C0085136	neoplasia del sistema nervioso central		Adolescencia y adultez	4	Problemas por cÃ¡ncer
44	C0678212	hernia de nÃºcleo pulposo de disco intervertebra		Todas las edades	5	Problemas por artropatÃas y ortopÃ©dicos
45	C0023418	leucemia		Adolescencia y adultez	4	Problemas por cÃ¡ncer
46	C0204324	procedimiento quirÃºrgico dental	(Outpatient OR ambulatory care)	Todas las edades	6	Problemas dentales

47	C0237078	atenciÃƒÂ³n mÃƒÂ©dica/odontolÃƒÂ³gica	elderly people OR elderly person OR old person OR old people	Adultos de 60 aÃ±os	6	Problemas dentales
48	C0026771	lesiones traumÃ¡ticas mÃºltiples		Todas las edades	13	Problemas por urgencias y traumas
49	C0018674	traumatismo craneoencefÃ¡lico		Todas las edades	13	Problemas por urgencias y traumas
50	C0339055	daÃ±o de globo ocular		Todas las edades	7	Problemas oftalmolÃ³gicos y otorrinonaringolÃ³gicos
51	C0010674	fibrosis quÃstica		Todas las edades	2	Problemas respiratorios
52	C0003873	artritis reumatoide		Todas las edades	5	Problemas por artropatÃas y ortopÃ©dicos
53	C1510472	dependencia de droga		NiÃ±ez y adolescencia	8	Problemas de salud mental
54	C1301740	analgesia durante el trabajo de parto		Personas gestantes	11	Problemas de maternidad y del reciÃ©n nacido
55	C0006434	lesiÃ³n traumÃ¡tica por quemadura		Todas las edades	13	Problemas por urgencias y traumas
56	C0018775	pÃ©rdida auditiva bilateral	elderly people OR elderly person OR old person OR old people	Adultos mayores	7	Problemas oftalmolÃ³gicos y otorrinonaringolÃ³gicos
57	C0035344	retinopatÃa de la prematurez	neonatorum	Neonatos	11	Problemas de maternidad y del reciÃ©n nacido
58	C0006287	displasia broncopulmonar del reciÃ©n nacido	neonatorum	Neonatos	11	Problemas de maternidad y del reciÃ©n nacido
59	C0521786	pÃ©rdida de la audiciÃ³n, neonatal	neonatorum	Neonatos	11	Problemas de maternidad y del reciÃ©n nacido
60	C0014544	Epilepsia, todos los tipos		Adolescencia y adultez	8	Problemas de salud mental
61	C0004096	asma bronquial		Adolescencia y adultez	2	Problemas respiratorios
62	C0030567	enfermedad de Parkinson		Todas las edades	12	Problemas neurolÃ³gicos,

						inmunolÃ³gicos y de desorden endocrino
63	C3495559	artritis idiopÃ¡tica juvenil		NiÃ±ez, adolescencia y adultez	5	Problemas por artropatÃas y ortopÃ©dicos
64	C0035078	insuficiencia renal	secondary prevention OR preventive care	Todas las edades	3	Problemas por diabetes mellitus, renales y de cirugÃa general
65	C4551649	displasia congÃ©nita de cadera		Todas las edades	5	Problemas por artropatÃas y ortopÃ©dicos
66	C0237078	atenciÃƒÂ³n mÃƒÂ©dica/odontolÃƒÂ³gica	pregnancy OR pregnant	Adolescencia y adultez	6	Problemas dentales
67	C0751967	esclerosis mÃºltiple remitente recidivante		Todas las edades	12	Problemas neurolÃ³gicos, inmunolÃ³gicos y de desorden endocrino
68	C0019163	hepatitis viral tipo B		Todas las edades	10	Problemas epidemiolÃ³gicos
69	C0019196	hepatitis viral tipo C		Todas las edades	10	Problemas epidemiolÃ³gicos
70	C0009404	neoplasia del intestino grueso		Adolescencia y adultez	4	Problemas por cÃ¡ncer
71	C0919267	neoplasia de ovario		Todas las edades	4	Problemas por cÃ¡ncer
72	C0005684	tumor maligno de vejiga		Adolescencia y adultez	4	Problemas por cÃ¡ncer
73	C0029463	osteosarcoma		Adolescencia y adultez	4	Problemas por cÃ¡ncer
74	C1260873	valvulopatÃa aÃ³rtica		Adolescencia y adultez	1	Problemas Cardiovasculares
75	C0005586	trastorno bipolar		Adolescencia y adultez	7	Problemas oftalmolÃ³gicos y otorrinonaringolÃ³gicos

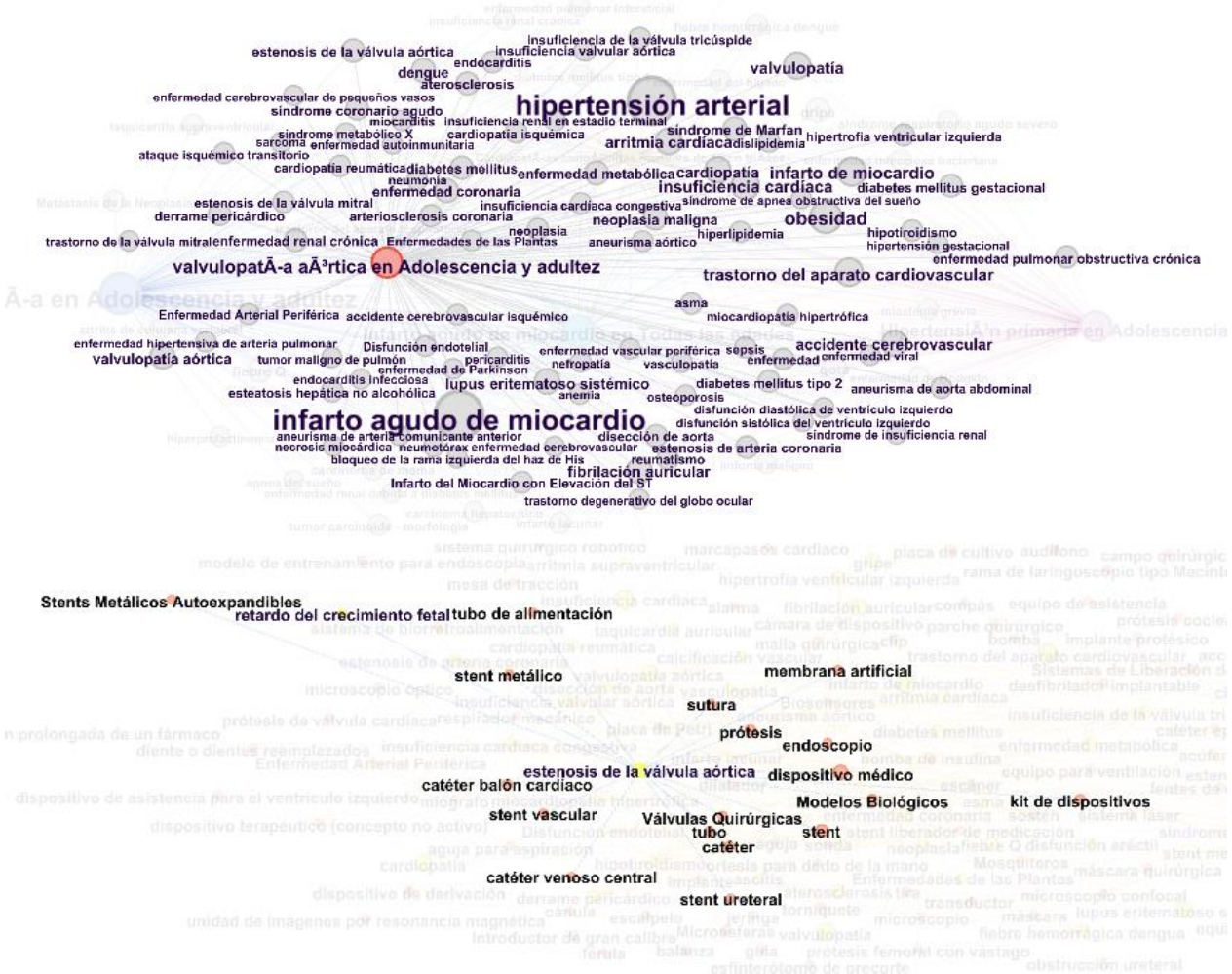
76	C0020676	hipotiroidismo		Adolescencia y adultez	12	Problemas neurolÃ³gicos, inmunolÃ³gicos y de desorden endocrino
77	C1384666	pÃ©rdida de la audiciÃ³n	Kids OR kid OR children OR child	NiÃ±ez	7	Problemas oftalmolÃ³gicos y otorrinonaringolÃ³gicos
78	C0024141	lupus eritematoso sistÃ©mico		Todas las edades	12	Problemas neurolÃ³gicos, inmunolÃ³gicos y de desorden endocrino
79	C0018824	valvulopatÃa		Adolescencia y adultez	1	Problemas Cardiovasculares
80	C0850666	infecciÃ³n causada por Helicobacter pylori		Todas las edades	10	Problemas epidemiolÃ³gicos
81	C0684249	cÃ¡ncer de pulmÃ³n		Adolescencia y adultez	4	Problemas por cÃ¡ncer
82	C0238462	carcinoma medular tiroideo		Adolescencia y adultez	4	Problemas por cÃ¡ncer
83	C0007134	carcinoma de cÃ©lulas renales		Adolescencia y adultez	4	Problemas por cÃ¡ncer
84	C0026764	mieloma mÃºltiple		Adolescencia y adultez	4	Problemas por cÃ¡ncer
85	C0497327	demencia		Todas las edades	12	Problemas neurolÃ³gicos, inmunolÃ³gicos y de desorden endocrino

