Escuela Técnica Superior de Ingeniería Informática

Universitat Politècnica de València

IMPLEMENTACIÓN DE TÉCNICAS DE STRING MATCHING Y SELECCIÓN SEMÁNTICA APROXIMADA EN UN MOTOR DE NORMALIZACIÓN TERMINOLÓGICA

Proyecto Final de Carrera

Ingeniería Informática

Autora: Estíbaliz Parcero Iglesias

Directores: Montserrat Robles Viejo

Jose Alberto Maldonado Segura

25 de Septiembre de 2012

A mi madre, Jacinta, a Alfonso
y a la memoria de mi padre.

Resumen

En este proyecto se presenta un sistema de normalización terminológica que busca conseguir un sistema eficaz en base a número de términos correctamente *mapeados* e integrarse dentro de un servidor terminológico. Se implementa un motor de comparación de términos basado en técnicas de correspondencia aproximada de cadenas (*approximate string matching*), ampliamente utilizadas en *data mining* y deduplicación de datos. Se estudiaron estas técnicas concluyendo que la técnica Jaccard consigue la mayor eficacia dentro del servidor terminológico objetivo. Una vez establecida la base teórica se describe la herramienta implementada como una aplicación web e integrada dentro del servidor terminológico. Se le añaden sustanciales mejoras que consiguen aumentar significativamente la eficacia. En conclusión se consigue una herramienta que centraliza el proceso de normalización terminológica facilitando y ahorrando tiempo al usuario.

Palabras clave: normalización terminológica, servidor terminológico, LOINC, *string matching*, *mapping*.

Tabla de contenidos

1. Introducción

La necesidad de utilizar vocabularios normalizados es vital para el uso, comunicación e integración de la información sanitaria. Se persigue que los conceptos clínicos estén representados por medio de términos que un sistema informático pueda entender de modo que puedan ser explotados por éstos [1]. Por tanto, las terminologías clínicas están orientadas y pensadas para dar soporte al *software* clínico[2], como por ejemplo a historia clínica electrónica, sistemas de ayuda al diagnóstico médico, generadores de extractos de información clínica, guías clínicas electrónicas, sistemas de alerta frente a emergencias médicas, seguimiento de datos demográficos, calidad asistencial, salud pública, etc.

La adopción de una terminología pública, controlada y bien formada facilita la comparación, consistencia e interoperabilidad de los datos [3]. Sin embargo, la mayoría de centros sanitarios utiliza una terminología y codificación propia, esto dificulta la comunicación entre ellos e impide la reutilización de datos y *software*. El uso de una terminología común permitiría, junto con un modelo de datos común, el intercambio de datos clínicos entre sistemas independientes en pos de mejorar la atención al paciente y facilitar las actividades de investigación clínica.

Este trabajo se centra en la elaboración de una herramienta de normalización terminológica en LOINC para pruebas clínicas de laboratorio. A continuación se detallan los principales problemas a los que nos enfrentaremos.

Además de existir múltiples sistemas terminológicos y de codificación propia, nos encontramos con la inmensidad y continua evolución del lenguaje clínico y las sutiles diferencias existentes entre los conceptos, aparentemente muy parecidos. Por tanto, la normalización de la terminología propietaria, es decir, la terminología local de un centro, consume mucho tiempo y es un proceso muy especializado que requiere personal cualificado.

El proceso de normalización requiere herramientas de alineación. Estas herramientas, dado un término local, buscan el término más adecuado de una terminología estándar. Sin embargo, las herramientas de alineación existentes no están integradas en el sistema, de manera que se necesita un procesado manual previo de una batería de términos para su posterior análisis y búsqueda de posibles candidatos en la terminología objetivo. La búsqueda del candidato correcto es realizada por un especialista que evalúa la correspondencia entre términos basándose en la información que le ha proporcionado el centro de origen. Y, por último, la incorporación de los resultados al servidor terminológico (lugar donde se almacenan los resultados) necesitará un procesado posterior generalmente también manual.

Otro de los problemas que nos encontramos, es la posibilidad de que coexistan varios idiomas en un mismo servidor terminológico, ya que cada centro clínico puede utilizar un idioma distinto o incluso utilizar varios indistintamente.

Para resolver estos problemas en este PFC se propone el desarrollo de un motor de comparación de pruebas de laboratorio para facilitar el trabajo del técnico especialista

en normalización. El motor es el encargado de realizar las comparaciones entre los términos estandarizados y los que no lo están.

La implementación de una interfaz gráfica facilita la evaluación de los resultados facilitados por del motor de comparación.

Además, ambas partes de la aplicación se integrarían en el sistema, de modo que pueda existir conexión directa al servidor terminológico, facilitando así la recuperación e incorporación de datos.

En nuestro afán de ahorrar tiempo al usuario, se plantea la implementación dentro del motor de comparación de un tipo de técnicas que puedan ejecutarse sin la necesidad de un preprocesado previo de los términos a analizar, por tanto, necesitamos técnicas que no sean sensibles a errores tipográficos, que detecten similitudes en la raíz de las palabras para permitir variaciones léxicas o incluso la detección de raíces comunes entre términos en distintos idiomas (frecuente en el vocabulario clínico). Unas técnicas que encajan en este contexto son las técnicas de correspondencia aproximada o *approximate string matching*.

