



Proyecto de Fin de Carrera  
Julio 2010  
M<sup>a</sup>Jesús Portolés Sánchez

# Búsqueda Semántica en Repositorios de Conceptos Biomédicos Estandarizados:

**C.T.Hunter**

Directores:  
Montserrat Robles Viejo  
Carlos Angulo Fernandez



**Indra**





# Índice

- 1. Introducción**
- 2. Objetivos**
- 3. Estado del Arte**
  - 3.1. Definiciones Importantes**
  - 3.2. Herramientas de Acceso a Terminologías**
- 4. C.T.Hunter**
  - 4.1. SNOMED CT**
  - 4.2. UMLS**
  - 4.3. CTS**
  - 4.4. LexGrid y LexBIG**
  - 4.5. C.T.Hunter**
    - 4.5.1. Base de Datos**
    - 4.5.2. Consultas Básicas**
    - 4.5.3. La Interfaz Gráfica**
    - 4.5.4. Consultas Semánticas**
- 5. Resultados y Conclusiones**
- 6. Trabajo Futuro**
- 7. Agradecimientos**
- 8. Glosario**
- 9. Bibliografía**

## 1. Introducción

Hoy en día la información sanitaria de un paciente está dispersa entre diferentes organizaciones, principalmente entre atención primaria y atención especializada, pero también entre diferentes hospitales y servicios de salud regionales. Como consecuencia los recursos de datos están distribuidos y son heterogéneos, creando un gran desfase entre el valor potencial y el valor real de la información presente en los sistemas de Historia Clínica Electrónica (HCE). Eliminar ese desfase haciendo un uso eficiente de los datos presentes en estos sistemas puede mejorar significativamente el cuidado del paciente, la eficiencia clínica, la seguridad del paciente y mejorar las actividades de investigación clínica.

Conseguir un historial clínico electrónico que incluya toda la información sanitaria existente sobre un paciente con independencia del lugar donde haya sido generada o dónde se encuentre almacenada y mantenida requiere la interoperabilidad semántica de los sistemas de HCE. La interoperabilidad semántica es la capacidad de los sistemas informáticos de comunicar, incorporar y usar información generada por sistemas externos. Para que esa información tenga un significado claro, no ambiguo, es necesario describirla mediante anotaciones terminológicas que sean accesibles desde ambos sistemas, el que generó la información y el que la recibe. El enlace terminológico entre los datos y las terminologías clínicas es un requisito previo para conseguir interoperabilidad semántica en cualquier dominio. El intercambio de datos debe hacerse de forma significativa, evitando cualquier posibilidad de error o mala interpretación, asegurando que la información conserva al ser recibida el significado original asignado cuando fue generada.

Sin embargo, a lo largo de los años, han proliferado sistemas diferentes no sólo entre centros de países distintos, sino incluso entre diferentes centros de una misma zona, y diferentes

departamentos dentro de un mismo centro. La existencia de sistemas distintos diseñados sin coordinación implica diferente modelo para la representación de la información, y la enorme amplitud del dominio médico ha dado lugar a su vez a multitud de terminologías diferentes diseñadas para su aplicación en sub-dominios concretos.

Cada terminología abarca un dominio y está diseñada para el uso en un ámbito distinto; son las terminologías de dominio. No existe una terminología universal, ya que no existe una estructura universal que permita representar completamente todo el conocimiento, incluso si nos limitamos a tratar únicamente con el vocabulario médico. Además, una terminología de tal magnitud resultaría inmanejable, debido al volumen de datos necesario para abarcar todo el dominio completo que además no deja de crecer.

Se acepta que la mejor solución al problema de la interoperabilidad semántica es emplear una terminología de referencia común con la que todos los sistemas puedan comunicarse, manteniendo al mismo tiempo sus propias terminologías locales intactas y generando mapeos (correspondencias) entre éstas y aquella.

Fue bajo la premisa de crear esta “terminología de referencia” [1] por lo que el Colegio de Patólogos Americanos inició la terminología SNOMED RT, el punto de partida para la actual SNOMED CT [2]. Esta terminología abarca un amplio espectro del dominio clínico. La arquitectura de SNOMED-CT, formalmente diseñada en jerarquías de conceptos relacionadas entre sí cuya implementación se valida mediante técnicas de lógica descriptiva [3] que permite la deducción de conocimiento nuevo a partir del conocimiento ya existente. Actualmente, 15 países, entre ellos España, pertenecen a la asociación IHTSDO [4], que se ocupa del desarrollo, revisión, mejora, mantenimiento e implantación de SNOMED-CT.

Sin embargo, SNOMED sólo se distribuye como terminología, sin apenas mapeos definidos con otras terminologías ni

herramientas propias de búsqueda semántica.

Para poder integrar y enlazar diferentes terminologías existe UMLS (Unified Medical Language System) [5], un proyecto que se ocupa de reunir las terminologías médicas más utilizadas, enlazarlas y distribuirlas, con el objetivo de contribuir al desarrollo de herramientas capaces de sacar partido al conocimiento formalizado en ellas.

De todas las herramientas de búsqueda terminológica disponibles actualmente, pocas aprovechan todo el potencial de las terminologías. Existen muchos buscadores de términos de SNOMED (SNOB [6], SnoFlake [7], CliniClue [8], etc.) que no aprovechan las características jerárquicas de la terminología para hacer búsquedas avanzadas, ni permiten su enlace con otras terminologías.

Otras herramientas, como Bioportal [9], UMLS o CTS [10], se basan en la idea de almacenar distintas terminologías bajo una misma distribución para permitir acceder a cualquiera de ellas mediante funciones genéricas de búsqueda. Al igual que las anteriores, estas funciones genéricas no sacan partido a las particularidades únicas de las terminologías a las que acceden para facilitar las búsquedas.

La herramienta presentada en este trabajo pretende, empleando tecnologías de probada utilidad (UMLS, CTS y LexGrid [11]), acceder a la terminología SNOMED CT extraída a partir de la distribución disponible en UMLS y realizar consultas avanzadas sobre la misma aprovechando las características propias de esta terminología.

## 2. Objetivos

El objetivo principal de este trabajo de final de carrera es investigar los medios actuales de acceso y búsqueda de datos de terminologías para, una vez evaluadas las opciones, implementar una herramienta capaz de acceder a una terminología SNOMED-CT, almacenada en un sistema gestor de bases de datos relacional y realizar consultas sobre ella.

La parte principal de la herramienta a implementar serán las funciones de consulta, que deberían poderse publicar como servicio web para poder ser accedidas y utilizadas por otras herramientas.

Las funciones avanzadas a implementar permitirán realizar búsquedas semánticas complejas, por tipos de términos y atributos. Para probar la funcionalidad de estas funciones, se implementó también una Interfaz Gráfica de Usuario (IGU).

### **3. Estado del Arte**

Dado que el objetivo principal de este proyecto es implementar un servidor de términos y unas herramientas de acceso al mismo, el estado del arte se ha planteado en dos apartados: un primer apartado para definir los conceptos iniciales importantes que deben quedar claros desde el principio, y un segundo apartado para la revisión de herramientas libres actuales para el acceso a terminologías. Este último apartado no habla de las herramientas que se han empleado finalmente en el proyecto, ya que cada una de ellas es tratada en su propio capítulo.

#### **3.1. Definiciones Importantes:**

Previo a la presentación de este proyecto hemos de definir algunos términos importantes.

Una terminología es una representación del conocimiento mediante vocabularios controlados, un subconjunto del lenguaje natural con significados normalizados.

En 1998, J. Cimino [12] definió las características deseables que debían reunir estos vocabularios controlados: amplitud de contenido, orientación a conceptos con definiciones lógicas formales susceptibles de ser interpretadas por un ordenador. Cada concepto debería tener un significado (no ser vago) y sólo uno (no ser ambiguo), mientras que un significado no debería estar representado por más de un concepto (no debería ser redundante). Barry Smith ofreció un punto de vista diferente planteando el cambio de paradigma de la orientación a conceptos a la orientación a *universales* que recojan las características de instancias concretas de la realidad en biomedicina [13].

Ejemplos de terminologías de términos clínicos son SNOMED CT, LOINC [14], ICD [15], MeSH [16], etc.



Ontología es la conceptualización de una realidad, o sea una idealización, mediante un conjunto de términos estructurados jerárquicamente que describen y clasifican las instancias concretas de un dominio incluyendo la conceptualización de las relaciones entre ellas, que puede usarse como fundamento para una base de conocimiento. La ontología atañe a la definición de la estructura de un dominio, no a los datos o conocimientos contenidos siguiendo esa estructura. En ese sentido, todas las terminologías organizadas tienen, por definición, una ontología más o menos compleja por encima, al menos de forma inherente. Así, mientras las terminologías recogen el mundo de las instancias o realidades las ontologías definen el dominio en el que estas realidades cobran sentido, y, por tanto, es difícil concebir ambas construcciones por separado.

Arquetipo, nombre dado en la norma EN13606 (similares en su función a las plantillas -o en inglés *templates*- en HL7 v3 [17]), es una estructura que permite definir los conceptos y estructuras de información de alto nivel que se manejan en un sistema de información. Estas estructuras son composiciones y agregaciones de otras estructuras más simples que dan soporte a la información concreta del dominio que han de describir, que vienen definidas en un modelo de datos de referencia, o sencillamente *modelo de referencia*, para dicho dominio concreto. Las normas y metodología de creación de arquetipos sigue el paradigma de lo que se denomina *Modelo Dual* de desarrollo, que consiste en la separación en dos niveles de las estructuras que darán soporte por un lado a la información concreta de cada dominio (los modelos de referencia) y por otro al conocimiento existente sobre dicho dominio (los propios arquetipos). La definición de arquetipo o plantilla debe ser fiel a las necesidades o restricciones médicas y permite abstraerse de las características técnicas del modelo de referencia subyacente.

Búsqueda Semántica se refiere a una búsqueda de conceptos no sólo por comparación de palabras o sintagmas (búsqueda sintáctica), sino por deducciones lógicas. Una búsqueda por palabras permite, por

ejemplo, buscar conceptos cuyo nombre contenga la palabra “fractura”. Una búsqueda semántica podría buscar conceptos que contengan “fractura” y que además “esté localizada en el hueso tarso”. Este tipo de búsquedas sólo pueden hacerse con apoyo sobre terminologías estructuradas donde pueda tenerse en cuenta la variedad de atributos y sus posibles valores tanto como las relaciones y tipos de relación posible entre sus conceptos, dotando al sistema informático de una base de relaciones lógicas sobre la cual filtrar y depurar los resultados de las búsquedas.

### 3.2. Herramientas de Acceso a Terminologías:

Otro de los objetivos de este proyecto es investigar y aprovechar, en la medida de lo posible, las herramientas de software libre ya implementadas para el acceso a terminologías.

La mayoría de estas herramientas son buscadores en red, como SnoCAT [18], SnoFlake o el buscador de VetMed [19]. Estas páginas son, fundamentalmente, formularios de introducción de palabras de búsqueda para encontrar conceptos según su nombre o código, que acceden a su propio repositorio o base de datos terminológica para realizar las búsquedas. Pueden mostrar los datos de los conceptos encontrados, pero no pueden realizar búsquedas por atributos ni filtrar resultados por jerarquías. Aunque son útiles y rápidas, carecen de modos de búsqueda avanzados en los que la semántica pueda realizar filtrados lógicos y no permiten el acceso a sus servidores desde herramientas externas. SnoFlake requiere registrarse, gratuitamente, antes de poder usarse.

El browser de CliniClue (ver fig. 1) es una de las herramientas descargables vía web gratuitas más conocidas. Requiere registro en línea, pero también ofrece una versión de prueba de 10 días sin registro. El programa obtiene la terminología SNOMED mediante descarga desde un servidor de actualizaciones propio, y permite que el usuario seleccione la distribución que le resulte más conveniente. No sólo permite búsquedas, también permite la navegación de SNOMED mediante un visor en el que se puede, interactivamente, *saltar* a través de sus términos siguiendo las relaciones que aplican a cada uno de los términos.

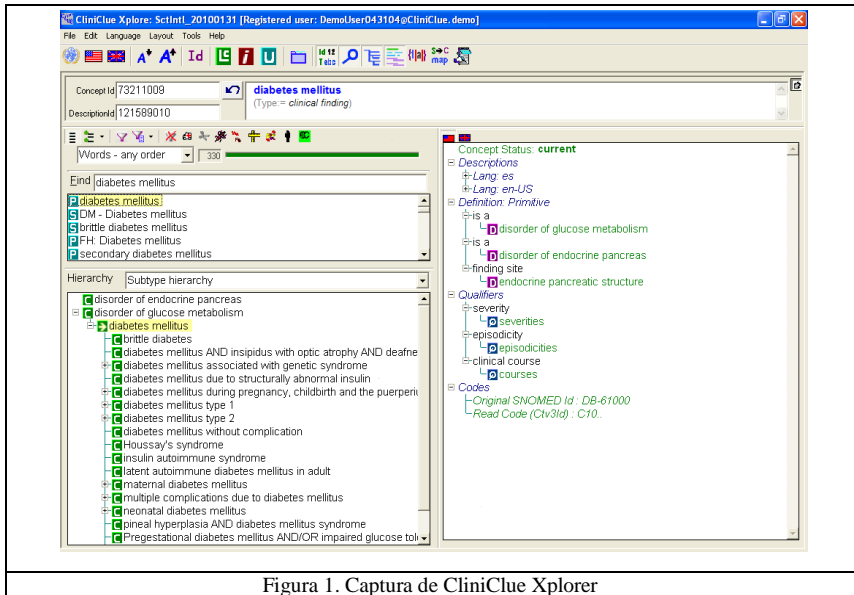


Figura 1. Captura de CliniClue Xplorer

Una ventana-visor en la parte inferior izquierda de la interfaz muestra inicialmente las principales jerarquías de SNOMED. Pulsando el botón junto a cada jerarquía se despliega un sub-menú con los conceptos que cuelgan directamente de ella y, a su vez, de cada concepto se pueden desplegar sus conceptos “hijo”.