ANEXO: CARPETA DE ALMACENAMIENTO DE ARTÍCULOS DE PATOLOGÍAS



Nombre	Fecha de modificación	Tipo	Tamaño
sin cuerpo	05-07-2022 17:27	Carpeta de archivos	
GES_1_Grupo_3_d__P0_	07-07-2022 21:15	Documento de te...	24 KB
GES_1_Grupo_3_d__P1_	07-07-2022 21:15	Documento de te...	23 KB
GES_1_Grupo_3_d__P2_	07-07-2022 21:15	Documento de te...	17 KB
GES_1_Grupo_3_d__P4_	07-07-2022 21:15	Documento de te...	17 KB
GES_1_Grupo_3_P0_	07-07-2022 21:15	Documento de te...	39 KB
GES_1_Grupo_3_P2_	07-07-2022 21:15	Documento de te...	31 KB
GES_2_Grupo_1_d__P3_	07-07-2022 21:15	Documento de te...	41 KB
GES_2_Grupo_1_P0_	07-07-2022 21:15	Documento de te...	25 KB
GES_2_Grupo_1_P3_	07-07-2022 21:15	Documento de te...	40 KB
GES_2_Grupo_1_P4_	07-07-2022 21:15	Documento de te...	26 KB
GES_3_Grupo_4_d__P0_	07-07-2022 21:16	Documento de te...	32 KB
GES_3_Grupo_4_d__P1_	07-07-2022 21:16	Documento de te...	30 KB
GES_3_Grupo_4_d__P2_	07-07-2022 21:16	Documento de te...	24 KB
GES_3_Grupo_4_d__P3_	07-07-2022 21:16	Documento de te...	42 KB
GES_3_Grupo_4_d__P4_	07-07-2022 21:16	Documento de te...	19 KB
GES_3_Grupo_4_P0_	07-07-2022 21:16	Documento de te...	8 KB
GES_3_Grupo_4_P1_	07-07-2022 21:16	Documento de te...	23 KB
GES_3_Grupo_4_P4_	07-07-2022 21:16	Documento de te...	7 KB
GES_4_Grupo_4_d__P0_	07-07-2022 21:16	Documento de te...	29 KB
GES_4_Grupo_4_d__P1_	07-07-2022 21:16	Documento de te...	26 KB

ANEXO: CAPTURAS DE LA RED DE INFORMACIÓN


hipertensión arterial
infarto agudo de miocardio
valvulopatía
insuficiencia de la válvula tricúspide
estenosis de la válvula aórtica
insuficiencia valvular aórtica
endocarditis
dengue
aterosclerosis
enfermedad cerebrovascular de pequeños vasos
síndrome coronario agudo
miocarditis
insuficiencia renal en estadio terminal
síndrome metabólico X
cardiopatía isquémica
síndrome de Marfan
hipertrofia ventricular izquierda
sarcoma
enfermedad autoinmunitaria
arritmia cardíaca
dislipidemia
ataque isquémico transitorio
cardiopatía reumática
diabetes mellitus
enfermedad metabólica
cardiopatía
infarto de miocardio
neumonía
enfermedad coronaria
insuficiencia cardíaca
diabetes mellitus gestacional
estenosis de la válvula mitral
insuficiencia cardíaca congestiva
síndrome de apnea obstructiva del sueño
derrame pericárdico
arteriosclerosis coronaria
neoplasia maligna
obesidad
neoplasia
hipotiroidismo
trastorno de la válvula mitral
enfermedad renal crónica
Enfermedades de las Plantas
aneurisma aórtico
hiperlipidemia
hipertensión gestacional
enfermedad pulmonar obstructiva crónica
valvulopatÃ-a aÃ³rtica en Adolescencia y adultez
trastorno del aparato cardiovascular
asma
Enfermedad Arterial Periférica
accidente cerebrovascular isquémico
miocardiopatía hipertrófica
enfermedad hipertensiva de arteria pulmonar
Disfunción endotelial
enfermedad vascular periférica
sepsis
accidente cerebrovascular
valvulopatía aórtica
tumor maligno de pulmón
pericarditis
nefropatía
vasculopatía
enfermedad
enfermedad viral
enfermedad de Parkinson
endocarditis infecciosa
lupus eritematoso sistémico
diabetes mellitus tipo 2
aneurisma de aorta abdominal
esteatosis hepática no alcohólica
anemia
osteoporosis
disfunción diastólica de ventrículo izquierdo
disfunción sistólica del ventrículo izquierdo
aneurisma de arteria comunicante anterior
disección de aorta
síndrome de insuficiencia renal
necrosis miocárdica
neumotórax
enfermedad cerebrovascular
estenosis de arteria coronaria
bloqueo de la rama izquierda del haz de His
reumatismo
fibrilación auricular
Infarto del Miocardio con Elevación del ST
trastorno degenerativo del globo ocular
Stents Metálicos Autoexpandibles
retardo del crecimiento fetal
tubo de alimentación
membrana artificial
stent metálico
sutura
prótesis
endoscopio
estenosis de la válvula aórtica
dispositivo médico
catéter balón cardíaco
Modelos Biológicos
kit de dispositivos
stent vascular
Válvulas Quirúrgicas
tubo
stent
catéter
catéter venoso central
stent ureteral

ANEXO : CUADERNO DE PROGRAMACIÓN DEL TRABAJO DESARROLLADO

Redes de información de patologías GES con datos estandarizados de UMLS, a partir de la minería de entidades biomédicas en papers cientificos de la base de datos pubmed central.

---------- OBETIVO N° 1 ---------- Adquirir y adecuar el Metathesauro del sistema de lenguaje médico unificado y el listado específico de prestaciones del AUGE, para posteriormente generar el léxico, establecer una clasificación para cada uno de los 85 problemas y las expresiones para la búsqueda de artículos científicos.

In [1]:
```
import time
import os
import pandas as pd

metricas = []
```

Primero se selecciona el directorio de la carpeta principal que contiene este cuaderno y los distintos archivos necesarios como los archivos de la UMLS MRCONSO.RRF y MRSTY.RRF, además de los problemas ges.

In [2]:
```
directory=r'/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db/trabajo de título Herman Chin
os.chdir(directory)
```

Acondicionar datos de UMLS metathesaurus - pasar de formatos .RRF a csv procesables

In [ ]:
```
#### función para obtener etiquetas de las columnas de los distintos archivos RRF
def openmrfiles():
    with open(os.path.join("MRFILES.RRF"), encoding="utf-8") as f:
            mrfiles= f.read()

    mrfiles=mrfiles.split('\n')
    cont=0
    for i in mrfiles:
        i = i.split("|")
        mrfiles[cont]=i
        cont=cont+1

    return(mrfiles)
```

In [ ]:
```
### Adquisición de MRCONSO.RRF como csv, el cual contiene las codificacion de todos los
with open(os.path.join("MRCONSO.RRF"), encoding="utf-8") as f:
        conso= f.read()

conso=conso.split('\n')
cont=0
for i in conso:
    i = i.split("|")
    conso[cont]=i
    cont=cont+1
columns = openmrfiles() #para obtener nombre de columnas
columns = columns[9][2] ##selección de columnas atribuidas a conso
columns = columns.split(",")

conso=pd.DataFrame(data=conso)
del(conso[18]) #dato sobrante
conso.columns = columns
conso.to_csv("umls_conso.csv", sep=';')
```

In [ ]:
```
#### Adquisición de MRSTY.RFF como csv, el que tiene la codificación de los tipos de gru
with open(os.path.join("MRSTY.RRF"), encoding="utf-8") as f:
```

```
        sty= f.read()

sty=sty.split('\n')
cont=0
for i in sty:
    i = i.split("|")
    sty[cont]=i
    cont=cont+1
columns = openmrfiles()
columns = columns[22][2] ##columnas atribuidas a sty
columns = columns.split(",")

sty=pd.DataFrame(data=sty)
del(sty[6]) #dato sobrante
sty.columns = columns
sty.to_csv("umls_sty.csv", sep=";")
```

A continuación se acondicionan los datos del listado específico de rpestaciones del GES, disponible en la web https://auge.minsal.cl/problemasdesalud/lep en formato excel, con el objetivo de obtener los 85 problemas de salud asociados.

In [ ]:

```
########## Arreglar excel del listado específico de prestaciones del GES para obtener lo
ges = pd.read_excel("LEP_2019_Excel.xlsx") ## Listado específico de prestaciones
aux1=0
aux2=0
aux3=0
aux4=0
aux5=0
bol=ges.isnull()
fges=ges.copy()

for i in range(len(ges)):
    if bol.loc[i,"N°"] == False:
        aux1 = ges.loc[i,"N°"]

    else:
        fges.loc[i,"N°"] = aux1
for i in range(len(ges)):
    if bol.loc[i,"Problema de salud"] == False:
        aux2 = ges.loc[i,"Problema de salud"]

    else:
        fges.loc[i,"Problema de salud"] = aux2
for i in range(len(ges)):
    if bol.loc[i,"Intervención sanitaria"] == False:
        aux3 = ges.loc[i,"Intervención sanitaria"]

    else:
        fges.loc[i,"Intervención sanitaria"] = aux3
for i in range(len(ges)):
    if bol.loc[i,"Prestación o grupo de prestaciones"] == False:
        aux4 = ges.loc[i,"Prestación o grupo de prestaciones"]

    else:
        fges.loc[i,"Prestación o grupo de prestaciones"] = aux4
for i in range(len(ges)):
    if bol.loc[i,"Glosa"] == False:
        aux5 = ges.loc[i,"Glosa"]

    else:
        fges.loc[i,"Glosa"] = aux5

fges.to_csv("catálogo ges acondicionado.csv",index=False, sep=";")
```

```
##Queda un dataframe con los 85 poblemas de salud y se guarda en csv
entidades=fges.drop_duplicates(subset=["N°"], keep="first")
entidades = entidades[["N°","Problema de salud"]]
entidades.to_csv("85 problemas.csv",index=False, sep=";")
```