Objetivos

Desarrollar un sistema de normalización terminológica en LOINC, que además de unificar los procesos de importación de conjuntos de pruebas no estandarizadas, comparación con el sistema de referencia y exportación de las pruebas una vez estandarizadas al servidor de terminología, automatice el proceso de comparación.

Evitar al técnico especialista la tarea de preprocesado de términos locales (reescritura del término, traducción al idioma de la terminología de referencia, completado de información, etc).

Reducir el tiempo que el especialista debe dedicar a la tarea de búsqueda del término estándar apropiado mediante la selección y aplicación automática de reglas establecidas para la obtención de listas de términos ya normalizados, ordenados según un coeficiente de similitud, proporcionado por una técnica basada en la correspondencia por cadenas.

Realimentar la base de conocimiento de correspondencia semántica entre términos locales y estandarizados (el servidor terminológico) de manera que la incorporación de los datos a ésta sea sencilla.

Facilitar la propuesta de incorporación de nuevos términos en la terminología estándar que actualmente no existen.

Estructura de la memoria

En primer lugar se presenta el contexto en el que se ha desarrollado este trabajo.

A continuación se describe la normalización terminológica con LOINC [4], haciendo previamente un recorrido por las terminologías de ámbito clínico, explicando concretamente la terminología utilizada en este proyecto, LOINC, y por último, se presenta el servidor terminológico con el trabajaremos, el servidor terminológico de BITAC, la empresa para la cual se ha desarrollado la herramienta.

Posteriormente se describe el estado del arte sobre técnicas de correspondencia aproximada que se usarán en el motor de comparación y sobre herramientas existentes para normalización terminológica.

En capítulo 5 se describen los materiales y métodos utilizados para las dos principales partes del proyecto: el motor de comparación y la interfaz gráfica

En el capítulo 6 se realiza la descripción de la herramienta desarrollada: funcionamiento general, detalles de implementación, mejoras funcionales y descripción de los casos de uso.

Finalmente se muestran los resultados obtenidos, las conclusiones de nuestro trabajo y las líneas de trabajo futuro para este proyecto.

2. Contexto de Trabajo

IBIME

Este proyecto final de carrera ha sido llevado a cabo en el grupo de investigación IBIME (Informática Biomédica). Este grupo de investigación pertenece al instituto ITACA (Instituto de Aplicaciones de las Tecnologías de la Información y las Comunicaciones Avanzadas) de la UPV (*Universitat Politècnica de València*). IBIME está centrado en el uso y desarrollo de métodos y herramientas para la adquisición, procesado y gestión de datos y conocimiento biomédico. Se trata de un equipo de investigación multidisciplinar en estrecha relación con las instituciones y profesionales de la salud.

BITAC

BITAC es una pequeña empresa afincada en Barcelona dedicada a tareas de normalización de datos de laboratorio, utilizando la terminología LOINC, para diversos centros hospitalarios. Son miembros de la Comunidad LOINC y colaboran en el desarrollo de su vocabulario. La primera finalidad de este proyecto ha sido desarrollar un producto final para la empresa BITAC que les ayude a conseguir uno de sus objetivos, consolidar el uso de LOINC.

3. Normalización terminológica con LOINC

Como ya se ha indicado anteriormente el objetivo principal de este proyecto es el desarrollo de una herramienta de normalización terminológica utilizando el estándar de facto LOINC. En esta sección se da una introducción a las terminologías clínicas, poniendo especial hincapié en la terminología LOINC. Además se introduce el sistema terminológico de BITAC, que será el utilizado para el desarrollo de nuestra Sistema de Ayuda a la Normalización Terminológica, SANT.

Terminologías clínicas

Las terminologías clínicas o sistemas de codificación son listas estructuradas de términos asociados a sus definiciones encargadas de describir de manera clara una gran variedad de conceptos clínicos relacionados en última instancia con el cuidado y tratamiento del paciente.

Estos términos cubren conceptos de diferentes tipos como enfermedades, diagnósticos, hallazgos clínicos, operaciones, tratamientos, fármacos, pruebas de laboratorio, procedimientos, etc., y pueden ser usados para completar y desambiguar conceptos dentro de la historia clínica del paciente, sea historia clínica en papel o historia clínica electrónica.

La codificación clínica y los sistemas de clasificación terminológica forman parte del camino hacia la implementación de un lenguaje estandarizado para la salud: un lenguaje común (informatizado) de uso global [5]. Con dicho lenguaje común todos los actores que intervienen en el proceso de intercambio de información clínica podrán compartir información relativa a los conceptos detallados anteriormente de manera ordenada y exenta de ambigüedades.

En este sentido se han realizado muchos esfuerzos por conseguir la normalización terminológica de términos clínicos, por ello se han desarrollado una gran cantidad de terminologías que abarcan distintos dominios. Algunos ejemplos se enumeran a continuación:

NANDA, NIC, NOC cubren aspectos de diagnóstico, intervenciones y objetivos en el campo de la enfermería.