SNOB (SNOMED Browser), otro programa gratuito de búsqueda, requiere cargar la terminología SNOMED desde los archivos de una distribución oficial obtenida de antemano. Además de las funciones habituales, permite editar conceptos y guardar los cambios en el mismo formato que la distribución oficial, lo que la convierte en una herramienta interesante para la traducción de términos o simplemente para crear extensiones locales de SNOMED.

Ambas herramientas, CliniClue Xplorer y SNOB, ofrecen también servicios web de búsqueda sobre sus respectivos servidores de terminología. Además de las búsquedas por palabras que ya ofrecen los buscadores en red, ambos permiten filtrar los resultados por jerarquías. SNOB permite, además, introducir filtros o “reglas” (ver fig. 2) para incluir valores de atributos en los parámetros de búsqueda (por ejemplo, buscar sólo conceptos en la jerarquía “Procedimiento” que tengan un atributo “Con uso de dispositivo” cuyo valor sea “Endoscopio”). Sin embargo, ninguna permite el uso de otras terminologías, ni el acceso de herramientas externas a sus funciones.

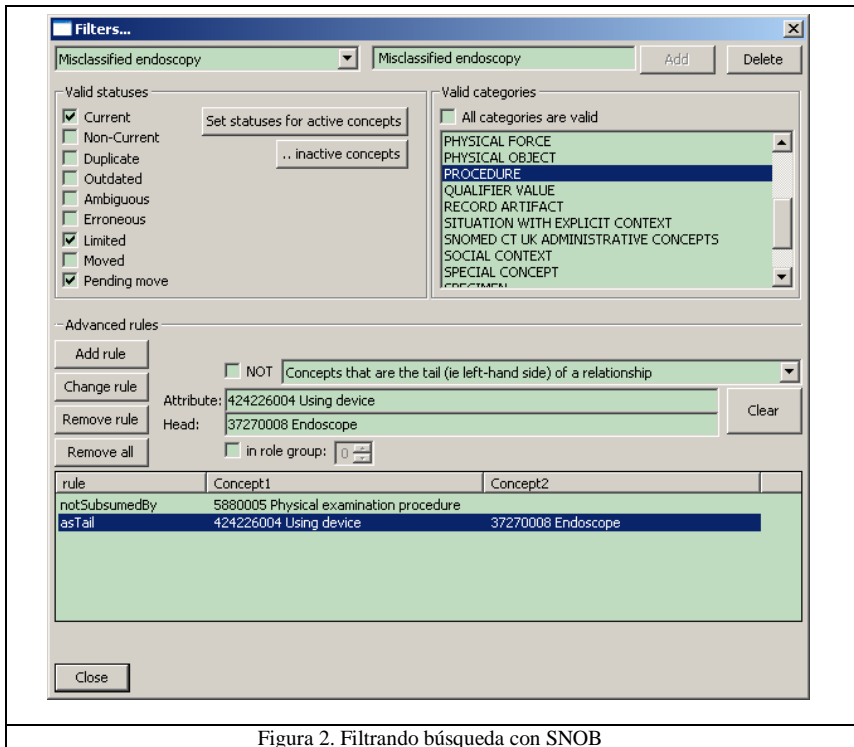


Figura 2. Filtrando búsqueda con SNOB

SnoCode de MedSight [20], una herramienta de pago, permite seleccionar texto en MSWord y codificar los términos médicos con su código SNOMED, entre otras funciones útiles como mostrar el nombre del término relacionado con un código al seleccionarlo. Sin embargo, SnoCode tan solo ofrece búsquedas tipo diccionario de Word, y no está abierto a búsquedas más complejas, ni a su uso en programas externos.

La herramienta Protegé [21] para edición y consulta de ontologías puede usarse, en teoría, para cargar SNOMED, ya que es una terminología que ha sido desarrollada y validada mediante lógica descriptiva, por lo que es internamente coherente en cuanto a jerarquías (relaciones *IS-A*) y a relaciones inter-jerárquicas entre términos. Sin embargo, el proceso para hacerlo pasa por convertir el formato de distribución oficial de SNOMED (tres archivos de texto plano que contienen los datos de los términos, atributos y relaciones, respectivamente) a formato OWL [22] con un script en lenguaje Perl proporcionado por la propia IHSTDO. El archivo generado es demasiado grande para que Protegé sea capaz de cargarlo en máquinas de 32bits. Sobre máquinas de 64bits con suficiente memoria RAM, el proceso es posible, pero el acceso posterior a la terminología es lento [23], ya que Protegé no está diseñado para la explotación de volúmenes de datos tan grandes.

Bioportal (ver fig. 3) es una herramienta de NCBO (National Center for Biomedical Ontology) [24] que permite tanto la carga de terminologías y ontologías por red a una base de datos como el acceso posterior a las mismas mediante una pagina web, o mediante funciones de servicio web. Esta característica hace que puedan programarse herramientas que utilicen estos servicios, de forma gratuita, aunque para poder añadir aportes propios a la base de datos se precisa crear una cuenta en la comunidad NCBO. Todas las terminologías y nuevas ontologías cargadas son validadas antes de permitir su almacenamiento y posterior acceso.



Figura 3. Captura de BioPortal

Aunque conveniente, esta característica podría limitar el uso de cualquier herramienta implementada que use Bioportal para acceder a los datos, ya que su funcionamiento dependerá del estado de un servidor ajeno. Por otro lado, las funciones de servicio web son genéricas para permitir el acceso a distintas terminologías. Crear nuestra herramienta basándonos en Bioportal habría supuesto a la larga tener que modificar las funciones que implementáramos para hacer búsquedas específicas en SNOMED a través de un servicio web, que a su vez accedería al servicio web de Bioportal, sobre el que no tendríamos control alguno.

Las herramientas empleadas finalmente en este proyecto para el desarrollo de la herramienta C.T.Hunter (la terminología SNOMED CT, UMLS, CTS y LexBIG) se describen en el siguiente capítulo.

## **4. C.T.Hunter**

Antes de presentar la herramienta resultado de este trabajo a la que hemos denominado C.T.Hunter (Cazador de Términos Clínicos) es necesario describir en profundidad los componentes que se han empleado en su creación. Estos componentes son la propia terminología SNOMED CT sobre la que se pretendía hacer búsquedas de términos, el Unified Medical Language System (UMLS), las APIs de Common Terminology Services (CTS) y la herramienta LexBIG.

### **4.1. SNOMED CT**

Systematic Nomenclature of Medicine – Clinical Terms es una terminología de términos clínicos que surge de la fusión de las terminologías SNOMED RT (Reference Terminology) desarrollada por el College of American Pathologist (CAP) y Clinical Terms Version 3 (CTV3) de la National Health Service (NHS) de Reino Unido. Actualmente los derechos de distribución y mantenimiento los tiene la IHTSDO (International Health Terminology Standards Development Organisation), una organización internacional sin ánimo de lucro afincada en Dinamarca. Esta organización fue fundada en el 2007 por 9 países (Australia, Canadá, Dinamarca, Lituania, Países Bajos, Nueva Zelanda, Suecia, Reino Unido y Estados Unidos) con el objetivo de desarrollar y mantener sistemas internacionales de terminologías clínicas, motivo por el cual adquirió los derechos de SNOMED. En la actualidad el número de países que la componen se ha incrementado a 15, entre ellos España.

SNOMED es el vocabulario clínico más extenso disponible en inglés, o en cualquier otro idioma. Contiene más de 311000 conceptos activos, así como sus relaciones y propiedades, y está traducido al danés, alemán y español de Argentina, entre otros [25].



La distribución de SNOMED en España corre a cargo del Ministerio de Sanidad. Sin embargo, la parte del núcleo traducida al español se encuentra en una versión en español de Argentina, lo que conlleva que se deba realizar una primera labor de validación de términos no comunes al español de España, para obtener de este modo una extensión que pueda ser adoptada como vocabulario de referencia. El Ministerio de Sanidad y Política Social se propone coordinar diferentes grupos de expertos para realizar estos trabajos de validación, así como la creación de extensiones traducidas para todas aquellas Comunidades Autónomas en las que existen lenguas cooficiales, enriqueciendo la edición nacional y permitiendo el intercambio de contenido multilingüe [26].

### **Estructura [27]:**

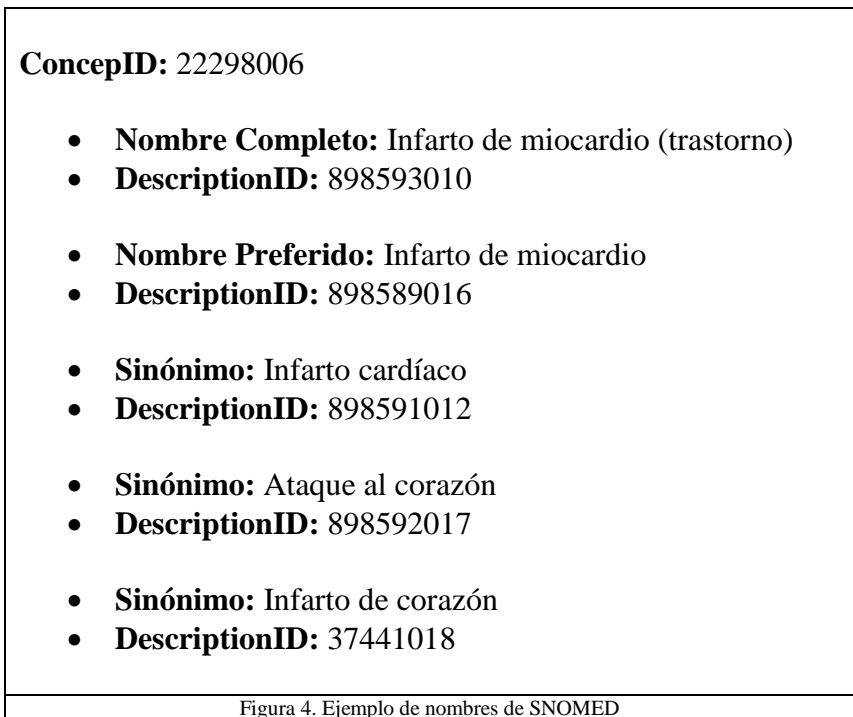
**Concepto:** Los conceptos en SNOMED son significados clínicos a los que se ha asignado un identificador numérico único (ConceptID). Este identificador no implica un significado jerárquico o implícito de ningún tipo.

Un concepto en SNOMED también tiene asociadas unas descripciones o términos. En este contexto, “término” es una frase que se utiliza para referenciar un concepto. Cada descripción tiene su propio identificador único (*DescriptionID*) y varias descripciones pueden relacionarse directamente con un solo concepto. Se definen varios tipos de descripciones:

- Nombre Completo (Fully Specified Name) se corresponde con el término médico menos ambiguo para representar el concepto asociado.
- Nombre Preferido se corresponde con los términos más usados para referirse a un concepto.

- Sinónimo a cualquier otra designación que el concepto pueda tener y que no entre en ninguno de los dos tipos anteriores.

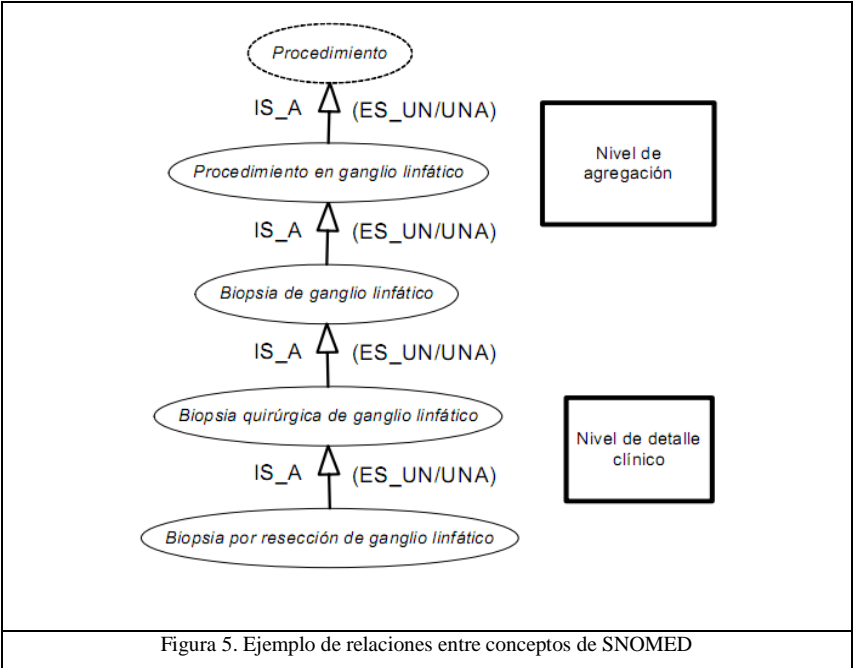
Un concepto puede tener varios Nombres Preferidos y Sinónimos, e incluso pueden existir otros conceptos que tengan un mismo Sinónimo, pero cada concepto puede tener un solo Nombre Completo y sólo uno. Un ejemplo de esta estructura puede verse en la figura 4.



Los conceptos en SNOMED también están relacionados unos con otros. Estas relaciones sirven para dotar a los conceptos de un significado que un programa informático puede ser capaz de utilizar para realizar razonamiento lógico. La relación *IS-A* es la que

estructura las jerarquías principales de SNOMED-CT. Los términos más generales en cada una de las jerarquías les dan nombre y cada nivel por debajo en la jerarquía es una especialización, de granularidad más fina: Un concepto es más o menos específico dependiendo de la cantidad de niveles de términos que derivan de él o el número de niveles de términos que en la jerarquía *IS-A* quedan por encima, de los que deriva.