In [3]:
```
sty=pd.read_csv("umls_sty.csv",sep=";")
conso=pd.read_csv("umls_conso.csv",sep=";")
```

```
/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db/Anaconda3/envs/gunlp/lib/python3.9/site-p
ackages/IPython/core/interactiveshell.py:3398: DtypeWarning: Columns (11) have mixed typ
es.Specify dtype option on import or set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

Una vez obtenido los 85 problemas GES, se procedió a categorizarlos en un archivo excel en 6 distintas dimensiones, primero se le asignó un código UMLS CUI que englobaba el concepto principal de cada problema, haciendo una búsqueda en el portal web del metathesauro https://uts.nlm.nih.gov/uts/umls/home. En segundo lugar se asignó un criterio de búsqueda para los problemas que abarcan una población objetivo en específico. En tercer lugar se clasificaron con la población objetivo a la que abarcan cada uno. En cuarto lugar a cada uno de los problemas se le asign+o uno de 13 grupos segun su relación entre cada uno. Y finalmente se le asignó el nombre de cada grupo.

Para el presente trabajo, se seleccionan los problemas pertenecientes al grupo 1 -> 'Problemas Cardiovasculares'.

In [4]:
```
problemas=pd.read_excel("Categorización 85 problemas.xlsx")
problemas = problemas[problemas["Grupo"]==1]
problemas
```

Out[4]:

	N°	CUI	Problema de salud	Criterio de búsqueda	Población objetivo	Grupo	Nombre
1	2	C0152021	CardiopatÃas congÃ©nitas menores de 15	Kids OR kid OR children OR child	NiÃ±ez	1	Problemas Cardiovasculares
4	5	C0155626	Infarto agudo de miocardio	NaN	Todas las edades	1	Problemas Cardiovasculares
20	21	C0085580	HipertensiÃ³n primaria	NaN	Adolescencia y adultez	1	Problemas Cardiovasculares
24	25	C0340914	alteraciÃ³n del sistema de marcapasos cardÃac...	NaN	Adolescencia y adultez	1	Problemas Cardiovasculares
73	74	C1260873	valvulopatÃa aÃ³rtica	NaN	Adolescencia y adultez	1	Problemas Cardiovasculares
78	79	C0018824	valvulopatÃa	NaN	Adolescencia y adultez	1	Problemas Cardiovasculares

In [5]:
```
cods_problemas =  pd.merge(problemas,conso, on="CUI",how="inner").drop_duplicates()
cods_problemas = cods_problemas[["N°","Problema de salud","CUI","Criterio de búsqueda","
cods_problemas
```

Out[5]:

	N°	Problema de salud	CUI	Criterio de búsqueda	Población objetivo	Grupo	Nombre	LAT	SAB	S
0	2	CardiopatÃ-as congÃ©nitas menores de 15	C0152021	Kids OR kid OR children OR child	NiÃ±ez	1	Problemas Cardiovasculares	ENG	SNOMEDCT_US	Congeni he disea

1	2	CardiopatÃ-as congÃ©nitas menores de 15	C0152021	Kids OR kid OR children OR child	NiÃ±ez	1	Problemas Cardiovasculares	ENG	SNOMEDCT_US	Congeni he disea
2	2	CardiopatÃ-as congÃ©nitas menores de 15	C0152021	Kids OR kid OR children OR child	NiÃ±ez	1	Problemas Cardiovasculares	ENG	SNOMEDCT_US	Congeni he disea
3	2	CardiopatÃ-as congÃ©nitas menores de 15	C0152021	Kids OR kid OR children OR child	NiÃ±ez	1	Problemas Cardiovasculares	ENG	MDR	He disea congeni
4	2	CardiopatÃ-as congÃ©nitas menores de 15	C0152021	Kids OR kid OR children OR child	NiÃ±ez	1	Problemas Cardiovasculares	ENG	MDR	He disea congeni
...	...	...	...	...	...	...	...	...	...	
171	79	valvulopatÃa	C0018824	NaN	Adolescencia y adultez	1	Problemas Cardiovasculares	SPA	MDRSPA	Cardiopa valvu NEC
172	79	valvulopatÃa	C0018824	NaN	Adolescencia y adultez	1	Problemas Cardiovasculares	SPA	MDRSPA	Enfermed valvu cardia
173	79	valvulopatÃa	C0018824	NaN	Adolescencia y adultez	1	Problemas Cardiovasculares	SPA	MDRSPA	Enfermed valvu cardia
174	79	valvulopatÃa	C0018824	NaN	Adolescencia y adultez	1	Problemas Cardiovasculares	SPA	MDRSPA	Trastorn de válvulas coraz
175	79	valvulopatÃa	C0018824	NaN	Adolescencia y adultez	1	Problemas Cardiovasculares	SPA	MDRSPA	Valvulopa cardia

176 rows × 10 columns

Una vez reunido el léxico UMLS en relacion a los 85 problemas del GES, se procede a establecer los distintos términos de búsqueda para encontrar papers asociados a cada problema. Por cada término de los 85 problemas se generan dos expresiones, una sólo con el término del nombre del CUI más la población objetivo en caso de que se necesite, y otra expresión con las mismas características más la expresión "related diseases" para enfocar la búsqueda en patologías relacionadas

In [6]:
```
terms = cods_problemas[["N°","Grupo","Nombre","STR","Criterio de búsqueda","LAT"]]
terms = terms[terms["LAT"] == "ENG"].drop_duplicates() ## Se seleccionan sólo los termin
terms = terms[terms.index%2 == 0] ## para utilizar la mitad de las expresiones
terms
```

Out[6]:

	N°	Grupo	Nombre	STR	Criterio de búsqueda	LAT
0	2	1	Problemas Cardiovasculares	Congenital heart disease	Kids OR kid OR children OR child	ENG
6	2	1	Problemas	Congenital heart disease NOS	Kids OR kid OR children OR	ENG

			Cardiovasculares		child	
8	2	1	Problemas Cardiovasculares	Heart Diseases, Congenital	Kids OR kid OR children OR child	ENG
10	2	1	Problemas Cardiovasculares	Congenital cardiac disorders	Kids OR kid OR children OR child	ENG
20	5	1	Problemas Cardiovasculares	Acute myocardial infarction	NaN	ENG
28	5	1	Problemas Cardiovasculares	Acute myocardial infarction, NOS	NaN	ENG
30	5	1	Problemas Cardiovasculares	Acute myocardial infarction, unspecified site,...	NaN	ENG
32	5	1	Problemas Cardiovasculares	AMI - Acute myocardial infarction	NaN	ENG
36	5	1	Problemas Cardiovasculares	MI - acute myocardial infarction	NaN	ENG
40	5	1	Problemas Cardiovasculares	Acute myocardial infarction (disorder)	NaN	ENG
54	21	1	Problemas Cardiovasculares	Essential hypertension	NaN	ENG
62	21	1	Problemas Cardiovasculares	Essential hypertension NOS	NaN	ENG
64	21	1	Problemas Cardiovasculares	Hypertension NOS (& [essential])	NaN	ENG
66	21	1	Problemas Cardiovasculares	Hypertension, Essential	NaN	ENG
68	21	1	Problemas Cardiovasculares	Unspecified essential hypertension	NaN	ENG
72	21	1	Problemas Cardiovasculares	Systemic primary arterial hypertension	NaN	ENG
74	21	1	Problemas Cardiovasculares	Essential hypertension (disorder)	NaN	ENG
76	21	1	Problemas Cardiovasculares	Essential hypertension NOS (disorder)	NaN	ENG
94	25	1	Problemas Cardiovasculares	Disorder of cardiac pacemaker system (disorder)	NaN	ENG
98	74	1	Problemas Cardiovasculares	Aortic valve disease	NaN	ENG
102	74	1	Problemas Cardiovasculares	Aortic valve disease NOS	NaN	ENG
104	74	1	Problemas Cardiovasculares	Valve Disease, Aortic	NaN	ENG
106	74	1	Problemas Cardiovasculares	Aortic valve disorder	NaN	ENG
108	74	1	Problemas Cardiovasculares	Aortic valve disorder, NOS	NaN	ENG
110	74	1	Problemas Cardiovasculares	Aortic Valve Disorders	NaN	ENG
112	74	1	Problemas	AVD - Aortic valve disease	NaN	ENG

			Cardiovasculares			
114	74	1	Problemas Cardiovasculares	Aortic Heart Diseases	NaN	ENG
116	74	1	Problemas Cardiovasculares	Aortic Valvular Heart Disease	NaN	ENG
118	74	1	Problemas Cardiovasculares	Aortic valvular disorders	NaN	ENG
120	74	1	Problemas Cardiovasculares	Aortic valve disorders NOS (disorder)	NaN	ENG
134	79	1	Problemas Cardiovasculares	Valve Disease, Heart	NaN	ENG
136	79	1	Problemas Cardiovasculares	Heart valve disease	NaN	ENG
140	79	1	Problemas Cardiovasculares	Valvular heart disease	NaN	ENG
144	79	1	Problemas Cardiovasculares	Disease, Heart Valvular	NaN	ENG
146	79	1	Problemas Cardiovasculares	Valvular Disease, Heart	NaN	ENG
148	79	1	Problemas Cardiovasculares	Heart Disease, Valvular	NaN	ENG
150	79	1	Problemas Cardiovasculares	Heart valve disorder	NaN	ENG
152	79	1	Problemas Cardiovasculares	Heart valve disorder, NOS	NaN	ENG
154	79	1	Problemas Cardiovasculares	Heart valve disorders	NaN	ENG
158	79	1	Problemas Cardiovasculares	Cardiac valvulopathy	NaN	ENG
160	79	1	Problemas Cardiovasculares	Heart valve disorder (disorder)	NaN	ENG

In [7]:
```
exprs=[]
bol_ss = terms.isnull() ### Dataframe auxiliar para identificar si el problema tiene o n
for i in terms.index:
    iter_num =str(terms["N°"][i])
    cod = "_"+str(i)
    g_dir = directory+"\\"+"Grupo_"+str(terms["Grupo"][i])+"_"+str(terms["Nombre"][i])
    if bol_ss["Criterio de búsqueda"][i] != True:  ## Si tiene población objetivo, se ag
        iter_term = '"'+str(terms["STR"][i])+'"' + " AND "+"("+ str(terms["Criterio de b
        iter_term2 = iter_term+' AND "related diseases"' #por cada término se generan 2
    else:
        iter_term = '"'+str(terms["STR"][i])+'"'
        iter_term2 = iter_term+' AND "related diseases"'
    exprs.append([terms["Grupo"][i],iter_num,cod,iter_term,g_dir])
    exprs.append([terms["Grupo"][i],iter_num,cod+"_d",iter_term2,g_dir])

os.chdir(directory)
expresiones = pd.DataFrame(data=exprs, columns=["Grupo","N°","cod","expresion","director
expresiones.to_csv("expresiones de búsqueda problemas-patologias.csv",sep=";")
expresiones
```

Out[7]:

	Grupo	N°	cod	expresion	directorio

0	1	2	_0	"Congenital heart disease" AND (Kids OR kid OR...	/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db...
1	1	2	_0_d	"Congenital heart disease" AND (Kids OR kid OR...	/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db...
2	1	2	_6	"Congenital heart disease NOS" AND (Kids OR ki...	/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db...
3	1	2	_6_d	"Congenital heart disease NOS" AND (Kids OR ki...	/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db...
4	1	2	_8	"Heart Diseases, Congenital" AND (Kids OR kid ...	/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db...
...	...	...	...	...	...
77	1	79	_154_d	"Heart valve disorders" AND "related diseases"	/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db...
78	1	79	_158	"Cardiac valvulopathy"	/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db...
79	1	79	_158_d	"Cardiac valvulopathy" AND "related diseases"	/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db...
80	1	79	_160	"Heart valve disorder (disorder)"	/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db...
81	1	79	_160_d	"Heart valve disorder (disorder)" AND "related...	/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db...

82 rows × 5 columns

Para ordenar y almacenar los corpus que contienen los papers de cada grupo de patologías se procede a crear distintos directorios.

En la búsqueda de papers que se realizará en e objetivo n°2, se obtienen papers que no tienen un cuerpo o no tienen contenido como tal, los cuales se agruparán en la carpeta 'Papers sin cuerpo'

In [8]:
```
os.chdir(directory)
#os.mkdir("Papers sin cuerpo") ## En la búsqueda de papers que veremos a continuación,
##se obtienen papers que no tienen un cuerpo o no tienen contenido como tal, los que los
###

lista_directorios = []
for i in problemas["Grupo"].drop_duplicates().index:
    strg = "Grupo "+str(problemas["Grupo"][i])+" "+str(problemas["Nombre"][i])
    print(strg)
    #os.mkdir(strg)
    g_dir = directory+"/"+strg
    print(g_dir)
    os.chdir(g_dir)
    #os.mkdir("corpus patologias")
    p_dir = directory+"/"+strg+"/"+"corpus patologias"
    os.chdir(g_dir)
    #os.mkdir("corpus signos procedimientos y dispositivos medicos")
    ents_dir = directory+"/"+strg+"/"+"corpus signos procedimientos y dispositivos medic
    os.chdir(directory)
    lista_directorios.append([problemas["Grupo"][i],g_dir,p_dir,ents_dir])
os.chdir(directory)
df_dir=pd.DataFrame(data=lista_directorios,columns=["Grupo","directorio grupo","director
df_dir.to_csv("directorios de grupos.csv", sep=';')
df_dir
```