ICD es la clasificación internacional de enfermedades. Se utiliza para la codificación de enfermedades en multitud de registros incluyendo certificados de defunción o historias clínicas.

SNOMED CT es una terminología clínica integral compuesta por conceptos, descripciones y relaciones. Sus conceptos engloban diversos aspectos de la práctica clínica tales como hallazgos clínicos, procedimientos, organismos, conceptos anatómicos, etc. Los conceptos tienen asociadas descripciones que son los términos o nombres asignados a dicho concepto. Las relaciones sirven como enlace entre conceptos de SNOMED CT.

MedDRA, el Diccionario Médico para las Actividades Reguladoras, es una terminología médica utilizada para clasificar información relativa a acontecimientos

adversos asociados al uso de productos farmacéuticos, de dispositivos médicos o vacunas. Su uso permite el intercambio y análisis de datos relativos a la seguridad de estos productos.

MeSH (*Medical Subjects Headings*) es una agrupación de términos relevantes que conforman un vocabulario controlado para la clasificación de artículos y publicaciones científicas.

UMLS es un Sistema de Lenguaje Médico Unificado. Recoge un conjunto de vocabularios médicos y de herramientas que los enlazan a fin de facilitar la interoperabilidad entre sistemas informáticos.

LOINC es la terminología de interés en este proyecto puesto que se utiliza para normalizar pruebas de laboratorio. En el siguiente apartado se describe de forma detallada.

LOINC

La base de datos LOINC proporciona un conjunto de nombres universales y códigos identificativos para pruebas de laboratorio y para observaciones clínicas. LOINC facilita el intercambio y almacenamiento de términos como hemoglobina en sangre, potasio en suero o signos vitales, para su uso en medicina, en investigación, o en gestión de recursos.

Los códigos LOINC no pretenden abarcar toda la información relativa a una prueba de laboratorio o a una observación clínica, si no identificar inequívocamente dicha prueba u observación. De esta manera el código LOINC puede formar parte de un mensaje que contenga más campos que completen la información por ejemplo para un paciente dentro de su historia clínica electrónica o para una población dentro de un informe epidemiológico.

Cada término codificado en LOINC incluye seis campos o ejes que especifican:

1. Componente o analito. Ej: potasio, hemoglobina, antígeno de la hepatitis C…
2. Propiedad medida. Ej: Concentración de masa, actividad enzimática…
3. Temporalidad, es decir, si la medida es una observación en un momento determinado del tiempo o es durante un periodo de tiempo. Ej: Orina 24 horas.
4. Muestra. Ej: Orina, sangre, suero…
5. Tipo de escala. Ej: si la medida es cuantitativa, ordinal, nominal o narrativa.
6. Método, el utilizado en la medida, se especifica siempre que sea relevante, es decir, sólo cuando proporcione distinción con respecto a otro tipo de método por su relevancia clínica. Ej: radioinmunoensayo, PCR…

Un término LOINC se describe formalmente con la siguiente sintaxis:

<Componente>:<Propiedad>:<Temporalidad>:<Muestra>:<Escala>:<Método>

Los dos puntos forman parte del nombre formal y sirven para separar las principales partes del nombre, es decir, los seis ejes de LOINC.

A cada término resultante de una única combinación de los seis ejes se le asigna un código único y permanente que servirá para identificar pruebas de laboratorio en informes electrónicos.

Código LOINC	Término
25428-4	Glucose:ACnc:Pt:Urine:Ord:Test strip
4546-8	Hemoglobin A/Hemoglobin.total in Blood:MFr:Pt:Bld:Qn
19146-0	Reference lab test results:Find:XXX:Reference lab test:Nar

Tabla 1: Códigos y términos LOINC

El servidor terminológico de BITAC

El servidor terminológico o banco de datos utilizado en este proyecto es propiedad de la empresa BITAC, especialistas en normalización de datos de laboratorio, y hace uso de la terminología LOINC como terminología estándar.

Como punto de partida se dispone de este banco de datos compuesto por pruebas clínicas de distintos centros codificados según su sistema local. Junto a estos términos locales está la terminología de referencia LOINC y todos estos términos están asociados a un sistema de codificación del servidor terminológico denominado código Bitac encargado de relacionar términos locales con términos LOINC.

Así pues, es posible agrupar los términos por código Bitac, obteniendo un conjunto de sinónimos para un concepto dado. Este grupo es interesante por la información que aporta ya que puede incluir distintas formas de referirse a un concepto, términos con mayor o menor información relevante, y equivalencias de conceptos entre los idiomas utilizados en el banco de datos como se puede observar en el ejemplo siguiente:

Código Local	Componente	Código Bitac
488	ADH (VASOPRESINA)	1003126
BIO.1.2003	ADH(ARGININE VASOPRESSIN)	1003126
6892	ADH HORM. ANTIDIURETICA	1003126
1589	ADH plasma	1003126
3126-0	***Vasopressin***	***1003126***
7045	Vasopresina	1003126
204	ADH (Hormona Antidiuretica)	1003126

Tabla 2: Pruebas de laboratorio del servidor terminológico relacionadas por un mismo código Bitac

Las nuevas pruebas procedentes de centros externos se integran en el banco de datos sin enlazar en un primer momento a la terminología de referencia, con el código Bitac a cero. Se almacenan en su formato original, tal y como proviene del centro, con su codificación local.