La figura 5, extraída de la guía de usuario de SNOMED, muestra una de estas relaciones jerárquicas entre conceptos. El concepto “Biopsia por Resección del Ganglio Linfático” es muy concreto (granularidad más fina), mientras que “Procedimiento” puede referirse a muchos tipos de procedimiento (menos concreto, granularidad más gruesa).



Existen cuatro tipos diferentes de relaciones entre conceptos: Definitorias, Calificadoras, Históricas y Adicionales. Cada concepto de SNOMED, excepto el concepto raíz “Concepto de SNOMED CT”, del que todos los demás son descendientes, tiene al menos una relación *IS\_A* (ES-UN/A) con un concepto padre o *supertipo* (ver fig. 5). Estas relaciones *IS\_A* y las relaciones de atributos se consideran “características definitorias” de los conceptos en SNOMED, ya que se utilizan para representar un concepto lógicamente mediante sus relaciones con otros conceptos. Un ejemplo de un concepto de SNOMED y varios de sus atributos se muestra en la figura 6.

**ConceptID:** 263245004

**Nombre Completo:** Fractura del Hueso del Tarso (trastorno)

**Relaciones:**

- *ES UN/UNA (IS\_A):*
  - Fractura del Pie (trastorno)
- *SITIO DEL HALLAZGO (FINDING SITE):*
  - Estructura Ósea del Tarso (estructura corporal)
- *MORFOLOGIA ASOCIADA (ASSOCIATED MORPHOLOGY):*
  - Fractura (anomalía morfológica)

Figura 6. Ejemplo de Relaciones en SNOMED

Las relaciones *IS\_A* también se denominan “relaciones padre-hijo”, y son la base de las jerarquías de SNOMED. Un concepto puede tener varias relaciones *IS\_A* con otros conceptos, es decir, puede tener varios padres en jerarquías más altas puesto que es una relación transitiva. Por ejemplo, “Celulitis del Pie” es al mismo tiempo una “Celulitis” y un “Trastorno del Pie”, y por tanto tendrá dos relaciones *IS\_A*, una por cada uno de ellos.

Los atributos relacionan dos conceptos entre sí y establecen el tipo de relación entre ellos. Estas relaciones y las relaciones *IS\_A* permiten la representación lógica del significado de los conceptos. Los atributos se representan mediante dos campos para definir el tipo de atributo y el concepto asociado al mismo.

El tipo del atributo restringe a determinados tipos el concepto asociado, lo que se denomina *rango*. Por ejemplo, el rango para los atributos de tipo “SITIO DE HALLAZGO” abarca: “Estructura Anatómica” o “Estructura Corporal Adquirida”, y sólo pueden emplearse conceptos que pertenezcan a esas sub-jerarquías como concepto asociado a este tipo de atributo. De este modo, un valor lógico para un “Sitio de Hallazgo” puede ser un “Brazo” o una “Cicatriz Quirúrgica”, pero no tendría sentido que sea un “Escalpelo” (perteneciente a la jerarquía “Objeto Físico”).

Ciertos tipos de atributo sólo tienen sentido al aplicarse sobre ciertas jerarquías de conceptos, lo que se llama *dominio* del atributo. Los conceptos dentro de la jerarquía “Hallazgo Clínico”, por ejemplo, pueden tener atributos de tipo “MORFOLOGÍA ASOCIADA”, mientras que un atributo así en un concepto de “Procedimiento” no tendría sentido alguno.

## **Pre-coordinación y Post-coordinación:**

Una de las características de SNOMED es que muchos de sus conceptos son resultado del enlace de otros términos previamente definidos, lo que se conoce como pre-coordinación. Del mismo modo, se pueden establecer términos nuevos que no pertenezcan a la terminología simplemente enlazando términos que sí lo estén, lo que se conoce como post-coordinación.

SNOMED contiene muchos términos que son resultado de la coordinación de otros términos más simples. Si bien hay razones pragmáticas en cuanto a la manera en que SNOMED ha sido desarrollado para representar el lenguaje de uso clínico, y cómo ha ido

evolucionado con la incorporación de nuevos términos, lo cierto es que actualmente conviven términos post-coordinados predefinidos con diferentes niveles de complejidad semántica junto con algunos, no todos, de los componentes atómicos o conceptos esenciales que los constituyen. Por ejemplo: en el término predefinido “cáncer de colon no relacionado con poliposis hereditaria” (315058005), los términos atómicos serían 6:

- Cáncer: Término principal. Enfermedad, trastorno, hallazgo (dependiendo del contexto de la información)
- Colon: Órgano o posición anatómica (atributo que califica al término principal)
- No relacionado con: Relación ontológica. SNOMED no contiene un término que subsuma a todos los tipos de relaciones que existen.
- Poliposis: No existe el término general "poliposis" que subsuma a todas las poliposis existentes en SNOMED en diferentes posiciones anatómicas. Existen las poliposis gástricas, duodenales, etc. (específicas) pero no la poliposis genérica, como concepto.
- Intestinal: Órgano o posición anatómica (atributo que califica al término "poliposis") Para que SNOMED cubriera todo el espectro de posiciones anatómicas y morfologías debería contemplar el uso de una ontología completa sobre la anatomía, como hace por ejemplo la ontología FMA [28] (Foundational Model of Anatomy)
- Hereditaria: agente causante del síndrome o de la patología, en este caso genético o familiar (hereditario).

Aunque siempre fuera posible describir perfectamente un concepto clínico complejo con los términos post-coordinados predefinidos de SNOMED, la no existencia de sus términos atómicos impide el uso semántico avanzado de la terminología: por ejemplo la granularidad (generalidad o concreción de los términos) no cubre todos los grados en cada jerarquía y carece de las relaciones inherentes a sus términos atómicos.

## **Distribución:**

La distribución oficial de SNOMED ofrecida por la IHTSDO para sus miembros se compone de tres archivos de texto con los datos separados de los conceptos, las relaciones y los atributos. Aparte de esto, se incluyen archivos útiles como documentación, archivos de sub-set que clasifican subconjuntos de conceptos de SNOMED según algunos criterios (por ejemplo su uso en inglés de Reino Unido en contraposición con los términos empleados en América, o términos empleados en la rama de patología) y crossmaps, archivos que permiten establecer mapeos con otras terminologías (aunque de momento sólo existen mapeos con ICD-9-CM, LOINC y las versiones anteriores de SNOMED RT). También incluye utilidades, como un script para el cambio de formato a OWL.

No contiene herramientas de búsqueda, sólo contenido sobre el que aplicar herramientas informáticas construidas *ad-hoc*. Con sólo dos enlaces sobre otras terminologías vigentes, se hace evidente la necesidad de encontrar métodos de enlace alternativos que permitan acceder a otras terminologías especializadas del dominio biomédico para aumentar la potencia del razonamiento lógico-semántico.

## 4.2. UMLS

UMLS (Unified Medical Language System) es un proyecto creado por la Biblioteca Nacional de Medicina (NLM por sus siglas en inglés) de los EEUU para unificar las diferentes terminologías médicas existentes. Sus tres componentes principales son:

- El *MetaThesaurus*
- Una taxonomía o red semántica de tamaño reducido (poco más de un centenar de clases)
- Un conjunto de herramientas léxicas sólo disponibles en inglés

Podemos entender el *MetaThesaurus* como un gran compendio de diccionarios donde las palabras son términos clínicos y los diferentes idiomas/diccionarios son las diferentes terminologías que las contienen y que UMLS abarca (SNOMED CT, ICD-9-CM, MeSh, entre otras). La propia NLM se encarga de integrar las terminologías más relevantes en el *MetaThesaurus* (unas 150 en la última versión del 2009), así como de su actualización y mantenimiento.

Cada terminología es independiente y establece unas relaciones entre sus términos y una codificación propia de los mismos. Pero además, dentro del *MetaThesaurus* se establecen relaciones de equivalencia entre términos de diferentes terminologías que aluden al mismo elemento (concepto o universal), como si se tratara de una traducción entre idiomas. Para hacer esto posible, cada terminología es sometida a un proceso de *mapeo* de sus términos con conceptos de UMLS ya existentes, y a cada uno se le asigna un identificador CUI (Common Unique Identifier) del concepto que representa. Si el concepto no existe todavía en UMLS, se crea uno nuevo, al que se



asigna un nuevo CUI, tras lo cual se le asocia el término importado normalmente. De esta forma no se pierden los términos originales y se permite la coexistencia de distintas terminologías aunque subconjuntos de sus términos se solapen unos con otros.

El objetivo del NLM al crear UMLS era facilitar el acceso homogéneo a terminologías y, además, facilitar la creación de herramientas semánticas para aprovechar estos recursos. Por ello, además del identificador de concepto, a los términos importados se les asigna otro atributo importante, el Tipo Semántico (o clase semántica), que los enlaza con la Red Semántica de UMLS. Esta asociación permite diferenciar entre significados de un mismo término según el contexto en el que aparece. En las figuras 7 y 8 se pueden ver los diferentes Tipos Semánticos divididos entre “Entidades” y “Eventos”.

<i><b>Eventos</b></i>	
<ul style="list-style-type: none"> <li>Activity</li> <li>Behavior               <ul style="list-style-type: none"> <li>Social Behavior</li> <li>Individual Behavior</li> </ul> </li> <li>Daily or Recreational Activity</li> <li>Occupational Activity</li> <li>Health Care Activity               <ul style="list-style-type: none"> <li>Laboratory Procedure</li> <li>Diagnostic Procedure</li> <li>Therapeutic or Preventive Procedure</li> </ul> </li> <li>Research Activity               <ul style="list-style-type: none"> <li>Molecular Biology Research Technique</li> <li>Governmental or Regulatory Activity</li> <li>Educational Activity</li> </ul> </li> <li>Machine Activity</li> <li>Phenomenon or Process               <ul style="list-style-type: none"> <li>Human-caused Phenomenon or Process</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Environmental Effect of Humans</li> <li>Natural Phenomenon or Process</li> <li>Biologic Function               <ul style="list-style-type: none"> <li>Physiologic Function</li> <li>Organism Function</li> <li>Mental Process</li> <li>Organ or Tissue Function</li> <li>Cell Function</li> <li>Molecular Function                   <ul style="list-style-type: none"> <li>Genetic Function</li> </ul> </li> </ul> </li> <li>Pathologic Function               <ul style="list-style-type: none"> <li>Disease or Syndrome</li> <li>Mental or Behavioral Dysfunction</li> <li>Neoplastic Process</li> <li>Cell or Molecular Dysfunction</li> <li>Experimental Model or Disease</li> </ul> </li> <li>Injury or Poisoning</li> </ul>

**Figura 7. Tipos Semánticos de UMLS (Eventos)**

## Entidades

Physical Object	Organic Chemical
Organism	Nucleic Acid, Nucleoside, or Nucleotide
Plant	Organophosphorus Compound
Alga	Amino Acid, Peptide, or Protein
Fungus	Carbohydrate
Virus	Lipid
Rickettsia or Chlamydia	Steroid
Bacterium	Eicosanoid
Archaeon	Inorganic Chemical
Animal	Element, Ion, or Isotope
Invertebrate	Body Substance
Vertebrate	Food
Amphibian	Conceptual Entity
Bird	Idea or Concept
Fish	Temporal Concept
Reptile	Qualitative Concept
Mammal	Quantitative Concept
Human	Functional Concept
Anatomical Structure	Body System
Embryonic Structure	Spatial Concept
Anatomical Abnormality	Body Space or Junction
Congenital Abnormality	Body Location or Region
Acquired Abnormality	Molecular Sequence
Fully Formed Anatomical Structure	Nucleotide Sequence
Body Part, Organ, or Organ Component	Amino Acid Sequence
Tissue	Carbohydrate Sequence
Cell	Geographic Area
Cell Component	Finding
Gene or Genome	Laboratory or Test Result
Manufactured Object	Sign or Symptom
Medical Device	Organism Attribute
Research Device	Clinical Attribute
Clinical Drug	Intellectual Product
Substance	Classification
Chemical	Regulation or Law
Chemical Viewed Functionally	Language
Pharmacologic Substance	Occupation or Discipline
Antibiotic	Biomedical Occupation or Discipline
Biomedical or Dental Material	Organization
Biologically Active Substance	Health Care Related Organization
Neuroreactive Substance or Biogenic Amine	Professional Society
Hormone	Self-help or Relief Organization
Enzyme	Group Attribute
Vitamin	Group
Immunologic Factor	Professional or Occupational Group
Receptor	Population Group
Indicator, Reagent, or Diagnostic Acid	Family Group
Hazardous or Poisonous Substance	Age Group
Chemical Viewed Structurally	Patient or Disabled Group

Figura 8. Tipos Semánticos de UMLS (Entidades)

La herramienta de instalación del *Metathesaurus* de UMLS, *MetamorphoSys*, permite elegir las terminologías que se van a instalar (en nuestro caso, SNOMED-CT Edición en Español) generando unos archivos en un formato RRF (Rich Release Format), además de facilitar programas *script* para la inserción de los datos en una base de datos relacional, por ejemplo MySQL [29].

Consideramos que UMLS es la distribución más conveniente de terminologías para nuestro proyecto, con unas características muy útiles para la búsqueda semántica y para la traducción entre terminologías. Sin embargo, aunque incluye un buscador propio para realizar consultas sobre las terminologías instaladas, UMLS no ofrece ni *interfaces* de programación (API) ni servicios para la programación de buscadores, ya sean accediendo a una distribución en local o en remoto mediante servicio web. Podemos emplear UMLS para obtener nuestra base de datos de terminología SNOMED-CT pero para acceder a ella necesitaremos otras herramientas, que se presentan a continuación.

### 4.3. CTS

CTS, Common Terminology Services, es una iniciativa desarrollada primeramente por HL7 que ya forma parte de la Organización Internacional de Estándares (ISO). CTS define una serie de *Interfaces* de Aplicación (API) y un modelo de datos que permiten integrar y acceder de manera uniforme a diferentes terminologías.