```
Grupo 1 Problemas Cardiovasculares
/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db/trabajo de título Herman Chinga/Grupo 1 P
roblemas Cardiovasculares
```

Out[8]:

	Grupo	directorio grupo	directorio patologias	directorio entidades
0	1	/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db...	/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db...	/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db...

---------- OBJETIVO 2 ----------

Hacer distintas búsquedas de artículos científicos para cada problema del AUGE seleccionado y almacenar cada corpus obtenido en distintos directorios, con el objetivo de aplicar técnicas de reconocimiento de entidades con procesamiento de lenguaje natural en cada artículo y así obtener las patologías relacionadas a cada problema.

In [9]:
```
import os
import pandas as pd
import time

sin_cuerpo_dir = r'/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db/trabajo de título Herm
## Recursos de objetivo n°1
## expresiones de búsqueda
os.chdir(directory)
expresiones = pd.read_csv("expresiones de búsqueda problemas-patologias.csv", sep=";")
df_dir = pd.read_csv('directorios de grupos.csv',sep=";")
expresiones_dirs =pd.merge(expresiones,df_dir,on="Grupo",how="right")
```

EN PROCESAMIENTO DE LENGUAJE NATURAL MODIFICAR DIRECTORIOS DE RECURSOS

Para realizar la adquisición de corpus de literatura biomédica asociada a los términos de búsqueda del GES, se procedió a buscar recursos en github que permitieran acceder a los papers completos de la base de datos pubmed central, seleccionando el recurso de acceso abierto parse-pmc del usuario cyclecycle disponible en https://github.com/cyclecycle/parse-pubmed. El cual utiliza las librerías biopython y los recursos de UMLS Entrez. Los distintos corpus se almacenan en los directorios creados anteriormente.

Mecanismo de búsqueda para encontrar papers, a la función search ingresa el número del problema ges y la expresión de búsqueda para otorgarle nombre a cada paper adquirido y tambíen ingresa el directorio en el que se guardará cada paper.

In [ ]:
```
from Bio import Entrez
from pmc import PMCArticleSet

def search(n,cod,term,g_dir):
    try:
        Entrez.email = 'herman.chinga@alumnos.uv.cl'
        busq=Entrez.esearch(db="pmc",
                            sort='relevance',
                            retmax='10',
                            retmode="null",
                            term=term)
        print("Buscando término ", term)

        results=Entrez.read(busq)
        ids = results["IdList"]
        print("Se encontraron ",len(ids), " papers")

        handle = Entrez.efetch(db='pmc',
                               id=ids,
                               rettype='full',
```

```
                                retmode='xml')
        xml = handle.read()
        #print(xml)

        articles = PMCArticleSet(xml).articles  # List of PMCArticle objects
        cont = 0
        for article in articles:
            if article.body != "":
                #print(article.body)
                print("guardando paper ",cont)
                os.chdir(g_dir)
                nombre = str("GES_"+str(n)+cod+"_P"+str(cont)+"_.txt")
                doc= open(nombre,"w",encoding='utf8')
                paper = str(article.ids)+"||\n"+str(article.title)+"||\n"+str(article.bo
                doc.write(paper)
                doc.close()
                cont = cont+1
            else:
              #  print("paper sin cuerpo")
                os.chdir(sin_cuerpo_dir)
                nombre = str("GES_"+str(n)+cod+"_P"+str(cont)+"_.txt")
                doc= open(nombre,"w",encoding='utf8')
                paper = str(article.ids)+"||\n"+str(article.title)+"||\n"+str(article.ab
                doc.write(paper)
                doc.close()
                cont = cont+1
    except:
        metricas.append(["error en búsqueda de papers",term])
        print("error")
```

In [ ]:
```
start = time.time()
cont=0
for i in expresiones.index:
    cont+=1
    print('va en ',cont,' de ',len(expresiones_dirs["N°"]))
    print(expresiones_dirs["N°"][i],expresiones_dirs["expresion"][i],expresiones_dirs["d
    search(expresiones_dirs["N°"][i],expresiones_dirs["cod"][i],expresiones_dirs["expres
os.chdir(directory)
end = time.time()
metricas.append(["Descarga de papers-problemas ",(end-start)/60])
```

Eliminar papers de tamaño mayor a 150Kbytes

In [ ]:
```
os.chdir(directory)
papers_ids =[]
for i in df_dir.index:
    p_dir = df_dir["directorio patologias"][i]
    os.chdir(p_dir)
    papers = os.listdir(p_dir)
    total = len(papers)
    cont=0
    grupo=df_dir["Grupo"][i]
    metricas.append(["Cantidad de papers encontraados en grupo",str(grupo),total])
    for txt in papers:
        tamaño = os.path.getsize(txt)
        if tamaño > 150000:
            cont+=1
            os.remove(txt)
    metricas.append([str("Papers eliminados por tamaño en grupo ")+str(grupo),cont])
metricas
```

Eliminar papers duplicados

In [ ]:
```
import ast
os.chdir(directory)
papers_ids =[]
for i in df_dir.index:
    p_dir = df_dir["directorio patologias"][i]
    os.chdir(p_dir)
    papers = os.listdir(p_dir)
    total = len(papers)
    cont=0
    grupo=df_dir["Grupo"][i]
    for txt in papers:
        arch = open(txt, "r",encoding='utf8')
        # print(txt)
        paper = arch.read()
        arch.close()
        paper = paper.split("||\n")
        #print(paper[0])
        ids = ast.literal_eval(paper[0])
        paper_id = ids["pmc"]
        if paper_id not in papers_ids:
            papers_ids.append(paper_id)
            #print(paper_id)
        else:
            os.remove(txt)
            cont+=1
    metricas.append([str("Papers duplicados eliminados en grupo ")+str(grupo),cont])
    # print(paper_id)
    papers = os.listdir(p_dir)
    total = len(papers)
    metricas.append(["total de papers en grupo ",str(grupo),total])
metricas
```

Recursos para procesamiento de lenguaje natural para identificar patologías en corpus utilizando modelo pre-entrenado biobert de hugging-face y medcat

In [10]:
```
os.chdir(directory)
from transformers import AutoTokenizer, BertForTokenClassification, pipeline
from nltk.corpus import stopwords
from medcat.cat import CAT

path = "/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db/trabajo de título Herman Chinga/b
tokenizer = AutoTokenizer.from_pretrained(path,return_tensors='pt')
model =     BertForTokenClassification.from_pretrained(path)
ner = pipeline("ner", model=model, tokenizer=tokenizer,aggregation_strategy='max')

cat = CAT.load_model_pack('/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db/trabajo de tít
```

```
The CDB was exported by an unknown version of MedCAT.
{
  "Model ID": null,
  "Last Modifed On": null,
  "History (from least to most recent)": [],
  "Description": "No description",
  "Source Ontology": null,
  "Location": null,
  "MetaCAT models": {},
  "Basic CDB Stats": {},
  "Performance": {
    "ner": {},
    "meta": {}
  },
  "Important Parameters (Partial view, all available in cat.config)": {
    "config.ner['min_name_len']": {
      "value": 3,
```

```
      "description": "Minimum detection length (found terms/mentions shorter than this w
ill not be detected)."
    },
    "config.ner['upper_case_limit_len']": {
      "value": 3,
      "description": "All detected terms shorter than this value have to be uppercase, o
therwise they will be ignored."
    },
    "config.linking['similarity_threshold']": {
      "value": 0.2,
      "description": "If the confidence of the model is lower than this a detection will
be ignore."
    },
    "config.general['spell_check']": {
      "value": true,
      "description": "Is spell checking enabled."
    },
    "config.general['spell_check_len_limit']": {
      "value": 7,
      "description": "Words shorter than this will not be spell checked."
    }
  },
  "MedCAT Version": null
}
```


A continuación se expresan dos funciones, la primera es para obtener las patologías encontradas en cada paper mediante la técnica de NER de biobert.

In [11]:
```python
def related_pats(name,n,g):
    #### ubicar en directorio de papers
    #### Lectura de papers
    cont=0
    try:
        arch = open(name, "r", encoding='utf8')
        paper = arch.read() ## se guarda variable con texto
        arch.close()
        paper = paper.split("||\n")
        ##### caracteristicas del paper
        title = paper[1]

        ### procesamiento de texto
        sw_nltk = stopwords.words('english')
        cuerpo = paper[2]
        cuerpo = cuerpo.split(". ") #### N bins
        proc_text = []

        print("procesamiento de texto")
        for i in range(len(cuerpo)):
            if len(cuerpo[i].split()) > 10:
                words=[]
                words = [word for word in cuerpo[i].split() if word.lower() not in sw_nl
                cuerpo[i] = " ".join(words)
                proc_text.append(cuerpo[i])
        print("texto procesado, eliminadas frases cortas y stop words")
        print("NER - patologÃas y signos con pipeline transformers")
        pats=[]
        #### METODO NER
        for parrafo in proc_text:
            for entity in ner(parrafo): ###3 SE DETECTAN PATOLOGÃ□AS ASOCIADAS CON METOD
                if entity['entity_group'] == 'DISEASE':
                    pats.append(entity["word"])
        # print(pats)
        df_pats = pd.DataFrame()
        df_pats["NÂ°"]=[n]*len(pats)
        df_pats["cod"] = [g]*len(pats)
```