En la siguiente tabla tenemos algunos ejemplos de pruebas para las que aún no se ha encontrado una correspondencia con la terminología de referencia:

Código Local	Componente	Prop.	Tiempo	Muestra	Escala	Método
10301	ANTICUERPOS IgM ANTI CARDIOLIPINA [ACA] EN SUERO ACA IgM			SUERO		Enzimoinmuno-análisis
EBG	AC.IgG ANTI-EPSTEIN BARR			SUERO	N	
7093	Kiwi					
1589	PCR HERPESVIRIDAE			L.C.R.	Ord	
2590	ACIDO VANILMANDÉLICO (HISTORICO			ORINA DE 24 HORAS + ClH		Cromat. intercambio iónic
1761	Seleni a sèrum					
2772	Subpoblaciones Linfocitarias				Texto	Citometria de flujo
7609	Panipenem				Antibiótico	

Tabla 3: Pruebas de laboratorio del servidor terminológico sin relacionar con la terminología de referencia

Es habitual encontrar entre las pruebas locales términos con ejes no informados (ninguna de estas pruebas contiene información en el eje “propiedad”), ejes con información incorrecta (la prueba 7609 contiene una información errónea en el eje “escala”) o ejes con información que corresponde a un eje, o incluso a varios ejes distintos (las pruebas 10301 y 1761 contienen la información relativa al eje “muestra” en el eje “componente”).

4. Estado del arte

A continuación se expone el trabajo de otros autores en relación al tipo de técnicas que usará el motor de comparación de SANT, las técnicas de correspondencia aproximada de cadenas, y en relación a otras herramientas de normalización terminológica.

Correspondencia aproximada de cadenas

La correspondencia aproximada de cadenas es un tipo de técnica de búsqueda de cadenas que consiste en encontrar patrones coincidentes de manera aproximada entre las cadenas comparadas.

La correspondencia aproximada de cadenas se usa en varias áreas de la informática. Se ha utilizado en biología computacional, un área que ha cobrado gran importancia en los últimos años, para detectar correspondencias entre series de proteínas o de ADN. Estas secuencias representan el código genético de los seres vivos y buscar secuencias específicas resulta útil para localizar determinadas propiedades en la cadena de ADN, sin embargo, raramente las secuencias son siempre idénticas, debido a variaciones evolutivas o mutaciones, por tanto, en este campo la correspondencia aproximada de cadenas resulta especialmente útil [6].

En el campo del procesamiento de señales o del reconocimiento de formas también podemos encontrar ejemplos de uso. Es el caso del procesamiento de mensajes de voz que contienen unas órdenes específicas. Problemas en la decodificación, descompresión, grabación, pronunciación o variaciones en la voz puede dar como resultado una palabra o frase diferente a la almacenada en el sistema, de forma que la búsqueda exacta por palabras no resultaría efectiva, y sin embargo sí podría serlo la búsqueda por correspondencia aproximada [7]. Lo mismo ocurre con la detección de escritura manual, en la que un procesamiento de ésta daría lugar a una cadena de caracteres que, al igual que la cadena resultante de un mensaje de voz, puede contener errores, y una búsqueda exacta de cadenas no daría resultado [8].

Y por supuesto, en el campo de la recuperación de textos estas técnicas tienen mucha utilidad, por ejemplo en la búsqueda de textos digitalizados mediante OCR (*optical character recognition*) donde pueden aparecer errores tipográficos y errores en la digitalización [9]. Otras aplicaciones de procesamiento de textos como los correctores ortográficos utilizan estas técnicas para localizar los errores [10].

Un área en el que ha cobrado una gran importancia la correspondencia aproximada de cadenas ha sido en el del *Data Cleaning*. El *Data Cleaning* [11] es un proceso de tratamiento de colecciones de datos, como archivos o bases de datos. Busca conseguir datos limpios de duplicados, corregir errores tipográficos, desarrollar abreviaciones, evitar campos no informados, corregir campos contradictorios, etc.

Herramientas de normalización terminológica

Existen diversos tipos de herramientas para la normalización terminológica. Algunas buscan correspondencias directas con términos a partir de texto libre, como hace MedLine [12], otras herramientas realizan un preprocesado más o menos complejo,

como puede ser hacer una limpieza signos de puntuación o desarrollar las abreviaturas, a fin de encontrar correspondencias por búsqueda por palabras.

Intelligent Mapper es una aplicación para codificar pruebas locales de radiología en inglés a la terminología LOINC. Busca correspondencias directas entre palabras, y para que el proceso sea efectivo es necesaria una normalización previa de los términos que componen la descripción de la prueba [13].

Otro enfoque distinto a lo realizado en *Intelligent Mapper* es utilizar tablas de sinónimos para cada uno de los ejes de LOINC, de modo que si una prueba de laboratorio está dentro de una tabla de sinónimos que la relaciona con el término LOINC correspondiente se encuentra una correspondencia directa [14].