Las APIs de CTS describen la funcionalidad básica necesaria para el acceso al contenido terminológico desde sistemas de información. Se presenta como una API con el objetivo de dar libertad a los desarrolladores sobre cómo almacenar la información terminológica. De este modo no se fuerza a migrar datos o reescribir código si se pretende, por ejemplo, integrar CTS en un sistema ya implantado. CTS ofrece un modelo conceptual y de datos para la gestión y almacenamiento de terminologías.

CTS solo identifica las características funcionales comunes que una terminología debe proveer. Por ejemplo: dado un código de concepto, permite determinar si es válido dentro de la terminología. Los desarrolladores son libres de implementar las llamadas a las API del modo que les resulte más conveniente.

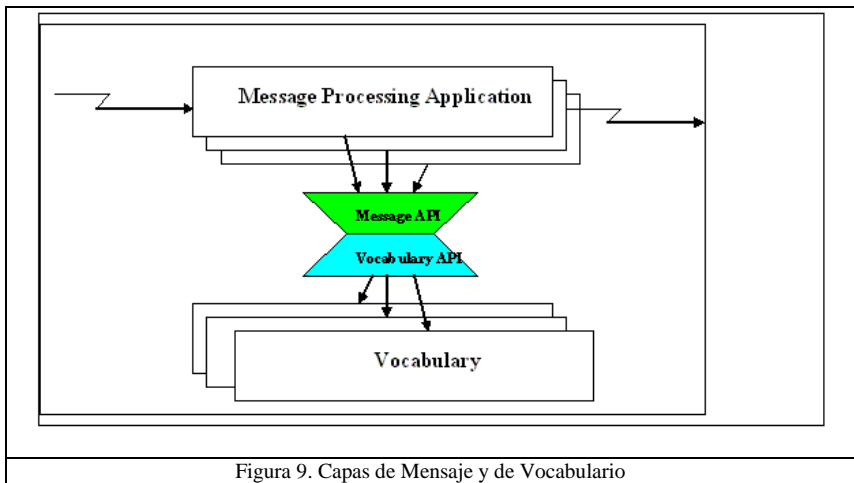
En este proyecto trabajamos con CTS 1.2, la última versión disponible (2004). Sin embargo, recientemente HL7 presentó un nuevo modelo actualizado para la nueva versión CTS 2, y está trabajando para presentar la versión funcional en breve.

La versión 1.2 de CTS se divide en dos capas, como se muestra en la figura 9.

La capa de Mensajes (**Message API**) es la capa superior. Se comunica con un software de mensajería en términos de dominios de vocabulario, contextos, conjuntos de valores, atributos codificados y otras características específicas del modelo de mensajes de HL7.

Esta capa no es interesante para este proyecto.

Por debajo, la capa de Vocabulario (**Vocabulary API**) se comunica con el software de almacenamiento de terminología utilizando un modelo conceptual propio que habilita el manejo de sistemas de código, códigos de concepto, definiciones, relaciones y otras entidades específicas de la terminología.



La API de Mensajes es específica para HL7, y su propósito principal es permitir a una amplia variedad de aplicaciones de procesamiento de mensajes crear, validar y traducir datos clínicos de forma consistente. La capa de API de Mensajes permite abstraerse a las aplicaciones de procesamiento de mensajes de la complejidad de las funciones específicas del vocabulario. Este nivel sirve de puente entre estas aplicaciones y la capa de funciones específicas de la terminología.

La API de Vocabulario, por otra parte, se ha diseñado para ser genérica y fácilmente utilizable en diferentes entornos, no sólo desde la capa de Mensajes.

La capa de Vocabulario trabaja sobre el supuesto de que existe una base de datos que contiene al menos una terminología o sistema de codificación (*CodeSystem*). Un sistema de codificación existente en la base de datos de vocabulario puede *contener* teóricamente un número ilimitado de conceptos codificados (*CodedConcepts*). Los conceptos codificados representan una clase o tipo dentro de un dominio particular. Cada concepto codificado puede estar definido únicamente en un sistema de codificación. Una vez definido, el significado del concepto codificado, que no sólo depende de la definición textual sino de su posición en la jerarquía de conceptos en el sistema de codificación y de sus relaciones explícitas con otros conceptos, no puede cambiar: es invariable. Los conceptos existentes pueden ser retirados y pueden añadirse conceptos nuevos, pero una vez definido, el significado de un concepto debe permanecer estático. La clase *CodeSystem* tiene los siguientes atributos (fig. 10).

- *codeSystem\_id*: un identificador único para el sistema de codificación. Dentro de HL7, un *codeSystemId* debería ser un OID (ISO [30] - *Object Identifier*). HL7 mantiene un registro de OIDs de sistemas de codificación, y anima a los usuarios a registrar cualquier nueva terminología en uso para que le sea asignado su identificador de objeto OID.
- *codeSystem\_name*: el nombre asignado al sistema de codificación (terminología) en el contexto de HL7, que posee un sistema de codificación propio para mantener registrados los nombres identificativos de todos los sistemas de codificación existentes.
- *fullName*: el nombre oficial del sistema de codificación. Para los sistemas de codificación registrados en el dominio de vocabulario del sistema de codificación propio de HL7, el atributo *fullName* es la designación o nombre del concepto (*ConceptDesignation*) en inglés para el concepto codificado correspondiente al “nombre del sistema de codificación”.

- *codeSystemDescription*: una descripción del propósito y contenido del sistema de codificación. Para sistemas registrados en el sistema de codificación propio de HL7, la descripción del concepto codificado en inglés del “nombre del sistema de codificación”.
- *copyright*: una nota de copyright opcional que, de estar presente, debería mostrarse siempre que se acceda al sistema de codificación.
- *supportedLanguages*: una lista de las lenguas que son completa o parcialmente soportadas dentro del sistema de códigos. Un lenguaje soportado se reconoce por el sistema de codificación y al menos algunos de los términos o propiedades de los conceptos estarán disponibles en ese lenguaje. Todos los sistemas de códigos deben soportar al menos un lenguaje. Mientras el servicio debe listar todas las etiquetas de la lengua principal, mostrar las lenguas alternativas es opcional (por ejemplo, si soporta “en-UK” (inglés de Reino Unido) debe listar “en” -inglés genérico- pero no es necesario que especifique la especialización para Reino Unido (“en-UK”). El primer lenguaje en la lista de lenguajes soportados se considera el lenguaje preferido para el sistema de codificación.
- *supportedRelations*: Representa los tipos de relaciones o roles soportados en el sistema de codificación. Las relaciones de subtipo se tratan como relaciones de primer orden en este modelo (código: *hasSubtype*, equivalente a la relación IS\_A de SNOMED). Un código de relación tiene un identificador de sistema de codificación y de concepto, así que es posible obtener los códigos de relación de cada una de las fuentes terminológicas almacenadas. Con objeto de mejorar la interoperabilidad (entre sistemas HL7), los códigos de relación deberían extraerse del sistema de codificación propio

(*ConceptRelationship*) de HL7 cuando sea posible.

- *supportedProperties*: Son los códigos de propiedades soportados por el sistema de codificación. También en este caso deberían usarse los códigos dentro del sistema de codificación (terminología propia) de HL7. Algunas posibles propiedades son *ActiveConcept* para señalar si el concepto es activo o se encuentra desfasado, o el atributo de CUI (Common Unique Identifier) que representa el identificador único de UMLS. Podemos decir que las “Propiedades” de CTS se corresponden con los “Atributos” de SNOMED.
- *supportedMimeType*: Una lista de tipos MIME usados en las designaciones/nombres, descripciones o propiedades en el sistema de codificación. Estos códigos deben sacarse del sistema de códigos oficialmente diseñado *Media Type* de HL7. El texto plano (text/plain) es el tipo MIME por defecto que debe ser soportado por todos los sistemas de código.
- *supportedRelationshipQualifiers*: Es la lista de calificadores de relación reconocidos por el sistema de códigos, por ejemplo, *hasSubtype* es el nombre de la relación equivalente a la relación IS\_A en SNOMED, pero si el calificador de relación es *inverse* (inverso), entonces equivaldrá a REVERSE\_IS\_A.

El uso de los códigos de HL7 para los diferentes atributos no es obligatorio (los programadores tienen libertad para implementar la base de datos y las funciones que acceden a ella), pero permite la interoperabilidad con otros sistemas de información basados en HL7.



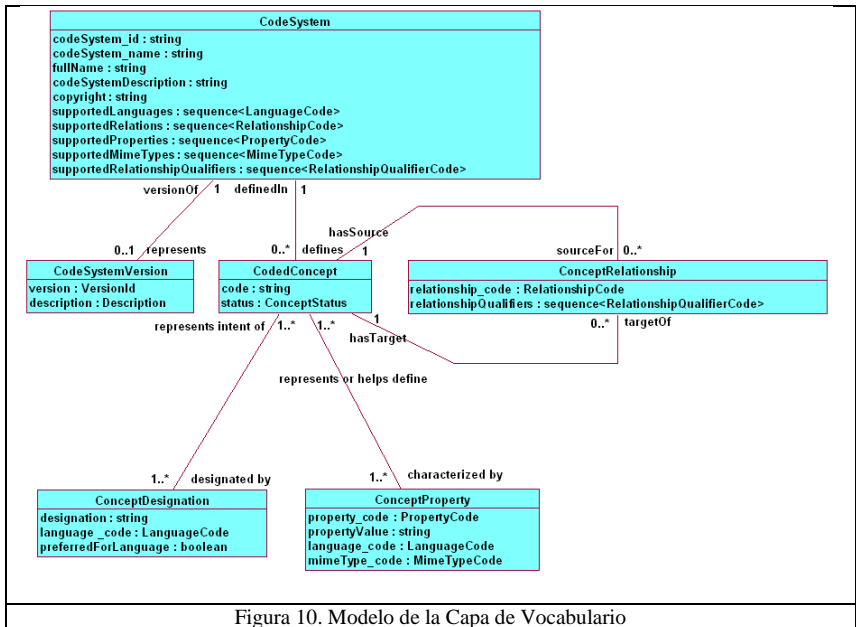


Figura 10. Modelo de la Capa de Vocabulario

Un Sistema de Codificación (CodeSystem) puede representar opcionalmente una versión concreta (CodeSystemVersion) de una terminología en cualquier momento dado.

Un concepto codificado (CodedConcept) es único dentro del CodeSystem que lo define, y debe tener al menos un nombre que lo designe (ConceptDesignation). Una de estas designaciones debe representar el nombre de uno o más conceptos codificados, siendo equivalentes a las descripciones (Nombre Preferido, Nombre Completo y Sinónimos) de SNOMED.

Los conceptos pueden, opcionalmente, ser caracterizados por propiedades (ConceptProperties). Una propiedad (ConceptProperty) debe representar o definir al menos un concepto. De este modo, las propiedades de CTS se corresponden con los atributos de SNOMED.

Un concepto puede ser fuente y/u origen de relaciones con otros conceptos (ConceptRelationships), de manera equivalente a como ocurre en SNOMED.

### **CodedConcept:**

Un concepto codificado tiene los siguientes atributos que lo caracterizan:

- *code*: un identificador único para el concepto o clase dentro del contexto del sistema de codificación/terminología (CodeSystem). Una vez asignado, el significado del código de concepto no puede cambiar nunca.
- *status*: representa el estado actual del concepto dentro del sistema de codificación. Los valores que puede tomar el atributo de status pueden extraerse de la terminología propia de HL7 llamada “ConceptStatus”. Entre los valores posibles están “propuesto”, “activo”, “eliminado” o “retirado”. La especificación actual, sin embargo, sólo reconoce por el momento la distinción entre “activo” y “no activo”.

### **ConceptDesignation:**

Una designación o nombre de concepto (ConceptDesignation) es un nombre u otro símbolo textual que representa los conceptos codificados. Estos nombres dependen del lenguaje. Los ConceptDesignation tienen los siguientes atributos:

- *designation*: Una cadena de texto que proporciona una representación externa de un concepto codificado.
- *languageCode*: un código de lenguaje que sigue las normas descritas en IETF RFC 3066 – Etiquetas [31] para la

Identificación de Lenguajes. Este código de lenguaje consiste en múltiples sub-etiquetas separadas por guiones ('-'). La primera sub-etiqueta identifica el código de lenguaje principal, por ejemplo ES (español) o EN (inglés). La segunda sub-etiqueta es opcional y, si está presente, será el código que determine el dialecto concreto de un país. Sub-etiquetas adicionales permiten refinar aún más el lenguaje. Por ejemplo, una etiqueta permitida puede ser “EN” (inglés, genérico), o también “EN-uk” (inglés de Reino Unido – United Kingdom).

- *preferredForLanguage*: si este atributo está activado (*TRUE*) significa que esta es la designación que debería usarse preferentemente para representar el concepto asociado en el lenguaje especificado cuando no hay otra información contextual presente. Sólo puede ser declarado un nombre como preferido para un concepto en cada lenguaje.

En correspondencia con SNOMED, los nombres designados como *preferredForLanguage* corresponden a los Nombres Preferidos. La falta de otros atributos para las designaciones hace que la única forma de diferenciar el Nombre Completo del resto de Sinónimos para un concepto dado sea por la etiqueta entre paréntesis al final del mismo, que indica su tipo semántico.

### **ConceptProperty:**

Una propiedad de concepto (ConceptProperty) es un atributo, faceta o cualquier otra característica que pueda representar o ayudar a definir el significado de los conceptos.

Los propiedades tienen los siguientes atributos en el modelo de CTS:

- *propertyCode*: una combinación del id del sistema de codificación/terminología y el concepto que identifica el tipo de propiedad. La terminología propia de HL7 ofrece códigos de propiedad propios.