```
        df_pats["title"] =[title]*len(pats)
        df_pats["STR"] = pats

        # print(data)
        os.chdir(directory)


        return(df_pats)
    except:
        cont+=1
        print("hubo error en lectura")

    metricas.append(["artículos con error de lectura ",cont])
```

Se procede a realizar la búsqueda de patologías en texto

In [ ]:
```
start = time.time()
os.chdir(directory)

for i in df_dir.index:
    g_dir = df_dir["directorio grupo"][i] ##entra a directorio de grupo
    os.chdir(g_dir)
    corpus_pats_dir= df_dir["directorio patologias"][i]

    papers_list = os.listdir(corpus_pats_dir)
    cont=0
    for name in papers_list:
        cont+=1
        os.chdir(corpus_pats_dir)
        print("leyendo paper", name)
        print("va en ",cont,"de ",len(papers_list))
        dname = name.split('_')
        nges = dname[1]  # Datos
        ngrupo = dname[3]
        pats=pd.DataFrame()
        pats = related_pats(name,nges,ngrupo)
        print("guardando patologias en",g_dir)
        os.chdir(g_dir)
        pats.to_csv("pre-patologias.csv",sep=";",mode="a")

end = time.time()
metricas.append(["Búsqueda de patologías",(end-start)/60])
```

Esta segunda función es para obtener cada código CUI de las patologías encontradas mediante medcat.

In [12]:
```
def medcat(term):
    entity = cat.get_entities(term)
    entity.pop("tokens")
    entity = entity["entities"]
    for clave, valor in entity.items():
        return(str(valor["cui"]))
```

Una vez obtenidas todas las patologías y almacenadas en el archivo csv, se procesan los datos para obtener las co-ocurrencias asociadas en cada paper

In [ ]:
```
start = time.time()
for i in df_dir.index:
    g_dir = df_dir["directorio grupo"][i]
    os.chdir(g_dir)
    entities=pd.read_csv("pre-patologias.csv",sep=";")
    entities["CUI"] = entities["STR"].apply(lambda x:  medcat(x))
    df2_ents=entities.groupby(by="CUI")["NÂ°"].describe()
    # df2_pats["CUI"]=df2_pats.index
```

```
    df2_ents=df2_ents["freq"]
    entities = pd.merge(entities,df2_ents, on="CUI",how="left").drop_duplicates(subset=[
    entities.to_csv("patologias encontradas.csv", sep=';')
end = time.time()
metricas.append(["Procesamiento de entidades - patologías",(end-start)/60])
```

Finalmente se seleccionan las patologías con según el grupo semántico respectivo

In [ ]:
```
sty=sty[["CUI","TUI"]]
for i in df_dir.index:
    g_dir = df_dir["directorio grupo"][i]
    os.chdir(g_dir)
    entities = pd.read_csv("patologias encontradas.csv", sep=';')
    entities = pd.merge(entities,sty, on="CUI",how="left")
    entities=entities[entities.TUI.isin(['T047','T191','T050'])].sort_values(by=["freq"]
    entities.to_csv("patologias encontradas.csv", sep=';')
```

In [16]:
```
entities = pd.read_csv("patologias encontradas.csv", sep=';')
entities.head(100)
```

Out[16]:

	Unnamed: 0	Unnamed: 0.1	Unnamed: 0.1.1	NÂ°	cod	title	STR	CUI	freq	TUI
0	0	26286	0.0	5	P6	Diagnostic value of copeptin combined with hyp...	Acute myocardial infarction	C0155626	918.0	T047
1	41	17159	0.0	74	d	Genetic Variants and Functional Analyses of th...	Acute myocardial infarction	C0155626	918.0	T047
2	30	15358	103.0	79	P3	Lipid levels and risk of new-onset atrial fibr...	acute myocardial infarction	C0155626	918.0	T047
3	31	6429	0.0	5	P7	Association of serum sclerostin and osteoprote...	Acute myocardial infarction	C0155626	918.0	T047
4	32	27343	1.0	5	P0	Role of Intravascular Ultrasound-Guided Percut...	acute myocardial infarction	C0155626	918.0	T047
...	...	...	...	...	...	...	...	...	...	...
95	204	28352	18.0	74	P4	Effects of Volatile versus Total Intravenous A...	hypertension	C0020538	894.0	T047
96	197	15518	17.0	5	P4	Risk of cardiovascular events in men treated f...	hypertension	C0020538	894.0	T047
97	198	13338	57.0	21	P9	Efficacy of Low-Dose Oral Liquid Morphine for ...	hypertension	C0020538	894.0	T047
98	199	9142	2.0	21	P0	The associations of low birth weight with prim...	hypertension	C0020538	894.0	T047
99	200	18377	49.0	79	P3	Electrocardiographic Versus Echocardiographic ...	hypertension	C0020538	894.0	T047

100 rows × 10 columns

Se realizan expresiones de búsqueda para corpus de las distintas patologías encontradas, con el objetivo de encontrar signos, procedimientos y dispositivos médicos asociados. Se dejará una expresi+on por cada CUI de patología encontrado.

In [17]:
```
os.chdir(directory)
for i in df_dir.index:
    expresiones = pd.DataFrame()
    lista=[]
    g_dir=df_dir["directorio grupo"][i]
    os.chdir(g_dir)
    pats=pd.read_csv("patologias encontradas.csv", sep=';')
    pats = pats[pats["freq"]>25].drop_duplicates(subset=["CUI"])
    for j in pats.index:
        expr1="'"+str(pats["STR"][j])+"'"+" AND (signs OR diagnosis)"
        lista.append([pats["STR"][j],pats["CUI"][j],expr1])
        expr2="'"+str(pats["STR"][j])+"'"+" AND (tratment* OR therapeutic*)"
        lista.append([pats["STR"][j],pats["CUI"][j],expr2])
        expr3="'"+str(pats["STR"][j])+"'"+" AND ('medical device' OR 'medical devices' O
        lista.append([pats["STR"][j],pats["CUI"][j],expr3])
    expresiones = pd.DataFrame(data=lista, columns=["STR","CUI","Expresión"])
    expresiones = expresiones.drop_duplicates(subset=["CUI","Expresión"])
    expresiones.to_csv("expresiones de búsqueda-entidades.csv",sep=";")
```

---------- OBJETIVO 3 ---------- Hacer búsquedas de artículos relacionados con cada una de las patologías asociadas a los problemas ges dentro de cada grupo, para posteriormente realizar procesamiento de lenguaje natural para encontrar signos, procedimientos y dispositivos médicos asociados a cada patología

In [18]:
```
os.chdir(directory)
df_dir = pd.read_csv('directorios de grupos.csv',sep=";")
```

Función para encontrar papers asociados a las entidades relacionadas de cada patología

In [19]:
```
from Bio import Entrez
from pmc import PMCArticleSet

def search(n,term,g_dir):
    try:
        Entrez.email = 'herman.chinga@alumnos.uv.cl'
        busq=Entrez.esearch(db="pmc",
                            sort='relevance',
                            retmax='10',
                            retmode="null",
                            term=term)
        print("Buscando término ", term)

        results=Entrez.read(busq)
        ids = results["IdList"]
        print("Se encontraron ",len(ids), " papers")

        handle = Entrez.efetch(db='pmc',
                               id=ids,
                               rettype='full',
                               retmode='xml')
        xml = handle.read()

        articles = PMCArticleSet(xml).articles  # List of PMCArticle objects
        cont = 0
        for article in articles:
            if article.body != "":
                os.chdir(g_dir)
```

```
                nombre = str("Pat_"+n+"_P"+str(cont)+"_.txt")
                doc= open(nombre,"w",encoding='utf8')
                paper = str(article.ids)+"||\n"+str(article.title)+"||\n"+str(article.ab
                doc.write(paper)
                doc.close()
                cont = cont+1
            else:
                os.chdir(sin_cuerpo_dir)
                nombre = str("Pat_"+n+"_P"+str(cont)+"_.txt")
                doc= open(nombre,"w",encoding='utf8')
                paper = str(article.ids)+"||\n"+str(article.title)+"||\n"+str(article.ab
                doc.write(paper)
                doc.close()
                cont = cont+1
    except:
        metricas.append(["error en búsqueda de papers",term])
        print("error en búsqueda")
```

Se procede a adquirir el corpus de las entidades asociadas a cada patologí

In [20]:
```
start=time.time()
for i in df_dir.index:
    g_dir = df_dir["directorio grupo"][i]
    os.chdir(g_dir)
    cont=0
    expresiones = pd.read_csv("expresiones de búsqueda-entidades.csv",sep=";")
    for j in expresiones.index:
        cont+=1
        print("expresion ",cont," de ",len(expresiones))
        print("Grupo",df_dir["Grupo"][i],"buscando papers ", expresiones["Expresión"][j]
        iden = str(j)+'_'+str(expresiones["STR"][j]+'_')
        print(iden)
        search(iden,str(expresiones["Expresión"][j]),df_dir["directorio entidades"][i])
end=time.time()
metricas.append(["tiempo de descarga de papers entidades", (end-start)/60])
```

```
expresion  1  de  213
Grupo 1 buscando papers  'Acute myocardial infarction' AND (signs OR diagnosis)
0_Acute myocardial infarction_
Buscando término  'Acute myocardial infarction' AND (signs OR diagnosis)
Se encontraron  0  papers
expresion  2  de  213
Grupo 1 buscando papers  'Acute myocardial infarction' AND (tratment* OR therapeutic*)
1_Acute myocardial infarction_
Buscando término  'Acute myocardial infarction' AND (tratment* OR therapeutic*)
Se encontraron  0  papers
expresion  3  de  213
Grupo 1 buscando papers  'Acute myocardial infarction' AND ('medical device' OR 'medical
devices' OR device*
2_Acute myocardial infarction_
Buscando término  'Acute myocardial infarction' AND ('medical device' OR 'medical device
s' OR device*
Se encontraron  10  papers
expresion  4  de  213
Grupo 1 buscando papers  'hypertension' AND (signs OR diagnosis)
3_hypertension_
Buscando término  'hypertension' AND (signs OR diagnosis)
Se encontraron  0  papers
expresion  5  de  213
Grupo 1 buscando papers  'hypertension' AND (tratment* OR therapeutic*)
4_hypertension_
Buscando término  'hypertension' AND (tratment* OR therapeutic*)
Se encontraron  0  papers
expresion  6  de  213
Grupo 1 buscando papers  'hypertension' AND ('medical device' OR 'medical devices' OR de
```