5. Materiales y métodos

En este apartado se describe el material y los métodos utilizados para conseguir alcanzar el objetivo de nuestro proyecto. La sección está dividida en tres subapartados correspondientes a las principales etapas de desarrollo: implementación del motor de comparación, implementación de la interfaz gráfica e implementación de mejoras.

El motor de comparación

A fin de comparar las pruebas de laboratorio locales con las ya *mapeadas*[1] en el servidor terminológico, se implementa un motor de comparación basado en técnicas de *approximate string matching*.

El motor de comparación que usará SANT utiliza una de las siguientes técnicas analizadas. A continuación se describen dichas técnicas y los experimentos realizados para la elección de la técnica más apropiada.

Modelos matemáticos de string matching

Para calcular la similitud entre términos se hace uso de técnicas de búsqueda aproximada de cadenas, más comúnmente conocidas por su término en inglés *approximate string matching* [15].

Estas técnicas consisten en aplicar una función de similitud a dos cadenas *simil(c1, c2)*, donde *c1* es la cadena para la cual queremos encontrar una coincidencia y *c2* es una cadena de nuestro documento objetivo (en nuestro caso el documento objetivo es un servidor terminológico), y obtener a partir de ella una puntuación.

Si dicha puntuación supera un umbral μ, ajustado según conveniencia, se añade ésta al conjunto de resultados. La cadena con la mejor puntuación será la candidata principal.

Las técnicas implementadas en el motor de comparación son aquellas basadas en la descomposición del término en unidades básicas. Este proceso se denomina *tokenización*, y como resultado del proceso obtendremos un conjunto de *tokens*, o lo que es lo mismo, de Q-gramas, es decir, subcadenas de tamaño Q [16]. Veamos una explicación más detallada a continuación.

[1] El término *mapping* proviene del término inglés *Data Mapping*, muy utilizado en la literatura, que indica el proceso de establecer relaciones sinónimas entre dos conceptos.

Dada una cadena *c* se introducen caracteres de inicio y de final de cadena (# y $ u otros símbolos no existentes en el alfabeto utilizado) y se obtiene una lista de Q-gramas mediante el uso de una ventana de tamaño *q* que se deslizará a través de los caracteres de la cadena:

Cadena original c	Posición de la ventana	Q-Grama extraído	Lista de Q-gramas
Glucosa	[#G]lucosa$	[#G]	[#G]
	#[Gl]ucosa$	[Gl]	[#G][Gl]
	#G[lu]cosa$	[lu]	[#G][Gl][lu]
	#Gl[uc]osa$	[uc]	[#G][Gl][lu][uc]
	#Glu[co]sa$	[co]	[#G][Gl][lu][uc][co]
	#Gluc[os]a$	[os]	[#G][Gl][lu][uc][co]
	#Gluco[sa]$	[sa]	[#G][Gl][lu][uc][co][sa]
	#Glucos[a$]	[a$]	[#G][Gl][lu][uc][co][sa][a$]

Figura 1: Ejemplo del proceso de *tokenización*

Un grupo de técnicas de *string matching* hacen uso de las operaciones de teoría de conjuntos aplicadas al conjunto de *tokens* generados por el proceso de *tokenización* para calcular la similitud entre cadenas. Como Intersect y Jaccard [17].

Una especialización de las técnicas descritas son aquellas que además de utilizar los conjuntos de *tokens*, les proporcionan un peso relativo a su frecuencia de aparición, de modo que puntúa más un *token* poco común que uno muy común, dando cuenta de la importancia de cada uno de ellos. La técnica cosineTFIDF [18] es de este tipo.

Otro tipo de funciones utilizadas son las basadas en distancias, como las distancias de edición de Levenshtein [19]. Existen métodos que combinan estas dos ideas, denominados métodos híbridos, como Monge Elkan [20] y SoftTFIDF [21].

Intersect es una técnica sencilla y barata desde el punto de vista computacional. El coeficiente de similitud Intersect entre dos cadenas c1 y c2 es la cantidad de *tokens* presentes en ambas. Dando como resultado un coeficiente sin normalizar:

$$simil_{Intersect}(c1, c2) = |tokens(c1) \cap tokens(c2)|$$

Para obtener un resultado normalizado se propone utilizar el coeficiente de Dice [22], consiguiendo así un resultado entre 0 y 1:

$$simil_{\text{intersect\_dice}}(Q, D) = \frac{2 * |Q \cap D|}{|Q| + |D|}$$

Jaccard es una técnica muy similar a la anterior, pero en este caso su resultado sí se normaliza, y lo hace en base a la unión de *tokens* existentes en ambos conjuntos de *tokens* a comparar. Entonces, el coeficiente de similitud de Jaccard entre dos cadenas *c1* y *c2* es el cociente de la cantidad de *tokens* presentes tanto en *c1* como en *c2* entre la suma de ambas cantidades:

$$simil_{Jaccard}(c1,c2) = \frac{|tokens(c1) \cap tokens(c2)|}{|tokens(c1) \cup tokens(c2)|}$$