- *propertyValue*: el valor textual de la propiedad asociada.
- *language\_code*: un código de lenguaje, que puede extraerse del sistema de códigos oficial diseñado por HL7 para este propósito. No todas las propiedades tienen códigos de lenguaje y, en caso de omisión, se da por supuesto que ningún lenguaje es aplicable para esa propiedad.
- *mimeType\_code*: un código que puede obtenerse del sistema de códigos Media Type de HL7 oficial diseñado para este propósito y que representa el tipo MIME asociado a la propiedad. No todas las propiedades tienen un tipo MIME asociado y el texto plano es el tipo por defecto.

Las propiedades de CTS se corresponden con los atributos de SNOMED. De esta forma, los códigos de propiedad (*propertyCode*) se corresponden con los códigos que tiene el propio SNOMED para sus atributos, además de los atributos añadidos por la distribución de UMLS que son el CUI y el Tipo Semántico, cada uno con sus propios códigos ya asignados.

### **ConceptRelationship:**

La clase de relaciones de conceptos (ConceptRelationship) representa relaciones binarias sobre el conjunto de conceptos definido en un sistema de codificación.

Cada relación debe tener exactamente un concepto codificado como fuente y otro como objetivo de la relación. El atributo *relationship\_code* identifica de que tipo de relación se trata.

El sistema de códigos de HL7 (ConceptCodeRelationship) contiene las definiciones de relaciones

usadas por la versión 3 del Vocabulario de HL7, así como las características de transitividad, reflexividad y simetría de las relaciones y el término utilizado para representar la relación inversa. Estas relaciones son genéricas, y tan sólo hasSubtype y PAR se corresponden con relaciones de SNOMED (IS\_A y REVERSE\_IS\_A respectivamente). En la tabla I se muestran algunos ejemplos de estas relaciones.

<b>Tabla I. Ejemplos de relaciones de conceptos y sus propiedades</b>					
<b>Concept Relationship</b>	<b>Descripción</b>	<b>Trans.</b>	<b>Refl.</b>	<b>Simtr.</b>	<b>Inv.</b>
hasSubtype	El concepto fuente tiene el concepto objetivo como subtipo	Si	No	No	isSubtypeOf
hasPart	El concepto fuente está compuesto por varias partes y el concepto objetivo es una de ellas	Si	Si	No	IsPartOf
smallerThan	El concepto fuente es más pequeño que el concepto objetivo	Si	No	No	greaterThan

Las siguientes funciones (tablas II y III) pertenecen a la capa de Vocabulario, divididas entre funciones de tiempo de ejecución (Runtime) y funciones de navegación (Browsing).

**Tabla II. Funciones de tiempo de ejecución (Runtime) de la capa de Vocabulario**

<b>Funcion</b>	<b>Entrada</b>	<b>Salida</b>	<b>Descripción</b>
getService Name		service_name	Devuelve el nombre asignado al servicio por el proveedor
getService Version		Identificador de la versión	Devuelve la versión actual del software del servicio.
getService Description		Descripción del servicio	Devuelve la descripción de la función del servicio, autores, copyright, etc.
getCTS Version		Número de versión del CTS empleado	Devuelve la versión de CTS implementada en este servicio.
getSuported CodeSystems	Tiempo limite, tamaño límite.	Lista de terminologías (code systems) y versiones soportadas por el servicio	Devuelve el identificador, nombre y versión de todas las terminologías soportadas por el servicio.
lookupCode SystemInfo	Nombre o id de la terminología.	Descripción de la terminología, incluyendo nombre, id, descripción, versión, lenguajes, relaciones y propiedades soportadas, etc.	Devuelve información detallada sobre una terminología específica.
isConcept IdValid	Id de terminología, flag indicando si los conceptos inactivos se consideran válidos.	True/False	Determinar si el concepto es válido en una terminología específica.
Lookup Designation	Id de la terminología y código del lenguaje objetivo.	Texto definitorio	Devuelve el nombre preferente para un código de concepto en el lenguaje determinado.
areCodes Related	Id de la terminología, id de los conceptos fuente y objetivo, código de relación, calificador y flags indicando si usar buscar solo relaciones directas o relaciones por transitividad.	True/False	Determinar si la relación especificada existe entre los códigos fuente y objetivo.

**Tabla III. Funciones de navegación (Browsing) de la capa de Vocabulario**

<b>Funcion</b>	<b>Entrada</b>	<b>Salida</b>	<b>Descripción</b>
getService Name		service_name	Devuelve el nombre asignado al servicio por el proveedor
getService Version		Identificador de la versión	Devuelve la versión actual del software del servicio.
getService Description		Descripción del servicio	Devuelve la descripción de la función del servicio, autores, copyright, etc.
getCTS Version		Número de versión del CTS empleado	Devuelve la versión de CTS implementada en este servicio.
getSupported Match Algorithms		Lista de algoritmos	Devuelve los algoritmos soportados por el servicio
getSupported CodeSystems	Tiempo y tamaño límite	Lista de terminologías soportadas y sus descripciones	Devuelve la lista de terminologías accesibles por el servicio
Lookup Concept CodesBy Designation	Texto a buscar y algoritmo, id de la terminología, id del lenguaje, flag indicando si los conceptos inactivos son válidos o no	Lista de códigos de concepto encontrados (emparejados con el identificador de su terminología)	Devuelve una lista de códigos de concepto cuyas designaciones se correspondan con la cadena de texto buscada, en el lenguaje introducido (si dichos códigos existen)
Lookup ConceptCodes ByProperty	Texto a buscar y algoritmo, id de la terminología donde buscar, id de lenguaje, flag indicando si los conceptos inactivos son válidos o no, lista opcional de propiedades que buscar	Lista de códigos de concepto y sus respectivos ids de terminología	Devuelve una lista de conceptos cuyas propiedades cumplen el criterio d búsqueda establecido.

Lookup Complete Coded Concept	Id de la terminología y código de concepto	Todos los datos conocidos sobre el concepto requerido (nombres, propiedades, relaciones, etc.)	Devuelve todos los datos del concepto introducido.
Lookup Designations	Id de la terminología, código de concepto, texto a buscar, algoritmo, lenguaje objetivo	Lista de nombres	Devuelve todos los nombres asociados al código.
Lookup Properties	Id de la terminología, código de concepto, texto a buscar, algoritmo, lista de propiedades en las que buscar, lista de tipos mime, lenguaje objetivo	Lista de propiedades (código de propiedad, valor, lenguaje, tipo mime)	Devuelve las propiedades solicitadas dado un id de terminología y un código de concepto que cumplan los criterios establecidos.
Lookup CodeExpansion	Id de terminología, código de concepto, código de relación, indicador de la dirección de la relación, lenguaje objetivo, tiempo y	Lista de expansión jerárquica del código	Lista recursiva de los códigos de concepto relacionados con el concepto proporcionado, incluyendo los nombres preferidos para los códigos

CTS sólo define un API, de modo que no ofrece implementación alguna para las funciones. A partir de aquí, se plantearon dos alternativas:

- 1) Cargar la terminología SNOMED extraída de la distribución oficial o de UMLS (esta última tiene medios para cargar la terminología, mientras que con la distribución oficial habría que crear herramientas de carga desde cero) en una base de datos, y a partir de ahí implementar las funciones de CTS en



el ámbito de nuestro proyecto.

- 2) Emplear otra herramienta que ya tuviera implementadas las funciones de CTS y ofreciera la posibilidad de cargar la terminología de modo que fuera compatible con estas.

La segunda opción era la más directa para alcanzar nuestros objetivos y, puesto que encontramos una herramienta (LexBIG, analizada en el siguiente punto) que cumplía estas condiciones, fue la opción seleccionada.

## 4.4. LexGrid y LexBIG

LexGrid es un modelo de datos y una implementación de la Clínica Mayo para almacenar de forma estándar vocabularios controlados y ontologías. El modelo LexGrid define un formato y una representación para vocabularios, y se pretende que sea lo suficientemente flexible como para representar con exactitud una gran variedad de terminologías y otros recursos léxicos. El modelo, además, define varios mecanismos de almacenamiento (bases de datos relacionales, LDAP, etc.) y formato XML.

El modelo LexGrid permite caracterizar las partes fundamentales de las terminologías de manera que todas pueden descomponerse y conceptualizarse para ser traducidas al modelo de datos LexGrid. Este modelo común es la parte fundamental del proyecto LexGrid. Una vez la información de vocabularios dispares se encuentre representada bajo el mismo modelo estándar, es posible construir repositorios compartidos para contenido terminológico e interfaces comunes y herramientas para acceder y gestionar el contenido.

A partir de este modelo de datos, la comunidad caBIG (Cancer Biomedical Informatics Grid) [32] refinó la implementación de la Clínica Mayo para dar lugar a la herramienta LexBIG.

LexBIG es un conjunto de servicios diseñados para almacenar y acceder metadatos de terminologías. Se trata de un proyecto específico que aplica la tecnología diseñada en el modelo LexGrid. El objetivo de LexBIG es construir servidores de vocabulario que puedan ser accedidos mediante una API bien estructurada, capaz de acceder y distribuir recursos. El servidor se crea a partir de tecnologías estándares bien conocidas. Los objetivos principales de este proyecto son:

- Proporcionar una implementación robusta, ampliable y open-source de servicios de vocabulario. La especificación de la API se basa, pero no se limita, a las APIs implementadas por caBIG (LexBIG API).
- Proporcionar una implementación simple para el almacenamiento y persistencia del vocabulario, permitiendo el uso de distintos mecanismos de manera transparente para los usuarios. Este objetivo incluye la posibilidad de elegir entre diferentes productos de almacenamiento de bases de datos: mySQL, PostgreSQL, HypersonicSQL, etc.
- Proporcionar herramientas de carga y distribución estándares para el contenido terminológico. Esto incluye, pero no se limita, al formato RFF de UMLS, el lenguaje de ontologías web OWL, y Open Biomedical Ontologies (OBO).

La herramienta LexBig permite, a partir de la distribución de una terminología, generar una base de datos (mySql, PostgreSQL, Oracle, HyperSonicSQL, etc.) cargada con el contenido terminológico y enlazarla para el uso de diferentes APIs cuyas funciones tiene ya implementadas (el CTS de HL7 presentado anteriormente y la propia API de LexBIG), aparte de tener sus propias herramientas de búsqueda. Basado en el modelo de información de LexGRID para representar y compartir recursos terminológicos a gran escala, LexBIG es una serie de programas para cargar, indexar, publicar y editar contenidos para un servidor de terminologías.

A partir de este proyecto, caBIG inició también un proceso para combinar LexBIG con los servicios de vocabulario EVS (Enterprise Vocabulary Server) [33] proporcionados por la NCI (National Cancer Institute) (ver fig. 11). A día de hoy, la distribución estándar de la herramienta LexBIG/LexEVS integra ambos servicios.

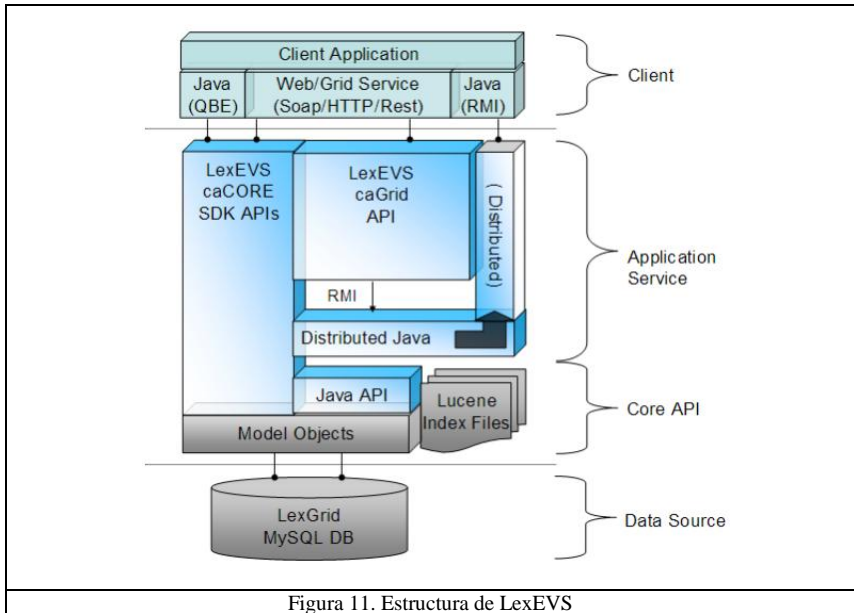


Figura 11. Estructura de LexEVS

LexEVS es una colección de interfaces programables que proporcionan a los usuarios la capacidad de acceder a terminologías ofrecidas por el proyecto NCI EVS. Las terminologías dentro de este proyecto se publican de modo open-source en el servidor de terminologías de LexEVS.

El objetivo de estos proyectos es el de unificar todos los vocabularios bajo un mismo modelo de información (LexGrid) capaz de representarlos, para facilitar el desarrollo de herramientas y APIs que lo soporten. Aunque el alcance de nuestro proyecto sólo pretende realizar búsquedas sobre una base de datos de SNOMED CT, LexBIG abre la posibilidad de ampliarla para soportar otras bases de datos que contengan otras terminologías. Por otro lado, al centrarse en funciones de búsqueda estándares para todas las terminologías que pretende soportar, carece de funciones para aprovechar las características específicas de cada terminología.