```
vice*
5_hypertension_
Buscando término  'hypertension' AND ('medical device' OR 'medical devices' OR device*
Se encontraron  10  papers
expresion  7  de  213
Grupo 1 buscando papers  'obesity' AND (signs OR diagnosis)
6_obesity_
Buscando término  'obesity' AND (signs OR diagnosis)
Se encontraron  0  papers
expresion  8  de  213
Grupo 1 buscando papers  'obesity' AND (tratment* OR therapeutic*)
7_obesity_
Buscando término  'obesity' AND (tratment* OR therapeutic*)
Se encontraron  0  papers
expresion  9  de  213
Grupo 1 buscando papers  'obesity' AND ('medical device' OR 'medical devices' OR device*
8_obesity_
Buscando término  'obesity' AND ('medical device' OR 'medical devices' OR device*
Se encontraron  10  papers
expresion  10  de  213
Grupo 1 buscando papers  'Myocardial Infarction' AND (signs OR diagnosis)
9_Myocardial Infarction_
Buscando término  'Myocardial Infarction' AND (signs OR diagnosis)
Se encontraron  1  papers
expresion  11  de  213
Grupo 1 buscando papers  'Myocardial Infarction' AND (tratment* OR therapeutic*)
10_Myocardial Infarction_
Buscando término  'Myocardial Infarction' AND (tratment* OR therapeutic*)
Se encontraron  0  papers
expresion  12  de  213
Grupo 1 buscando papers  'Myocardial Infarction' AND ('medical device' OR 'medical devic
es' OR device*
11_Myocardial Infarction_
Buscando término  'Myocardial Infarction' AND ('medical device' OR 'medical devices' OR
device*
Se encontraron  10  papers
expresion  13  de  213
Grupo 1 buscando papers  'heart failure' AND (signs OR diagnosis)
12_heart failure_
Buscando término  'heart failure' AND (signs OR diagnosis)
Se encontraron  10  papers
expresion  14  de  213
Grupo 1 buscando papers  'heart failure' AND (tratment* OR therapeutic*)
13_heart failure_
Buscando término  'heart failure' AND (tratment* OR therapeutic*)
Se encontraron  10  papers
expresion  15  de  213
Grupo 1 buscando papers  'heart failure' AND ('medical device' OR 'medical devices' OR d
evice*
14_heart failure_
Buscando término  'heart failure' AND ('medical device' OR 'medical devices' OR device*
Se encontraron  10  papers
expresion  16  de  213
Grupo 1 buscando papers  'cardiovascular diseases' AND (signs OR diagnosis)
15_cardiovascular diseases_
Buscando término  'cardiovascular diseases' AND (signs OR diagnosis)
Se encontraron  10  papers
expresion  17  de  213
Grupo 1 buscando papers  'cardiovascular diseases' AND (tratment* OR therapeutic*)
16_cardiovascular diseases_
Buscando término  'cardiovascular diseases' AND (tratment* OR therapeutic*)
Se encontraron  10  papers
expresion  18  de  213
Grupo 1 buscando papers  'cardiovascular diseases' AND ('medical device' OR 'medical dev
ices' OR device*
17_cardiovascular diseases_
```

```
Grupo 1 buscando papers  'ischemic stroke' AND ('medical device' OR 'medical devices' OR
device*
197_ischemic stroke_
Buscando término  'ischemic stroke' AND ('medical device' OR 'medical devices' OR device
*
Se encontraron  10  papers
expresion  199  de  213
Grupo 1 buscando papers  'pneumonia' AND (signs OR diagnosis)
198_pneumonia_
Buscando término  'pneumonia' AND (signs OR diagnosis)
Se encontraron  0  papers
expresion  200  de  213
Grupo 1 buscando papers  'pneumonia' AND (tratment* OR therapeutic*)
199_pneumonia_
Buscando término  'pneumonia' AND (tratment* OR therapeutic*)
Se encontraron  0  papers
expresion  201  de  213
Grupo 1 buscando papers  'pneumonia' AND ('medical device' OR 'medical devices' OR devic
e*
200_pneumonia_
Buscando término  'pneumonia' AND ('medical device' OR 'medical devices' OR device*
Se encontraron  10  papers
expresion  202  de  213
Grupo 1 buscando papers  'endothelial dysfunction' AND (signs OR diagnosis)
201_endothelial dysfunction_
Buscando término  'endothelial dysfunction' AND (signs OR diagnosis)
Se encontraron  10  papers
expresion  203  de  213
Grupo 1 buscando papers  'endothelial dysfunction' AND (tratment* OR therapeutic*)
202_endothelial dysfunction_
Buscando término  'endothelial dysfunction' AND (tratment* OR therapeutic*)
Se encontraron  10  papers
expresion  204  de  213
Grupo 1 buscando papers  'endothelial dysfunction' AND ('medical device' OR 'medical dev
ices' OR device*
203_endothelial dysfunction_
Buscando término  'endothelial dysfunction' AND ('medical device' OR 'medical devices' O
R device*
Se encontraron  10  papers
expresion  205  de  213
Grupo 1 buscando papers  'metastases' AND (signs OR diagnosis)
204_metastases_
Buscando término  'metastases' AND (signs OR diagnosis)
Se encontraron  0  papers
expresion  206  de  213
Grupo 1 buscando papers  'metastases' AND (tratment* OR therapeutic*)
205_metastases_
Buscando término  'metastases' AND (tratment* OR therapeutic*)
Se encontraron  0  papers
expresion  207  de  213
Grupo 1 buscando papers  'metastases' AND ('medical device' OR 'medical devices' OR devi
ce*
206_metastases_
Buscando término  'metastases' AND ('medical device' OR 'medical devices' OR device*
Se encontraron  10  papers
expresion  208  de  213
Grupo 1 buscando papers  'valvular disease' AND (signs OR diagnosis)
207_valvular disease_
Buscando término  'valvular disease' AND (signs OR diagnosis)
Se encontraron  10  papers
expresion  209  de  213
Grupo 1 buscando papers  'valvular disease' AND (tratment* OR therapeutic*)
208_valvular disease_
Buscando término  'valvular disease' AND (tratment* OR therapeutic*)
Se encontraron  10  papers
expresion  210  de  213
```

```
Grupo 1 buscando papers  'valvular disease' AND ('medical device' OR 'medical devices' O
R device*
209_valvular disease_
Buscando término  'valvular disease' AND ('medical device' OR 'medical devices' OR devic
e*
Se encontraron  10  papers
expresion  211  de  213
Grupo 1 buscando papers  'Type 1 diabetes mellitus' AND (signs OR diagnosis)
210_Type 1 diabetes mellitus_
Buscando término  'Type 1 diabetes mellitus' AND (signs OR diagnosis)
Se encontraron  0  papers
expresion  212  de  213
Grupo 1 buscando papers  'Type 1 diabetes mellitus' AND (tratment* OR therapeutic*)
211_Type 1 diabetes mellitus_
Buscando término  'Type 1 diabetes mellitus' AND (tratment* OR therapeutic*)
Se encontraron  0  papers
expresion  213  de  213
Grupo 1 buscando papers  'Type 1 diabetes mellitus' AND ('medical device' OR 'medical de
vices' OR device*
212_Type 1 diabetes mellitus_
Buscando término  'Type 1 diabetes mellitus' AND ('medical device' OR 'medical devices'
OR device*
Se encontraron  10  papers
```

Eliminar artículos mayores a 100kBytes

In [21]:
```
os.chdir(directory)
papers_ids =[]
for i in df_dir.index:
    e_dir = df_dir["directorio entidades"][i]
    os.chdir(e_dir)
    papers = os.listdir(e_dir)
    total = len(papers)
    cont=0
    grupo=df_dir["Grupo"][i]
    metricas.append(["Cantidad de papers encontraados en grupo",str(grupo),total])
    for txt in papers:
        tamaño = os.path.getsize(txt)
        if tamaño > 100000:
            cont+=1
            os.remove(txt)
    metricas.append([str("Papers entidades eliminados por tamaño en grupo ")+str(grupo),
metricas
```

Out[21]:
```
[['tiempo de descarga de papers entidades', 18.78231030702591],
 ['Papers entidades eliminados por tamaño en grupo 1', 198]]
```

Eliminar duplicados

In [22]:
```
import ast
os.chdir(directory)
papers_ids =[]
for i in df_dir.index:
    e_dir = df_dir["directorio entidades"][i]
    os.chdir(e_dir)
    papers = os.listdir(e_dir)
    total = len(papers)
    cont=0
    grupo=df_dir["Grupo"][i]
    for txt in papers:
        arch = open(txt, "r",encoding='utf8')
        # print(txt)
        paper = arch.read()
        arch.close()
        paper = paper.split("||\n")
```

```
        #print(paper[0])
        ids = ast.literal_eval(paper[0])
        paper_id = ids["pmc"]
        if paper_id not in papers_ids:
            papers_ids.append(paper_id)
            #print(paper_id)
        else:
            os.remove(txt)
            cont+=1
    metricas.append([str("Papers entidades duplicados eliminados en grupo ")+str(grupo),
    # print(paper_id)
metricas
```

Out[22]:
```
[['tiempo de descarga de papers entidades', 18.78231030702591],
 ['Papers entidades eliminados por tamaño en grupo 1', 198],
 ['Papers entidades duplicados eliminados en grupo 1', 591]]
```

Procesamiento de lenguaje natural para encontrar signos, procedimientos y dispositivos médicos

In [23]:
```
os.chdir(directory)
from transformers import AutoTokenizer, BertForTokenClassification, pipeline
from nltk.corpus import stopwords
from medcat.cat import CAT

path = "/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db/tésis final/biobert_diseases_ner"
tokenizer = AutoTokenizer.from_pretrained(path,return_tensors='pt')
model =     BertForTokenClassification.from_pretrained(path)
ner = pipeline("ner", model=model, tokenizer=tokenizer,aggregation_strategy='max')

cat = CAT.load_model_pack('/media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db/tésis final/me
```