Cosine Tf-Idf trata de encontrar la similitud coseno entre dos cadenas *c1* y *c2*. La similitud coseno entre dos vectores v1 y v2 consiste en calcular el producto escalar de ambos y dividirlo por la raíz cuadrada del sumatorio de los componentes del vector *w1* por la raíz cuadrada del sumatorio de los componentes del vector *w2*. Para ello es necesario dar un peso a cada uno de lo *tokens* que indique la relevancia de dicho *token* en términos de frecuencia de aparición, de modo que compongamos los vectores *w1* y *w2*.

$$simil_{CosineTfIdf}(c1,c2) = \sum_{t \epsilon tokens(c1) \cap tokens(c2)} \left(w_1(t, tokens(c1)) \cdot w_2(t, tokens(c2))\right)$$

Y la función para calcular el peso normalizado de un *token* es:

$$w(t,S) = \frac{w\prime(t,S)}{\sqrt{\sum_{t' \epsilon S} w\prime(t,S)^2}} \quad , \quad w'(t,S) = tf(t,S) \cdot idf(t)$$

Donde *tf* (*term frequency*) es la frecuencia del *token* en la cadena e *idf* (*inversal document frequency*) es la frecuencia del *token* en todo el documento o conjunto de cadenas.

Levenshtein consiste básicamente en calcular el coste de transformar una cadena en otra:

$$dist_{\text{Levenshtein}}\,(c1,c2) = \min\,(s_1, s_2, s_3 \ldots)$$

Donde s_1, s_2, s_3... definen el coste de secuencias de operaciones de edición tales como *copia*, *inserción*, *substitución* y *borrado* a realizar para convertir la cadena *c1* en *c2*. El resultado es peor cuanto mayor sea este coste, es decir, las cadenas son más diferentes.

En Monge Elkan el cálculo de la función de similitud se hace a partir del cálculo de las distancias mínimas de edición entre *tokens*, y normalizando con respecto al tamaño en *tokens* de la cadena origen:

$$simil_{MongeElkan}(c1,c2) = \frac{1}{K}\sum_{i=1}^{K}(_j = 1^L \max\left(simil'(c1_i, c2_j)\right)$$

Donde *simil'* es una función secundaria basada en distancias, una variante de la métrica de Levenshtein, *K* es la cantidad de *tokens* de c1 y *L* es la cantidad de *tokens* de c2.

Y por último, en la técnica SoftTFIDF el *token* se pondera según una función entre las frecuencias de aparición del *token* en la cadena a comparar y en el banco de datos general, de manera intuitiva cuanto más frecuente sea el *token* menos significativo resulta.

Elección de la técnica de *string matching*

En esta sección describiremos la fase de experimentación. Mediante los experimentos que a continuación se indican se pretende dar una medida cuantitativa de la eficacia de cada uno de los métodos expuestos anteriormente. Los resultados de estos experimentos nos permitirán tomar una decisión en cuanto a qué método elegir para implementar en nuestra herramienta de normalización terminológica.

Dentro del grupo IBIME, donde se ha desarrollado este proyecto, se realizó un estudio que englobaba diversas técnicas de *string matching* y se analizaban desde un punto estadístico para decidir qué técnica era la más apropiada [23]. Este estudio compara las técnicas Jaccard, Levenshtein, Monge-Elkan y SoftTF-IDF descritas anteriormente y concluye que la técnica Jaccard usando una *tokenización* en Q-gramas de 2 es la más apropiada.

A continuación se describen los experimentos que se han realizado para este proyecto, comparando las técnicas Intersect, Jaccard y CosineTF-IDF anteriormente descritas que complementa al estudio anterior y que llegan a la misma conclusión de que la técnica adecuada para nuestra herramienta es Jaccard utilizando Q-gramas de 2.

Planteamiento

Para realizar los experimentos necesitamos, dado el banco de datos del servidor terminológico B, que contiene por una parte un conjunto de términos de la terminología de referencia LOINC L y por otra un conjunto de términos de diversos centros que han sido normalizados mediante el código Bitac C (compartido por el término LOINC y éste), extraer un conjunto de pruebas P suficientemente grande para ser representativo. Por ello, se escogen 3.000 términos de entre los que conforman el conjunto C, que suponen aproximadamente una décima parte de éstos, asegurándose así que siempre existirá al menos un sinónimo de cada término del subconjunto de prueba, el representante LOINC, dentro del conjunto L.

Este subconjunto P obtenido se elimina del servidor terminológico B para que no interfiera en el análisis y sobre él se aplicarán cada una de las técnicas de *string matching* expuestas para comparar su eficacia.

Entonces, por cada prueba de P se obtiene un conjunto de resultados o candidatos asociados ya a un código enlace con la terminología LOINC, estos resultados están ordenados por la puntuación de similitud.

La eficacia del método analizado la mediremos en función del porcentaje de aciertos, considerando acierto el hecho de encontrar una prueba candidata correcta entre los 10 primeros resultados. Se ha elegido el número 10 porque el técnico especialista puede revisar fácilmente 10 resultados de un solo vistazo y así fue planteado en la especificación de requisitos.

Estos experimentos se harán utilizando, además de las técnicas descritas, distintos tipos de tokenización de manera que comprobemos la eficacia de cada una de las técnicas usando Q-gramas de tamaño 2 y 3.