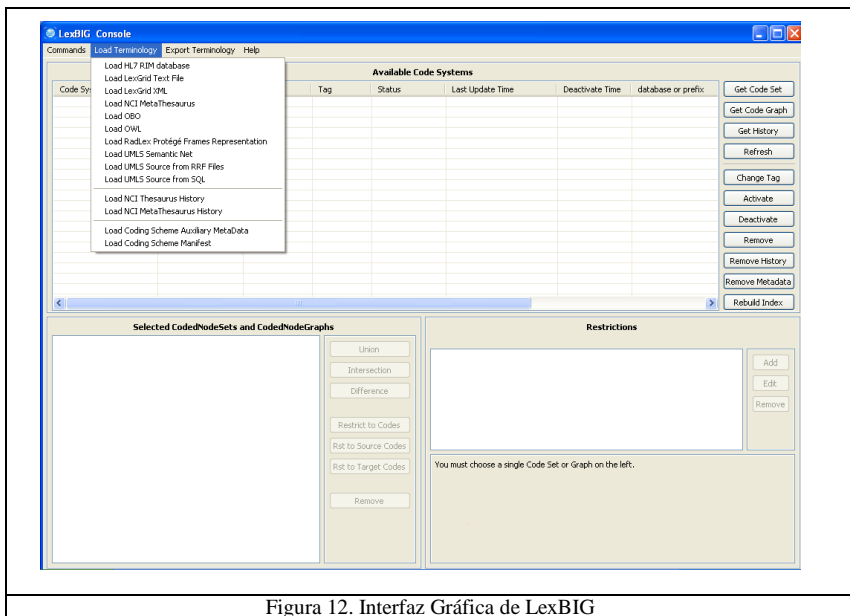


Figura 12. Interfaz Gráfica de LexBIG

La interfaz gráfica de LexBIG (fig. 12) permite tanto la carga de terminologías como la búsqueda de conceptos empleando sus propias APIs. Los pasos a seguir para realizar una búsqueda son complicados y poco intuitivos, pero las APIs de LexBIG, aunque no se han empleado en nuestro proyecto por falta de documentación al respecto, presentan propiedades que pueden resultar interesantes a la hora de ampliar nuestras funciones de búsqueda.

La característica a resaltar es la capacidad de realizar búsquedas con varias restricciones. Las funciones definidas por CTS sólo permiten buscar términos (textuales) que contengan ciertas palabras en un cierto campo (buscar “diabetes” en el nombre, o “alergia” en una propiedad hijo, por ejemplo). Las funciones propias de LexBIG permiten introducir varias restricciones (buscar la palabra “fractura de hueso” en el nombre, y que, por ejemplo, tenga como propiedad “Sitio del Hallazgo” = “Tarso”).

Esta característica permitiría mejorar las búsquedas sobre SNOMED aprovechando sus propiedades. Sin embargo, debido a la ausencia de manuales al respecto, la única forma de aprender a usar las funciones de esta API de LexBIG es estudiando los programas de ejemplo que se proporcionan en la distribución, y empleando prueba y error.

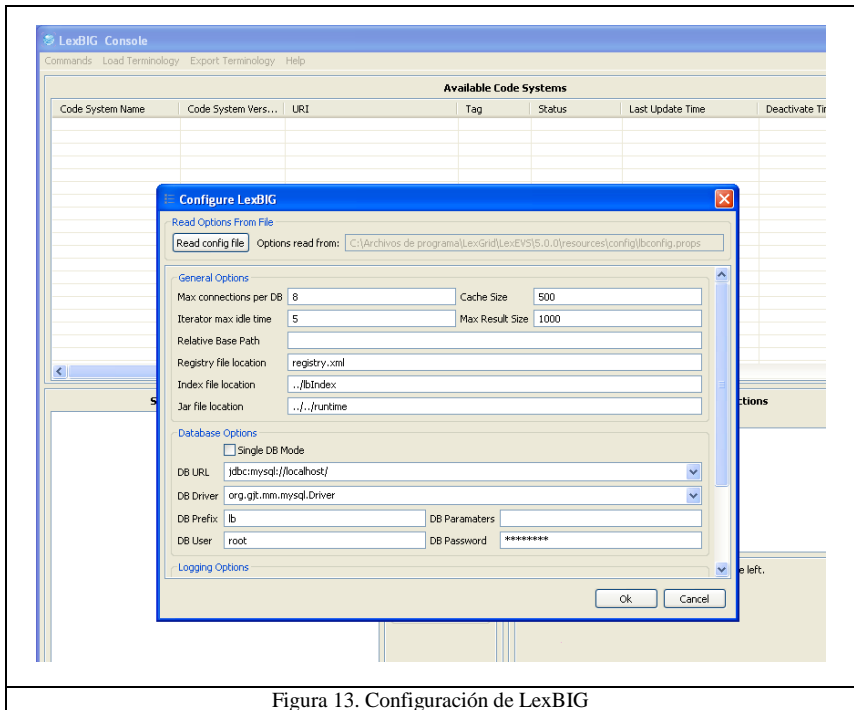


Figura 13. Configuración de LexBIG

Antes de iniciar la carga de terminologías se debe configurar LexBIG con los datos de la base de datos que se utilizará (mySQL en nuestro caso) y las opciones de conexión. Esto se puede hacer directamente desde la interfaz gráfica (fig. 13), o modificando el archivo *lbconfig.props* con un editor de texto. Una vez hecho esto, se puede comprobar si los parámetros introducidos no

provocan error empleando las funciones de la carpeta test dentro de la carpeta de instalación de LexBIG/LexEVS.

Para poder emplear las funciones implementadas de CTS con LexBIG es necesario descargar un paquete aparte proporcionado en la página de descargas del proyecto LexBIG, denominado “CTS deploy”, y descomprimirlo sobre el directorio de instalación de LexBIG. El paquete proporciona tanto las funciones como un programa propio de prueba.

Aunque la implementación de las funciones de CTS resulta útil, presenta algunos errores de implementación o incongruencias entre la especificación formal de HL7 y la implementación realizada. Por ejemplo, en la definición de HL7 acerca del comportamiento a implementar en las funciones de su API, se establece que todas las funciones que acepten como parámetro de entrada un entero indicando el máximo de resultados, deben considerar que si este parámetro es 0, se deben devolver todos los resultados. Sin embargo, la implementación hecha por los creadores de LexBIG ignora esta excepción, de forma que introduciendo 0 en ese dato, se realiza la búsqueda para no devolver ningún dato. Esto hace que la única forma en que podemos estar seguros de que vamos a recibir todos los resultados de una búsqueda sin que este tope máximo nos lo impida es introducir un valor arbitrariamente grande, y comprobar que el número de resultados recibidos no coincida con el parámetro enviado (si coincide el número de resultados obtenido con el valor del parámetro la probabilidad de que no hayan sido devueltos todos los resultados es muy alta).

### 4.5. C.T.Hunter

Uno de los objetivos de este proyecto es desarrollar un servidor de términos de SNOMED y una herramienta para acceder al mismo y realizar consultas, aprovechando en la medida de lo posible los recursos open-source que hubiera disponibles.

Aparte de las funciones de búsqueda básicas, se pretende también crear funciones de búsqueda adaptadas a las características propias de SNOMED (atributos, relaciones, jerarquías de conceptos, etc.).

La figura 14 muestra la organización final de los componentes que se han empleado en la implementación del servidor de términos y la herramienta C.T.Hunter.

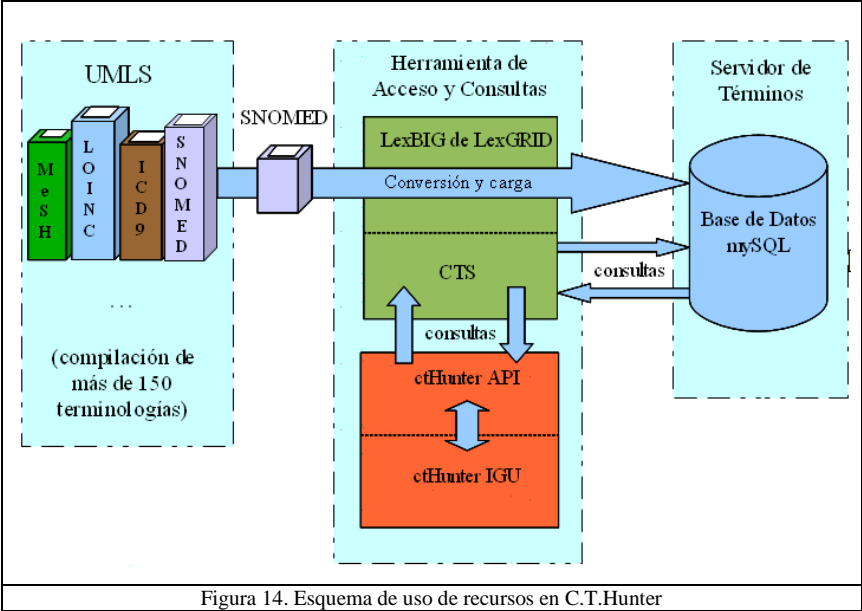


Figura 14. Esquema de uso de recursos en C.T.Hunter



### 4.5.1. Base de Datos

Inicialmente se extrajo la terminología SNOMED en español de una distribución de UMLS del 2007, para luego cargarla mediante LexBIG en una base de datos de MySQL. La carga fue larga, no sólo por el volumen de datos, sino por la inexistencia de tutorial alguno por parte de LexBIG, que obligó a realizarla a base de prueba y error, con el consiguiente retraso respecto a la planificación prevista.

Para cargar los datos terminológicos se pueden emplear dos métodos. Uno de ellos es seleccionar la opción de crear scripts de carga en MySQL al extraer la terminología de UMLS usando su herramienta de extracción *MetamorphoSys*, y seleccionar “cargar UMLS por SQL” en LexBIG. Sin embargo, este método es gravoso para el ordenador, en especial si la base de datos sobre la que trabaja LexBIG es la misma.

El otro método, que se empleó en la carga final, fue extraer la terminología normalmente con *MetamorphoSys* y cargarla mediante la opción “cargar UMLS desde RRF”. En la ventana correspondiente del programa, habrá que introducir la ruta donde se encuentran los archivos, y la abreviatura RSAB (Root Source Abreviation, Abreviatura de la Fuente Raíz) correspondiente a la terminología a cargar, que se puede consultar en la página de UMLS (en nuestro caso, “sctspa” para la distribución de SNOMED CT en español).

El primer intento se realizó mediante una máquina virtual, con la idea de utilizarla más adelante para simular las consultas por web. El primer paso fue extraer la terminología SNOMED en español de una distribución de UMLS mediante la herramienta *MetamorphoSys*, obteniéndola así en formato RRF (Rich Release Format, propio de UMLS). A partir de ésta, se inició la conversión a formato LexGRID y carga sobre MySQL mediante el

programa script de LexBIG. Sin embargo, las bajas prestaciones de la máquina virtual ralentizaron el proceso en demasía, tal que dos semanas después del inicio el tamaño de la base de datos final no alcanzaba un cuarto de lo esperado. Finalmente, sin forma de acelerar el proceso ni calcular cuánto podría tomarle al programa completar la tarea, se dio por imposible la carga mediante la máquina virtual y se pasó a realizar la importación sin virtualización.

Las siguientes cargas fallaron a mitad de carga por error de LexBIG al encontrarse con datos codificados con caracteres UTF-8, algo que de haber tenido un tutorial podría haberse evitado. De nuevo se procedió a extraer los datos de UMLS, eligiendo esta vez la opción “sin caracteres UTF-8”, tras lo cual finalmente se pudieron cargar los nuevos RRF mediante LexBIG sin error. El tiempo total empleado en la carga final fue de aproximadamente siete días utilizando la potencia de un PC de sobremesa.

Una vez cargada la base de datos, a partir de una API previamente definida con las funciones básicas (descritas en la tabla IV del apartado siguiente), se procedió a implementar dichas funciones apoyándose en las funciones del API de CTS ya implementadas para LexBIG.

#### 4.5.2. Consultas básicas

Uno de los requisitos planteados para las funciones básicas era que los datos devueltos pudieran codificarse fácilmente en XML, pensando en su futuro acceso mediante servicios web.

Las funciones básicas implementadas se encargan de realizar las llamadas correspondientes a las funciones de CTS y convertir la salida (ver tabla IV).

<b><i>Tabla IV. Funciones Básicas de C.T.Hunter</i></b>			
<b><i>Funcion</i></b>	<b><i>Entrada</i></b>	<b><i>Salida</i></b>	<b><i>Descripción</i></b>
descSearch	Palabras a buscar, algoritmo de búsqueda, máximo de resultados a devolver	Una lista de códigos de concepto	Búsqueda de conceptos dirigida solo a sus descripciones. Los códigos devueltos cumplen las condiciones de búsqueda introducidas.
propertySearch	Palabras a buscar, algoritmo, máximo de resultados a devolver, lista de propiedades en las que buscar	Una lista de códigos de concepto	Búsqueda de conceptos dirigida a las propiedades. Se buscan los conceptos que tengan al menos una de las propiedades indicadas que coincida con las palabras a buscar.
getData	Código de un concepto, dos flags para indicar si se quiere también buscar los padres e hijos del mismo	Una lista de strings con los datos etiquetados del concepto.	Devuelve todos los datos del concepto indicado.
getAlgorithms		Lista de strings con los algoritmos implementados en el servicio	Devuelve los algoritmos que se pueden introducir para realizar las búsquedas

**Tabla IV. Funciones Básicas de C.T.Hunter**

getParents	Código de concepto, número de generaciones que queremos, máximo de resultados a devolver	Lista de códigos de concepto	Devuelve los antecesores del concepto indicado, según la cantidad de generaciones atrás que se le pida consultar
getChildren	Código de concepto, número de generaciones que queremos, máximo de resultados a devolver	Lista de códigos de concepto	Devuelve los descendientes del concepto indicado, según la cantidad de generaciones en adelante que se le pida consultar
isValidCode	Código de concepto	Si o No	Devuelve si un código es válido (pertenece a la base de datos de SNOMED)
getSinonims	Código de concepto	Lista de sinónimos	Devuelve todos los sinónimos conocidos del concepto
getFSN	Código de concepto	Cadena	Devuelve el Nombre Completo (Fully Specified Name) del concepto
getPrefName	Código de concepto	Cadena	Devuelve el nombre preferido del concepto
getProperties	Código de concepto	Lista de propiedades y sus valores	Devuelve todas las propiedades del concepto y sus valores
getPropertiesOfKind	Código de concepto, lista de propiedades a devolver	Lista de propiedades y sus valores	Devuelve solo las propiedades indicadas del concepto y sus valores
getCUI	Código de concepto	Cadena	
getST	Código de concepto	Lista de cadenas	Devuelve la lista de Tipos Semánticos del código

Aparte de estas funciones se implementó una función interna extra llamada `initServices`, cuyo trabajo consiste en inicializar los servicios de *Runtime* y *Browser* de CTS para que las demás funciones puedan usarlas. Todas las demás funciones comprueban al ser llamadas si los servicios han sido inicializados y en caso negativo ejecutan la llamada a `initServices` antes de continuar.