```
Found an existing unziped model pack at: /media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db/
tésis final/medmen_wstatus_2021_oct, the provided zip will not be touched.
The CDB was exported by an unknown version of MedCAT.
{
  "Model ID": null,
  "Last Modifed On": null,
  "History (from least to most recent)": [],
  "Description": "No description",
  "Source Ontology": null,
  "Location": null,
  "MetaCAT models": {},
  "Basic CDB Stats": {},
  "Performance": {
    "ner": {},
    "meta": {}
  },
  "Important Parameters (Partial view, all available in cat.config)": {
    "config.ner['min_name_len']": {
      "value": 3,
      "description": "Minimum detection length (found terms/mentions shorter than this w
ill not be detected)."
    },
    "config.ner['upper_case_limit_len']": {
      "value": 3,
      "description": "All detected terms shorter than this value have to be uppercase, o
therwise they will be ignored."
    },
    "config.linking['similarity_threshold']": {
      "value": 0.2,
      "description": "If the confidence of the model is lower than this a detection will
be ignore."
    },
    "config.general['spell_check']": {
      "value": true,
```

```
      "description": "Is spell checking enabled."
    },
    "config.general['spell_check_len_limit']": {
      "value": 7,
      "description": "Words shorter than this will not be spell checked."
    }
  },
  "MedCAT Version": null
}
```

In [39]:

```
### Función para encontrar patologías asociadas a cada cui del ges encontradas en la lit

def related_ents(name,pat,carp):
    #### ubicar en directorio de papers
    #### Lectura de papers
    os.chdir(carp)

    try:
        arch = open(name, "r", encoding='utf8')
        paper = arch.read() ## se guarda variable con texto
        arch.close()
        paper = paper.split("||\n")
        ##### caracteristicas del paper
        title = paper[1]


        ### procesamiento de texto
        sw_nltk = stopwords.words('english')
        cuerpo = paper[3]
        cuerpo = cuerpo.split(". ") #### N bins
        proc_text = []

        print("procesamiento de texto")
        for i in range(len(cuerpo)):
            if len(cuerpo[i].split()) > 10:
                words=[]
                words = [word for word in cuerpo[i].split() if word.lower() not in sw_nl
                cuerpo[i] = " ".join(words)
                proc_text.append(cuerpo[i])
        print("texto procesado, eliminadas frases cortas y stop words")
        print("NER - signos con pipeline transformers")
        ents_ner=[]
        #### METODO NER
        for parrafo in proc_text:
            for entity in ner(parrafo): ###3 SE DETECTAN PATOLOGÍAS ASOCIADAS CON METODO
                if entity['entity_group'] == 'DISEASE':
                    ents_ner.append(entity["word"])

        #### Extraccion de signos detectados con NER
        print("identificacion de cuis con Medcat en entidades reconocidas por pipeline")
        signs=[]
        for ent in ents_ner:
            entity = cat.get_entities(ent)
            entity.pop("tokens")
            entity = entity["entities"]
            for clave, valor in entity.items():
                    if str(valor["type_ids"]) == "['T033']" or str(valor["type_ids"]) ==
                    signs.append(valor["cui"])
              # ents.append([valor["cui"],valor["pretty_name"],valor["type_ids"],valor["
        print("signos ok", len(signs))

        ### Extraccion de signos, tratamientos y dispositivos médicos
        print("identificacion de cuis con medcat en texto sin pipeline")
        trats = []
        devs = []
        for parrafo in proc_text:
```

```
            entity = cat.get_entities(parrafo)
            # print(entity)
            entity.pop("tokens")
            entity = entity["entities"]
            for clave, valor in entity.items():
              ######  SIGNOS
              # print(valor["type_ids"],valor["pretty_name"],valor['detected_name'],valo
              #### T020|Acquired Abnormality ##### |T190|Anatomical Abnormality ##### T0
                if str(valor["type_ids"]) == "['T020']" or str(valor["type_ids"]) == "['
                    signs.append(valor["cui"])
                  # print(valor["type_ids"],valor["pretty_name"],valor['detected_name'])

              # #######  TRATAMIENTOS
              ###############  T060|Diagnostic Procedure   ############# T058|Health Car
                if str(valor["type_ids"]) == "['T060']" or str(valor["type_ids"]) == "['
                    trats.append(valor["cui"])
                  # print(valor["detected_name"], valor["acc"])

              # #######  MEDICAL DEVICES
              ##########  T203|Drug Delivery Device              ##### T074|Medical Dev
                if str(valor["type_ids"]) == "['T203']" or str(valor["type_ids"]) == "['
                    devs.append(valor["cui"])
        ents = signs+trats+devs
        print("Signos añadidos, Procedimientos ok", len(trats), "Dispositivos medicos ok
        df_ents = pd.DataFrame()
        df_ents["STR"]=[pat]*len(ents)
        df_ents["title"] =[title]*len(ents)
        df_ents["CUI"] = ents
        return(df_ents)

    except:
        metricas.append(["Error en lectura",name])
        print("error de lectura")
```

In [40]:

```
start = time.time()
os.chdir(directory)
for i in df_dir.index:
    g_dir = df_dir["directorio grupo"][i] ##entra a directorio de grupo
    corpus_ents_dir= df_dir["directorio entidades"][i]
    papers_list = os.listdir(corpus_ents_dir)
    cont=0
    for name in papers_list:
        cont+=1
        os.chdir(corpus_ents_dir)
        print("leyendo paper", name)
        print("va en ",cont,"de ",len(papers_list))
        dname = name.split('_')
        print(dname)
        pat = dname[2]  # Datos
        os.chdir(g_dir)
        ents=pd.DataFrame()
        ents = related_ents(name,pat,corpus_ents_dir)
        print("guardando entidades en",g_dir)
        os.chdir(g_dir)
        ents.to_csv("pre-entidades.csv",sep=";",mode="a")
end = time.time()
metricas.append(["Búsqueda de entidades",(end-start)/60])
```

```
leyendo paper Pat_71_atherosclerosis__P4_.txt
va en  1 de  266
['Pat', '71', 'atherosclerosis', '', 'P4', '.txt']
procesamiento de texto
texto procesado, eliminadas frases cortas y stop words
NER - signos con pipeline transformers
identificacion de cuis con Medcat en entidades reconocidas por pipeline
```

```
signos ok 3
identificacion de cuis con medcat en texto sin pipeline
Signos añadidos, Procedimientos ok 203 Dispositivos medicos ok 60
guardando entidades en /media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db/trabajo de título
Herman Chinga/Grupo 1 Problemas Cardiovasculares
leyendo paper Pat_81_Pericardial effusion__P8_.txt
va en  2 de  266
['Pat', '81', 'Pericardial effusion', '', 'P8', '.txt']
procesamiento de texto
texto procesado, eliminadas frases cortas y stop words
NER - signos con pipeline transformers
identificacion de cuis con Medcat en entidades reconocidas por pipeline
signos ok 19
identificacion de cuis con medcat en texto sin pipeline
Signos añadidos, Procedimientos ok 139 Dispositivos medicos ok 3
guardando entidades en /media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db/trabajo de título
Herman Chinga/Grupo 1 Problemas Cardiovasculares
leyendo paper Pat_66_aortic stenosis__P2_.txt
va en  3 de  266
['Pat', '66', 'aortic stenosis', '', 'P2', '.txt']
procesamiento de texto
texto procesado, eliminadas frases cortas y stop words
NER - signos con pipeline transformers
identificacion de cuis con Medcat en entidades reconocidas por pipeline
signos ok 16
identificacion de cuis con medcat en texto sin pipeline
Signos añadidos, Procedimientos ok 337 Dispositivos medicos ok 1
guardando entidades en /media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db/trabajo de título
Herman Chinga/Grupo 1 Problemas Cardiovasculares
leyendo paper Pat_127_mitral stenosis__P4_.txt
va en  4 de  266
['Pat', '127', 'mitral stenosis', '', 'P4', '.txt']
procesamiento de texto
texto procesado, eliminadas frases cortas y stop words
NER - signos con pipeline transformers
identificacion de cuis con Medcat en entidades reconocidas por pipeline
signos ok 72
identificacion de cuis con medcat en texto sin pipeline
Signos añadidos, Procedimientos ok 202 Dispositivos medicos ok 9
guardando entidades en /media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db/trabajo de título
Herman Chinga/Grupo 1 Problemas Cardiovasculares
leyendo paper Pat_50_influenza__P9_.txt
va en  5 de  266
['Pat', '50', 'influenza', '', 'P9', '.txt']
procesamiento de texto
texto procesado, eliminadas frases cortas y stop words
NER - signos con pipeline transformers
identificacion de cuis con Medcat en entidades reconocidas por pipeline
signos ok 0
identificacion de cuis con medcat en texto sin pipeline
Signos añadidos, Procedimientos ok 96 Dispositivos medicos ok 44
guardando entidades en /media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db/trabajo de título
Herman Chinga/Grupo 1 Problemas Cardiovasculares
leyendo paper Pat_62_ischaemic heart disease__P0_.txt
va en  6 de  266
['Pat', '62', 'ischaemic heart disease', '', 'P0', '.txt']
procesamiento de texto
texto procesado, eliminadas frases cortas y stop words
NER - signos con pipeline transformers
identificacion de cuis con Medcat en entidades reconocidas por pipeline
signos ok 4
identificacion de cuis con medcat en texto sin pipeline
Signos añadidos, Procedimientos ok 87 Dispositivos medicos ok 105
guardando entidades en /media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db/trabajo de título
Herman Chinga/Grupo 1 Problemas Cardiovasculares
leyendo paper Pat_92_valvular stenosis regurgitation__P6_.txt
```