Resultados

Como resultado de los experimentos, véase figura 2 obtenemos que efectivamente la eficacia de la técnica Jaccard con *tokenización* de Q-gramas de 2 es la mayor, en torno a un 64%, seguida por la técnica Intersect con un 62%, que es muy parecida en implementación, y por último por la técnica Cosine TF IDF, que a pesar de ser una de las técnicas más elaboradas ya que necesita de un preprocesamiento de la base de datos para extraer la frecuencia de cada uno de los *tokens* del documento general y del término analizado, no consigue buenos resultados en comparación, quedándose en el mejor de los casos en un 52% de acierto.

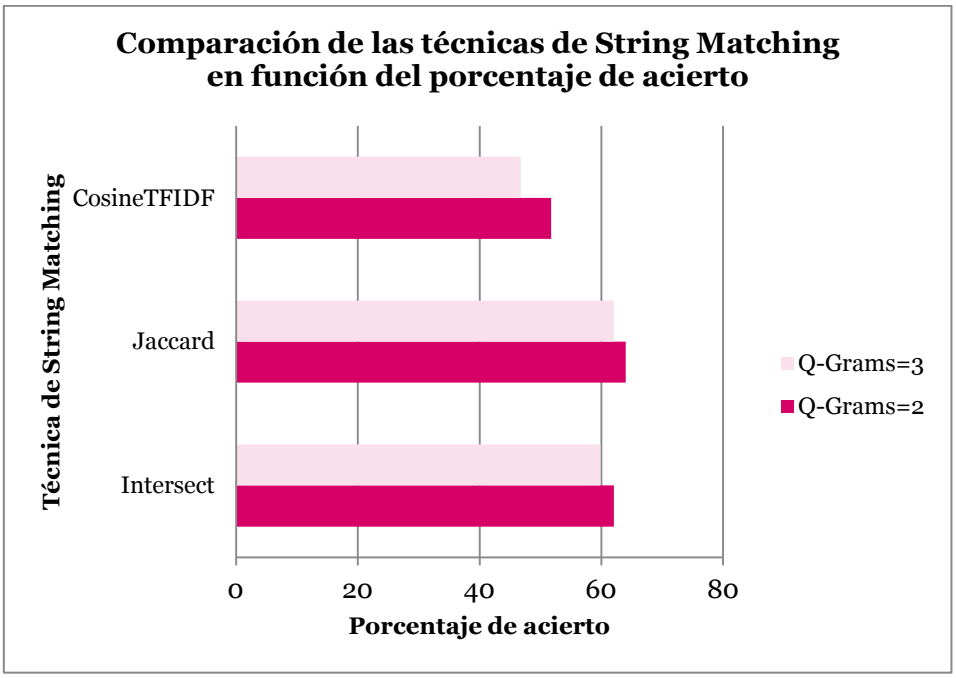

Figura 2: Comparación de las técnicas de *string matching* en base a su eficacia

Por lo tanto estos resultados respaldan también la decisión de tomar la técnica Jaccard con una *tokenización* en Q-gramas de 2 como la adecuada para ser implementada en el motor de comparación de SANT.

Implementación de la interfaz

En este apartado se procede a explicar las decisiones tomadas para la implementación de la interfaz gráfica para nuestro sistema.

Desarrollo web

Uno de los requisitos era obtener una herramienta ubicua, capaz de conectarse con el servidor terminológico, por lo tanto la opción más lógica era decantarse por una aplicación web, accesible desde cualquier equipo con un navegador compatible y una conexión a Internet.

Opciones disponibles

Debido a la complejidad del proyecto afrontado, se quiso elegir un entorno de programación que facilitara el trabajo. En la siguiente tabla se puede observar una tabla que compara los entornos (*frameworks*) más utilizados actualmente desarrollo web:

	GWT 2.4.0	Vaadin 6.8	Wicket 1.5.0
Core UI widgets	✶✶	✶✶✶	✶
No browser plug-in required	●	●	●
No JavaScript programming required	●	●	●
UI programming on client-side	Java, Javascript	Java, Javascript	Javascript
UI programming on server-side	-	Java, JVM languages	Java, JVM languages
Page request oriented			●
Application oriented	●	●	
Free for commercial use	●	●	●
License	Apache 2.0	Apache 2.0	Apache 2.0
Documentation	✶✶	✶✶✶	✶✶
Eclipse IDE support	✶✶✶	✶✶✶	✶✶
Number of downloadable add-ons	-	215*	54*

Tabla 4: Comparación de *frameworks* de desarrollo de aplicaciones web

La decisión sobre qué *framework* de desarrollo elegir se ha hecho en base a las siguientes razones:

- La licencia debe permitir que la aplicación sea comercializada de manera gratuita.
- A ser posible, se tiene que poder desarrollar exclusivamente en Java, tanto la aplicación en sí como las posibles extensiones que se diseñen, para acelerar el desarrollo y disminuir la complejidad del mismo.
- Del mismo modo es deseable que tenga soporte para el IDE Eclipse.
- Dado que se trabajará con datos clínicos, tanto la parte del servidor como la parte del cliente deben poder ofrecer seguridad de base, aunque posteriormente se puedan reforzar estos métodos.
- Debe poseer suficiente documentación.
- Ha de ser de acceso gratuito.