Además `initServices` también se encarga de almacenar en variables estáticas la lista de algoritmos disponibles y el identificador (id) del sistema de codificación (CodeSystem) SNOMED para reducir el tiempo de búsqueda al realizar consultas que necesiten estos datos (fig 14).

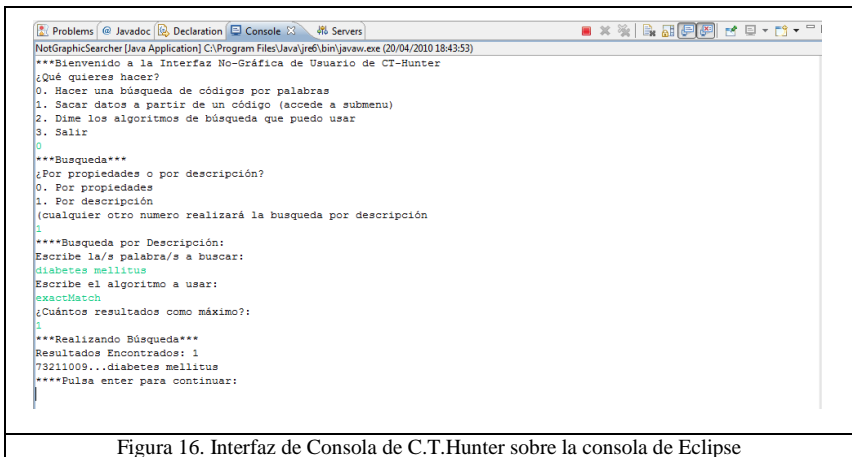
```
private void initServices() {  
    ServiceHolder.configureForSingleConfig();  
    servo = ServiceHolder.instance().getLexBIGService();  
    runner.setLexBIGService(servo);  
    browser.setLexBIGService(servo);  
  
    aAlgoritmos = browser.getSupportedMatchAlgorithms();  
    snomedSP_id=browser.getSupportedCodeSystems()[0].getCodeSystemId();  
}
```

Figura 15. Implementación de `initServices`

Nuevamente hubo un imprevisto cuando se procedió a probar las funciones básicas. Los datos cargados contenían todos los conceptos de SNOMED en español, pero no las relaciones propias de SNOMED (aunque sí contenían la mayoría, no todas, de las relaciones del tipo IS\_A). Esto limitó las funciones extendidas que se podrían implementar, dejando las búsquedas sobre propiedades de SNOMED como trabajo pendiente en futuras revisiones. No obstante, dicha implementación será sencilla una vez los datos completos estén cargados.

### 4.5.3. Interfaz Gráfica de Usuario

Previamente a la interfaz gráfica, se desarrolló además una interfaz de consola de prueba (ver fig. 16) con menús de texto sobre la consola de Eclipse.



```
Problems | Javadoc | Declaration | Console | Servers
NotGraphicSearcher [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (20/04/2010 18:43:53)
***Bienvenido a la Interfaz No-Gráfica de Usuario de CT-Hunter
¿Qué quieres hacer?
0. Hacer una búsqueda de códigos por palabras
1. Sacar datos a partir de un código (accede a submenu)
2. Dime los algoritmos de búsqueda que puedo usar
3. Salir
0
***Búsqueda***
¿Por propiedades o por descripción?
0. Por propiedades
1. Por descripción
(cualquier otro numero realizará la búsqueda por descripción)
1
***Búsqueda por Descripción:
Escribe la/s palabra/s a buscar:
diabetes mellitus
Escribe el algoritmo a usar:
exactMatch
¿Cuántos resultados como máximo?:
1
***Realizando Búsqueda***
Resultados Encontrados: 1
73211009...diabetes mellitus
***Pulsa enter para continuar:
```

Figura 16. Interfaz de Consola de C.T.Hunter sobre la consola de Eclipse

Una vez comprobadas las funcionalidades, se pasó a implementar la interfaz gráfica usando SWING [34], una librería gráfica para java.

La interfaz gráfica está dividida en dos partes. En la parte superior se encuentra la barra de menús y la barra de búsqueda. La parte inferior, inicialmente vacía, es la destinada a mostrar los resultados de las búsquedas realizadas, en forma de pestañas. El aspecto está diseñado para parecerse a un navegador web, con la posibilidad de navegar por las pestañas, cerrar aquellas búsquedas que ya no sean útiles, y acceder fácilmente a los datos de un concepto referenciado en una de las pestañas (fig. 17).

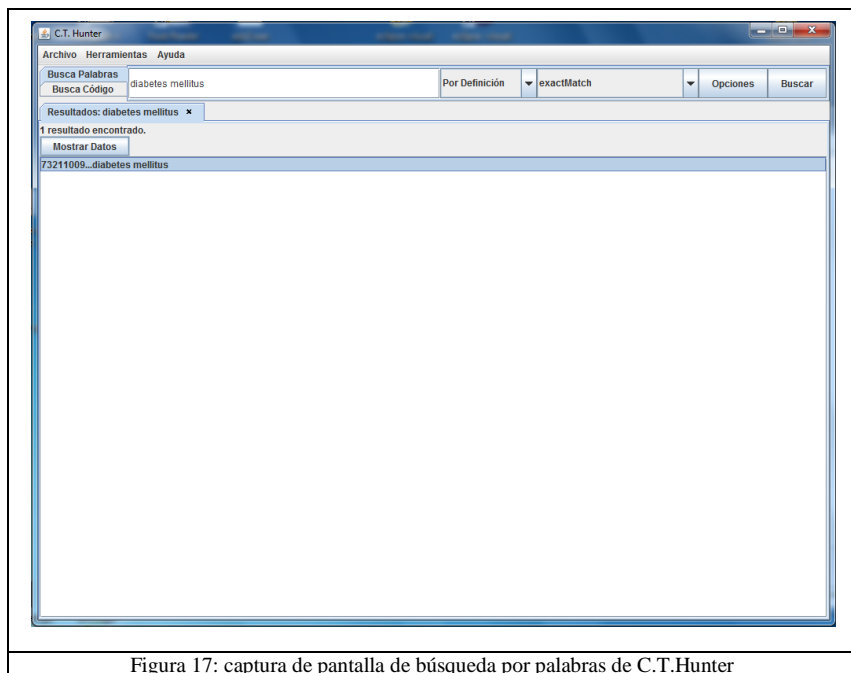


Figura 17: captura de pantalla de búsqueda por palabras de C.T.Hunter

La barra de búsqueda consta de dos pestañas que permiten elegir el modo de búsqueda a utilizar.

La búsqueda por palabras permite buscar conceptos cuyas definiciones o propiedades contengan una cierta cadena de palabras.

El primero de los campos de esta pestaña es editable y permite escribir las palabras a buscar. El siguiente campo es un selector que permite elegir si realizar la búsqueda sobre los nombres de los conceptos o sobre las propiedades. El siguiente selector permite elegir el algoritmo de búsqueda a utilizar. El botón de Opciones abre una ventana para seleccionar el tamaño máximo de los resultados de la búsqueda y las propiedades donde buscar (fig. 18). Finalmente, el

botón Buscar ejecuta la búsqueda.

Una vez la búsqueda se ha realizado, los resultados se muestran en la parte inferior de la interfaz como una nueva pestaña. En la figura 18 puede verse el único resultado de la búsqueda de “diabetes mellitus” con el algoritmo “exactMatch” (palabras exactas).

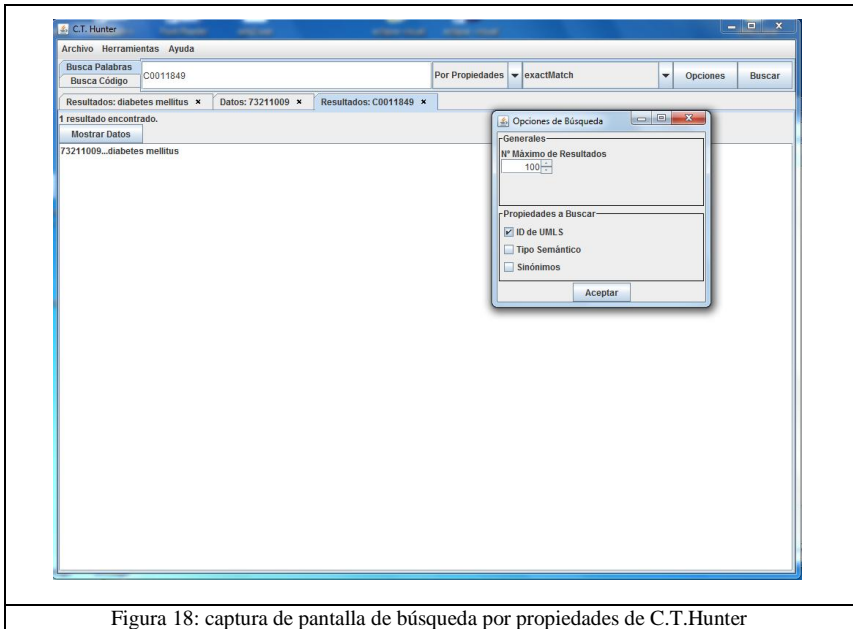


Figura 18: captura de pantalla de búsqueda por propiedades de C.T.Hunter

La pestaña de resultados de búsqueda permite seleccionar uno de los resultados y realizar una búsqueda de sus datos pulsando el botón Mostrar Datos.



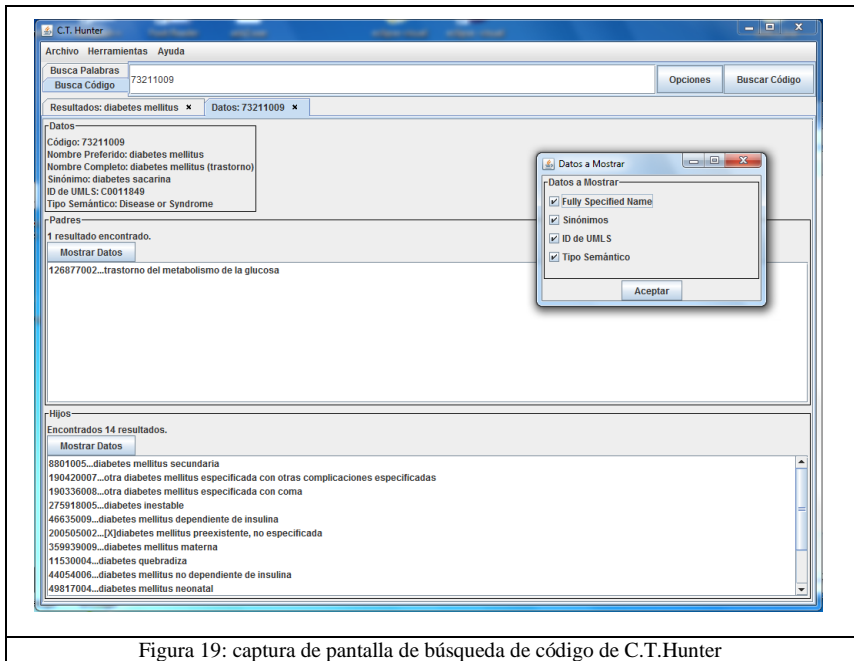


Figura 19: captura de pantalla de búsqueda de código de C.T.Hunter

Para buscar los datos de un concepto también puede usarse la barra de búsqueda, seleccionando la pestaña de **Buscar Código** (fig. 19). Esta pestaña contiene un espacio editable para introducir el código a buscar, un botón de opciones para seleccionar qué datos queremos que se muestren, y el botón (**Buscar Código**) para iniciar la búsqueda.

Una pestaña de datos con todos los datos mostrará un recuadro con los datos básicos, otro con una lista de los padres del concepto y otra con los hijos. Al igual que con las pestañas de resultados, se puede seleccionar un resultado de estas listas de padres o hijos y pulsar el botón para buscar automáticamente los datos. El resultado será el mismo que usando la barra de búsqueda.

#### 4.5.4. Consultas semánticas

Investigar cómo realizar consultas más complejas aprovechando tanto la estructura de SNOMED como los datos adicionales introducidos por la distribución de UMLS era uno de los objetivos de este proyecto.

Sin embargo, el hecho de que los datos de SNOMED en español cargados estén incompletos en lo que respecta a las relaciones y atributos propios de la terminología nos impide un desarrollo más exhaustivo por el momento.

<b><i>Función</i></b>	<b><i>Entrada</i></b>	<b><i>Salida</i></b>	<b><i>Descripción</i></b>
areCodesKin	Dos códigos de concepto	Si o No	Comprueba si los dos conceptos introducidos son hijos del mismo padre
getGeneration	Código de concepto, número de generación, flag indicando si buscamos antecesores o descendientes	Lista de conceptos	Devuelve una generación concreta de padres o hijos del concepto indicado

Las dos funciones extendidas implementadas son areCodesKin y getGeneration (ver tabla V).