```
va en  260 de  266
['Pat', '148', 'SARS - CoV - 2 infection', '', 'P6', '.txt']
procesamiento de texto
texto procesado, eliminadas frases cortas y stop words
NER - signos con pipeline transformers
identificacion de cuis con Medcat en entidades reconocidas por pipeline
signos ok 20
identificacion de cuis con medcat en texto sin pipeline
Signos añadidos, Procedimientos ok 129 Dispositivos medicos ok 20
guardando entidades en /media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db/trabajo de título
 Herman Chinga/Grupo 1 Problemas Cardiovasculares
leyendo paper Pat_133_dengue hemorrhagic fever__P9_.txt
va en  261 de  266
['Pat', '133', 'dengue hemorrhagic fever', '', 'P9', '.txt']
procesamiento de texto
texto procesado, eliminadas frases cortas y stop words
NER - signos con pipeline transformers
identificacion de cuis con Medcat en entidades reconocidas por pipeline
signos ok 15
identificacion de cuis con medcat en texto sin pipeline
Signos añadidos, Procedimientos ok 291 Dispositivos medicos ok 1
guardando entidades en /media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db/trabajo de título
 Herman Chinga/Grupo 1 Problemas Cardiovasculares
leyendo paper Pat_123_Tricuspid Regurgitation__P5_.txt
va en  262 de  266
['Pat', '123', 'Tricuspid Regurgitation', '', 'P5', '.txt']
procesamiento de texto
texto procesado, eliminadas frases cortas y stop words
NER - signos con pipeline transformers
identificacion de cuis con Medcat en entidades reconocidas por pipeline
signos ok 1
identificacion de cuis con medcat en texto sin pipeline
Signos añadidos, Procedimientos ok 8 Dispositivos medicos ok 1
guardando entidades en /media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db/trabajo de título
 Herman Chinga/Grupo 1 Problemas Cardiovasculares
leyendo paper Pat_154_asthma eczema__P7_.txt
va en  263 de  266
['Pat', '154', 'asthma eczema', '', 'P7', '.txt']
procesamiento de texto
texto procesado, eliminadas frases cortas y stop words
NER - signos con pipeline transformers
identificacion de cuis con Medcat en entidades reconocidas por pipeline
signos ok 22
identificacion de cuis con medcat en texto sin pipeline
Signos añadidos, Procedimientos ok 102 Dispositivos medicos ok 2
guardando entidades en /media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db/trabajo de título
 Herman Chinga/Grupo 1 Problemas Cardiovasculares
leyendo paper Pat_178_lacunar infarcts__P6_.txt
va en  264 de  266
['Pat', '178', 'lacunar infarcts', '', 'P6', '.txt']
procesamiento de texto
texto procesado, eliminadas frases cortas y stop words
NER - signos con pipeline transformers
identificacion de cuis con Medcat en entidades reconocidas por pipeline
signos ok 4
identificacion de cuis con medcat en texto sin pipeline
Signos añadidos, Procedimientos ok 252 Dispositivos medicos ok 8
guardando entidades en /media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db/trabajo de título
 Herman Chinga/Grupo 1 Problemas Cardiovasculares
leyendo paper Pat_12_heart failure__P6_.txt
va en  265 de  266
['Pat', '12', 'heart failure', '', 'P6', '.txt']
procesamiento de texto
texto procesado, eliminadas frases cortas y stop words
NER - signos con pipeline transformers
identificacion de cuis con Medcat en entidades reconocidas por pipeline
```

```
signos ok 0
identificacion de cuis con medcat en texto sin pipeline
Signos añadidos, Procedimientos ok 143 Dispositivos medicos ok 2
guardando entidades en /media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db/trabajo de título
 Herman Chinga/Grupo 1 Problemas Cardiovasculares
leyendo paper Pat_196_ischemic stroke__P3_.txt
va en  266 de  266
['Pat', '196', 'ischemic stroke', '', 'P3', '.txt']
procesamiento de texto
texto procesado, eliminadas frases cortas y stop words
NER - signos con pipeline transformers
identificacion de cuis con Medcat en entidades reconocidas por pipeline
signos ok 2
identificacion de cuis con medcat en texto sin pipeline
Signos añadidos, Procedimientos ok 77 Dispositivos medicos ok 0
guardando entidades en /media/uv/de2b20cc-8d6e-42d4-bda1-1606ac6745db/trabajo de título
 Herman Chinga/Grupo 1 Problemas Cardiovasculares
```

In [41]:
```
start = time.time()
os.chdir(directory)
for i in df_dir.index:
    try:
        g_dir = df_dir["directorio grupo"][i] ##entra a directorio de grupo
        os.chdir(g_dir)
        entities=pd.read_csv("pre-entidades.csv",sep=";")
        df2_ents=entities.groupby(by="CUI")["STR"].describe()
        df2_ents=df2_ents["freq"]
        entities = pd.merge(entities,df2_ents, on="CUI",how="left").drop_duplicates(subs
        entities.to_csv("entidades encontradas.csv", sep=';')
    except:
        print('error')
end = time.time()
metricas.append(["Procesamiento de entidades",(end-start)/60])
```

In [42]:
```
ents
```

Out[42]:

	STR	title	CUI
0	ischemic stroke	Statistical analysis plan for the 'Triple Anti...	C0019080
1	ischemic stroke	Statistical analysis plan for the 'Triple Anti...	C0019080
2	ischemic stroke	Statistical analysis plan for the 'Triple Anti...	C2825055
3	ischemic stroke	Statistical analysis plan for the 'Triple Anti...	C0019080
4	ischemic stroke	Statistical analysis plan for the 'Triple Anti...	C0032176
...	...	...	...
81	ischemic stroke	Statistical analysis plan for the 'Triple Anti...	C0920317
82	ischemic stroke	Statistical analysis plan for the 'Triple Anti...	C0008976
83	ischemic stroke	Statistical analysis plan for the 'Triple Anti...	C0008976
84	ischemic stroke	Statistical analysis plan for the 'Triple Anti...	C0087111
85	ischemic stroke	Statistical analysis plan for the 'Triple Anti...	C0679575

86 rows × 3 columns

In [ ]:
```
---------------- OBJETIVO 4 ------------------------------------------------
4.4      GENERAR NODOS Y LAZOS PARA FORMAR LA RED DE INFORMACIÓN.
```

Normalización de datos para generar las 4 entidades

In [66]:
```
def entidad(string):
    pats =['T047','T191','T050']
    signs = ['T033','T046','T020','T190','T049','T019','T184','T201','T043','T037']
    proc = ['T060','T059','T063','T062','T061','T058']
    devs = ['T203','T074','T075']
    if string in pats:
        return('Patología')
    if string in signs:
        return('Signo')
    if string in proc:
        return('Procedimiento')
    if string in devs:
        return('Dispositivo médico')
```

In [67]:
```
start = time.time()
os.chdir(directory)
conso = pd.read_csv("umls_conso.csv",sep=";")
conso=conso[["CUI","LAT","STR"]]
conso=conso[conso["LAT"]=="SPA"]

sty=pd.read_csv("umls_sty.csv", sep=";")
sty=sty[["CUI","STY","TUI"]]

for i in df_dir.index:

    g_dir = df_dir["directorio grupo"][i]
    os.chdir(g_dir)
    pats=pd.read_csv("patologias encontradas.csv",sep=";")
    pats=pats.rename(columns={'STR':'PAT'})
    pats=pats[["NÂ°","title","PAT","CUI","freq"]]
    pats=pd.merge(pats,conso,on="CUI",how="inner").drop_duplicates(subset=["NÂ°","title"
    pats=pd.merge(pats,sty,on="CUI",how="inner")
    pats['Entidad']=pats['TUI'].apply(lambda x: entidad(x))
    pats.to_csv("Entidades - patologías.csv",sep=";")

    ents=pd.read_csv("entidades encontradas.csv",sep=";")
    ents = ents.rename(columns={'STR':'PAT'})
    ents=pd.merge(ents,conso,on="CUI",how="inner").drop_duplicates(subset=["PAT","title"
    ents=pd.merge(ents,sty,on="CUI",how="left")
    ents.to_csv("entidades para red.csv",sep=";")
    ents['Entidad']=ents['TUI'].apply(lambda x: entidad(x))
    ents.to_csv("Entidades - signos procedimientos y dispositivos.csv",sep=";")

end = time.time()
metricas.append(["Normalización de entidades",(end-start)/60])
```

A continuación se programan los lazos y nodos para guardar el grafo de información correspondiente

In [71]:
```
import pandas as pd
import os
import networkx as nx
import os
import numpy as np

G = nx.Graph()

directory =r'C:\Users\Lenovo\Desktop\TESIS FINAL FINAL'
ents=pd.read_csv("Entidades - signos procedimientos y dispositivos.csv",sep=";")
os.chdir(directory)
conso = pd.read_csv("umls_conso.csv",sep=";")
conso=conso[["CUI","LAT","STR"]]
conso=conso[conso["LAT"]=="SPA"].drop_duplicates(subset="CUI")
```

```
#nodos
pats = pd.read_csv("Entidades - patologías.csv", sep=";")
pats=pats.drop(["STR","LAT"],axis=1)
pats=pats.rename(columns={'NÂ°':'N°'})

pats=pd.merge(pats,conso,on="CUI",how="inner")
x=pats["freq"].dropna()
Q1 = np.percentile(x,25)
pats=pats[pats["freq"]>Q1]
nodos_pats=pats[["STR","CUI","freq","STY","Entidad"]].drop_duplicates()
nodos_dict_pats = nodos_pats.set_index(["STR"]).T.to_dict("dict")
nodos=[]
for key, value in nodos_dict_pats.items():
    nodos.append((key, value))
G.add_nodes_from(nodos)

probs=pd.read_excel("Categorización 85 problemas.xlsx")
nprobs=probs["Problema de salud"]+" en "+probs["Población objetivo"].to_list()
probs["Problemas"] = nprobs
probs=probs[["Problemas","N°","CUI","Problema de salud","Población objetivo","Grupo","No
probs = probs[probs["Grupo"]==1]
nodos = []
nodos_dict_probs = probs.set_index(["Problemas"]).T.to_dict("dict")
for key, value in nodos_dict_probs.items():
    nodos.append((key, value))
    G.add_nodes_from(nodos)
##links
links=[]
pats2=pd.merge(pats[["N°","title","STR"]],probs[["Problemas","N°"]],on="N°")
for j in pats2.index:
    atribute={'artículo':pats2["title"][j]}
    links.append((pats2["Problemas"][j],pats2["STR"][j],atribute))
G.add_edges_from(links)
nx.write_gexf(G, "Red de informacion2.gexf")


nodos = []
ents=pd.read_csv("Entidades - signos procedimientos y dispositivos.csv",sep=";")
ents=ents.drop(["STR","LAT"],axis=1)
ents=pd.merge(ents,conso,on="CUI",how="inner")
nodos_dict_ents= ents[["STR","Entidad","freq","STY"]].set_index(["STR"]).T.to_dict("dict
for key, value in nodos_dict_ents.items():
    nodos.append((key, value))
    G.add_nodes_from(nodos)
##links
pats=pats.rename(columns={'STR_esp':'STR_pat'})
ents2=pd.merge(pats[["PAT","STR_pat"]],ents[["PAT","title","STR","STY","Entidad","freq"]
links=[]
for j in ents2.index:
    atribute={'artículo':ents2["title"][j]}
    links.append((ents2["STR_pat"][j],ents2["STR"][j],atribute))
G.add_edges_from(links)
nx.write_gexf(G, "Red de informacion3.gexf")
```

```
/tmp/ipykernel_7424/533920610.py:36: UserWarning: DataFrame columns are not unique, some
columns will be omitted.
  nodos_dict_ents=nodos_ents.set_index(['STR']).T.to_dict('dict')
```