La primera (ver fig. 20) permite consultar si dos conceptos descienden del mismo concepto padre (son hermanos). El algoritmo utilizado es sencillo: tras buscar los padres de ambos conceptos, se comprueba si alguno de ellos coincide. El hecho de que tan sólo una parte de estas relaciones IS\_A esté presente en la base de

datos limita la precisión del algoritmo.

```
public boolean areCodesKin(String code1, String code2) {  
  
    inicializa los servicios si aun no se ha hecho  
  
    l_padres1 = busqueda(code1, propiedad: PAR)  
    //PAR= parrents, padres  
    l_padres2 = busqueda(code2, propiedad: PAR)  
    //equ. a IS_A  
  
    for(cada padre1 en l_padres1)  
        for(cada padre2 en l_padres2)  
            if(padre1 = padre2) return  
                true;  
  
    return false;  
}
```

Figura 20. Algoritmo de areCodesKin

Para su ejecución se utiliza la función de CTS “lookupCodeExpansion”, que busca las propiedades indicadas de un concepto. La propiedad “PAR” (padres) en LexGrid equivale a la propiedad “IS\_A” de SNOMED, de modo que la función devolverá los antecedentes del concepto indicado. Una vez extraídos, la comparación y obtención de respuesta por parte de areCodesKin mediante un bucle es directa.

La segunda función, getGeneration (ver fig. 21), es una extensión de las funciones getParents y getChildren. Su objetivo es buscar una generación concreta de antecesores o descendientes, según se indique.

```

public boolean getGeneration(String code, bool parents, int
generation, int maxResults) {

    inicializa los servicios si aun no se ha hecho

    listal <- code;

    for(i=0 y hasta i=generation){
        if(parents) //busca generacion de padres
            lista2 <- busqueda(listal[j],
propiedad: PAR)
        else //busca generaci3n de hijos
            lista2 <- busqueda(listal[j],
propiedad: hasSubtype)

        listal = lista2
    }//fin del for

    return listal;
}

```

Figura 21. Algoritmo de getGeneration

El algoritmo simplemente realiza una b3squeda de una generaci3n cada vez, hasta que alcanza la indicada. Notar que para buscar hijos se utiliza la propiedad de LexGrid “hasSubtype”, que en SNOMED corresponde a la propiedad (relaci3n) “REVERSE\_IS\_A”.

El estado de la base de datos actual no permite la consulta de atributos y relaciones propias de SNOMED m3s all3 de las relaciones “IS\_A” y “REVERSE\_IS\_A”. Por este motivo, el resto de las funciones avanzadas no han podido implementarse, aunque su dise1o es posible y una vez cargada la nueva base de datos su incorporaci3n ser3a sencilla.

Un ejemplo de algoritmo avanzado que no ha sido implementado es la siguiente b3squeda jer3rquica que se muestra en la figura 22.

```

public boolean hierarchySearch(String words, String []
hierarchy, int maxResults) {

    inicializa los servicios si aun no se ha hecho

    lista1 <- descSearch(words, maxResults);

    for(i=0 y hasta i=tamaño de lista1){ //para cada
    elemento encontrado
        ok=false
        for(j=0 hasta j=tamaño de hierarchy o
        ok=true)
            //si pertenece a una de las jerarquias
            que buscamos
                if(areCodesRelated(lista1(i),
                hierarchy)){
                    lista2 <- lista1(i) //añade a
                    la lista
                    ok=true
                }
            } //fin del for hierarchy
    } //fin del for lista1

    return lista2;
}

```

Figura 22. Algoritmo propuesto de búsqueda con restricción de jerarquía en SNOMED

Este algoritmo realiza una búsqueda y filtra los resultados para devolver únicamente aquellos que pertenezcan a las jerarquías solicitadas.

## 5. Resultados y Conclusiones

Al finalizar este proyecto de fin de carrera se han cumplido los siguientes objetivos:

- 1) Se han investigado las diferentes herramientas libres de acceso a terminologías.
- 2) Empleando dos de estas herramientas (MetamorphoSys de UMLS y LexBIG/LexEVS), se ha creado un servidor de terminología SNOMED CT en español de Argentina extraída de una distribución de UMLS sobre una base de datos MySQL.
- 3) Así mismo se han implementado las funciones de consulta básicas para acceder al contenido terminológico almacenado en la base de datos apoyándose en otras dos herramientas (las APIs de CTS y su implementación proporcionada por LexBIG/LexEVS para acceder a los datos en el formato almacenado). Entre las funciones implementadas se encuentran la búsqueda por propiedades y por código y la extracción selectiva de datos dado un código de concepto.
- 4) Se han implementado dos de las funciones avanzadas que se pretendía: una función de búsqueda de una generación concreta de antecedentes o descendientes de un término, y una función que determina, dados dos códigos de términos distintos, determinar si son hermanos (descendientes del mismo concepto padre)
- 5) Y a su vez se ha implementado una Interfaz Gráfica de Usuario que emplea dichas funciones para realizar consultas de manera intuitiva.

## 6. Trabajo Futuro

El trabajo futuro que consideramos interesante realizar para sacar mayor partido a la herramienta C.T.Hunter en este proyecto se divide en los siguientes puntos.

- 1) Consideramos muy importante, por su utilidad, implementar el servicio web que permita la publicación de la funcionalidad de C.T.Hunter como servicio. Entre otras cosas, permitiría que otras herramientas, como LinKEHR [35] del grupo IBIME, pudieran acceder al contenido terminológico.
- 2) Cargar la base de datos original de SNOMED en inglés, que permitiría las búsquedas más complejas puesto que incorpora la tabla completa de relaciones entre conceptos de diferentes jerarquías (relaciones de asociación) que aportan valor semántico. La base de datos cargada, SNOMED en español desde UMLS, al tratarse de una versión antigua, carece de las relaciones propias de SNOMED que se pretendían aprovechar para las búsquedas avanzadas.
- 3) Cargar los archivos de SNOMED-CT facilitados por el Ministerio de Sanidad. La herramienta LexGrid está preparada para la carga de datos en muchos formatos, OWL, OBO, RRF y Red Semántica de UMLS, XML, etc. Sin embargo no puede cargar directamente los archivos txt en el formato ofrecido por IHSTDO a través del Ministerio. La carga de estos datos será importante, ya que la traducción al español actual de SNOMED-CT corresponde al dialecto argentino, de modo que hay términos que no se corresponden con los usados en el español de España. El Ministerio pretende llevar a cabo una nueva traducción y es de suponer que los nuevos archivos estén en el mismo formato que los actuales.

Una posibilidad viable es actualizar la propia base de datos de

MySQL con un script. Esto permitiría además conservar las propiedades de los datos extraídos de UMLS, y realizar actualizaciones periódicas de los datos cuando fuera necesario. Otras posibilidades incluyen usar las propias funciones de administrador de LexBIG para crear una función de carga adaptada a los archivos de IHTSDO (pero estos datos no contendrían los datos extra de UMLS), o utilizar el script en Perl ofrecido por IHTSDO para traducir sus ficheros a formato OWL y cargarlos usando la función correspondiente de LexBIG (pero además de que tampoco contendría los datos de UMLS, el propio LexBIG es incapaz de cargar archivos tan grandes -150MB- si no es en modo de carga UMLS). Existe también la posibilidad de que próximas actualizaciones de LexBIG permitan cargas de archivos OWL más grandes, o una función adaptada al formato de SNOMED, pero por el momento el script para modificar la base de datos es la opción viable y la más conveniente.

- 4) Implementar las funciones avanzadas que aprovechen por completo las propiedades y atributos de la terminología SNOMED cuando la base de datos esté completa. Algunos ejemplos serían las búsquedas de términos por múltiples propiedades y restringiendo por tipo de resultado (por ejemplo, buscar únicamente enfermedades -restricción de tipo- que provoquen cierto síntoma -propiedad 1- y durante cierto tiempo -propiedad 2- ). El uso de LexBig y UMLS permitiría también la traducción de términos a cualquier otra terminología cargada en la base de datos por medio del CUI (Common Unique Identifier de UMLS). De hecho se podrían extender las búsquedas simples para permitir elegir sobre qué terminología realizar las búsquedas.



## 7. Agradecimientos

A la Cátedra de Tecnologías para la Salud y el Bienestar de la Universidad Politécnica de Valencia y a INDRA por financiar y promover el desarrollo de este Proyecto de Fin de Carrera y al grupo IBIME de Ítaca (UPV) por toda la ayuda y conocimientos prestados para realizarlo.

## 8. Glosario

ADL: Lenguaje de representación de arquetipos (Arquetype Description Language)

API: Interfaz de Programación de Aplicaciones (Application Programming Interface), es un conjunto de funciones y procedimientos que ofrece una cierta biblioteca para ser utilizado por otro software.

FAQ: Frequently Asked Questions (Preguntas Más Frecuentes), se refiere a una lista de preguntas y respuestas que surgen frecuentemente dentro de un determinado contexto y para un tema en particular.

HL7: Health Level 7, organización que desarrolla especificaciones de estándares para el intercambio electrónico de información médica.

IGU: Interfaz Gráfica de Usuario, es la parte del programa que un usuario ve en su pantalla.

OWL: Acrónimo del inglés Web Ontology Language (Lenguaje Web de Ontologías), un lenguaje de marcado para publicar y compartir datos ontológicos.

IHSTDO: Acrónimo de International Health Terminology Standards Development Organisation (Organización Internacional para el Desarrollo de Estándares de Terminología Sanitaria), es una asociación sin ánimo de lucro que desarrolla y promueve el uso de SNOMED CT.

ISO: International Organization for Standardization, organización internacional de estandarización

NCBO: Acrónimo de National Center for Biomedical Ontology (Centro Nacional de Ontologías Biomédicas), es un consorcio de biólogos, clínicos, informáticos y ontólogos para desarrollar tecnologías de información biomédica.

OBO: Acrónimo de Open Biomedical Ontologies (Ontologías Biomédicas Abiertas, o Libres) es un proyecto para crear vocabularios controlados para uso compartido en diferentes dominios biológicos y médicos. Elemento central de Bioportal.

RRF: Rich Release Format, el formato de archivo utilizado por UMLS para almacenar sus datos

SQL: Es un lenguaje formal declarativo, estandarizado ISO, para manipular información en una base de datos.

## 9. Bibliografía

- [1] SNOMED RT: A Reference Terminology for Health Care; Kent A. Spackman, Roger A. Cote  
[http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2233423/pdf/procamia\\_efs00001-0675.pdf](http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2233423/pdf/procamia_efs00001-0675.pdf)
- [2] SNOMED CT, <http://www.ihtsdo.org/snomed-ct/>
- [3] Bader F, Calvanese D, McGuinness DL, Nardi D, Patel-Schneider PF, The description logic handbook theory, implementation and applications (2<sup>nd</sup> edition. Cambridge: Cambridge University Press, 2007
- [4] IHTSDO, <http://www.ihtsdo.org/>
- [5] Unified Medical Language System (UMLS), <http://www.nlm.nih.gov/research/umls/>
- [6] SNOB: A SNOMED Browser, <http://snob.eggbird.eu/>
- [7] SnoFlake de Dataline, <http://snomed.dataline.co.uk/>
- [8] CliniClue Xplore, <http://www.cliniclue.com/>
- [9] Bioportal de NCBO, <http://bioportal.bioontology.org/>
- [10] Common Terminology Services (HL7/ISO CTS), <http://informatics.mayo.edu/LexGrid/index.php?page=ctsspec>
- [11] LexGrid, <http://informatics.mayo.edu/LexGrid/>
- [12] Desiderata for Controlled Medical Vocabularies in the Twenty-First Century; James Cimino; Methods Inform Med 1998; 37:394-403

[13] Barry Smith, From concepts to clinical reality: an essay on the benchmarking of biomedical terminologies. Journal of Biomedical Informatics, Volume 39, Issue 3 (June 2006), Special issue: Biomedical ontologies, pages: 288 – 298, ISSN:1532-0464

[14] LOINC, <http://www.loinc.org/>

[15] ICD, <http://www.who.int/classifications/icd/en/>

[16] MeSH, <http://www.nlm.nih.gov/mesh/>

[17] HL7, <http://www.hl7.org/>

[18] SnoCAT, a SNOMED Categorizer, <http://eagl.unige.ch/SNOCat/>

[19] SNOMECE CT Browser de VetMed,  
<http://terminology.vetmed.vt.edu/SCT/menu.cfm>

[20] SnoCode de MedSight, <http://www.medsight-info.com/>

[21] Protégé, <http://protege.stanford.edu/>

[22] OWL Overview, <http://www.w3.org/TR/owl-features/>

[23] Applications of SNOMED and DL in Kaiser's EHR; Peter Hendler MD

[http://esw.w3.org/images/1/16/HCLSIG\\$\\$Meetings\\$\\$2009-11-02\\_F2F\\$PeterHendlerHCLS2009.ppt](http://esw.w3.org/images/1/16/HCLSIG$$Meetings$$2009-11-02_F2F$PeterHendlerHCLS2009.ppt)

[24] National Center for Biomedical Ontology (NCBO),  
<http://www.bioontology.org/>

[25] FAQ de UMLS,  
[http://www.nlm.nih.gov/research/umls/faq\\_main.html](http://www.nlm.nih.gov/research/umls/faq_main.html) (último acceso 24-4-2010)

[26] Ministerio de Sanidad, Snomed CT y la Historia Clínica Digital del SNS,

<http://www.msps.es/profesionales/hcdsns/areaRecursosSem/snomed-ct/snomedHCD.htm>

[27] SNOMED CT User Guide,

[http://www.ihtsdo.org/fileadmin/user\\_upload/Docs\\_01/SNOMED\\_CT\\_Publications/SNOMED\\_CT\\_User\\_Guide\\_20080731.pdf](http://www.ihtsdo.org/fileadmin/user_upload/Docs_01/SNOMED_CT_Publications/SNOMED_CT_User_Guide_20080731.pdf)

[28] Foundational Model of Anatomy,

<http://sig.biostr.washington.edu/projects/fm/AboutFM.html>

[29] MySQL, <http://www.mysql.com/>

[30] ISO, <http://www.iso.org>

[31] IETF RFC 3066, <http://www.ietf.org/rfc/rfc3066.txt>

[32] caBIG, <https://cabig.nci.nih.gov/>

[33] EVS, <https://cabig.nci.nih.gov/concepts/EVS/>

[34] SWING,

<http://java.sun.com/javase/6/docs/technotes/guides/swing/>

[35] LinkEHR (<http://www.linkehr.com/>)