Elegimos Vaadin

Una vez contemplados los requisitos que debe cumplir el entorno de programación seleccionado y si volvemos a consultar la tabla 4, podemos ver que Vaadin[2] es la opción disponible que mejor cumple estos requisitos, aunque GWT también podría haber sido seleccionado, la propiedad de Vaadin de no necesitar implementación en el cliente le da un extra de seguridad, ya que toda la lógica se ejecutará en el servidor y podrá estar mejor protegida, además gracias a esta característica:

- **Menor número de servicios web:** La interfaz de usuario se construye y ejecuta en el servidor, pero es dibujada en el cliente, por lo cual no es necesario crear servicios adicionales para recibir diálogos y descriptores.
- **Agilidad:** Con GWT es difícil mantener la agilidad, ya que no permite cambiar el código final sin tener que compilarlo a JavaScript de nuevo. Vaadin permite combinar y utilizar cualquiera de sus componentes ya compilados sin necesidad de tener que volver a generar el código JavaScript.
- **Capacidad de *testing*:** todo el código puede ser testeado con test jUnit normales, incluso la interfaz de usuario.

Vaadin permite que toda la aplicación esté escrita en Java, incluso las extensiones de dicho entorno, lo cual le conferirá al código de dicha aplicación más estabilidad y legibilidad.

Vaadin es un entorno de trabajo para el desarrollo de aplicaciones web en Java y está diseñado para crear aplicaciones interactivas que funcionen sobre el navegador sin necesitar ningún tipo de añadido, es decir, no requiere instalación de ningún *plugin* en el navegador. Vaadin usa una arquitectura orientada al servidor junto con un modelo de componentes reutilizables para simplificar la programación de las aplicaciones y para ofrecer una mayor seguridad de cara a la aplicación web.

Además Vaadin está diseñado especialmente para construir aplicaciones web, no sólo páginas web. La forma de programar una aplicación con Vaadin es muy similar a la forma en que se programaría una aplicación de escritorio. Esta característica es deseable ya que así se consigue mantener una apariencia de aplicación de escritorio a la que la mayoría de usuarios ya están acostumbrados.

Vaadin tiene una licencia de desarrollo Apache 2.0 lo cual permite al usuario del software desarrollado la libertad de usarlo para cualquier propósito, distribuirlo, modificarlo, y distribuir versiones modificadas de ese software. Esta licencia no exige que las obras derivadas (versiones modificadas) del software se distribuyan usando la misma licencia, ni siquiera que se tengan que distribuir como software libre u *open source*. La Licencia Apache sólo exige que se mantenga una noticia que informe a los receptores que en la distribución se ha usado código con la Licencia Apache.

[2] http://vaadin.com

Como este proyecto ha sido creado dentro de un entorno de trabajo con tiempo y recursos limitados, la eficiencia al desarrollar es muy importante, y Vaadin ofrece:

- **Documentación:** La documentación de Vaadin[3] es muy completa y de calidad, el libro de Vaadin [24], el cual se puede descargar de manera gratuita, simplifica mucho el comienzo del desarrollo.
- **Colaboración de la comunidad:** además de las funcionalidades proporcionadas, también se ofrece en la página web de Vaadin una recopilación de los componentes externos (a*ddons*) creados por la comunidad desarrolladora de este entorno de trabajo. Esto generalmente simplifica mucho el trabajo, ya que las necesidades que puedan surgir debido a la falta de algún componente en Vaadin quedan cubiertas por éstos. Además de esto ofrecen también soporte general del *framework* y de los *addons* a través de foros de consulta.
- **Estilos:** como muchas otras funcionalidades, los estilos son servicios proporcionados por Vaadin. Pero además, si se decide cambiar el estilo por uno distinto a los proporcionados por defecto se puede hacer de manera muy sencilla.
- **Compilación:** sólo es necesario recompilar el código implementado al añadir nuevos componentes.

Número de peticiones reducido

Una de las especificaciones del proyecto es que la herramienta será utilizada por un reducido número de personas, cada una desde su lugar de trabajo a través de la web. Esto implica un bajo número de peticiones al servidor y una amplia conexión (ancho de banda) entre la máquina del cliente y el servidor.

Addons de Vaadin empleados

Como se ha comentado anteriormente, el entorno de trabajo de Vaadin permite la inclusión de *addons* diseñados por la comunidad de desarrolladores. Éstos simplifican ampliamente el trabajo de la aplicación final y ayudan a Vaadin a dar un aspecto más pulido.

A continuación se indican los *addons* usados en este proyecto y su utilidad:

- **TableExport-1.2.9.jar:** librería encargada de gestionar la exportación de tablas a hojas de cálculo.
- **vaadin-chamalon-theme-1.1.0.jar:** librería que contiene los estilos *CSS* aplicados a la página.

[3] http://vaadin.com/learn/

6. Implementación de SANT

El desarrollo de SANT se ha realizado utilizando el IDE Eclipse, y el entorno de desarrollo Vaadin. Se ha desarrollado enteramente en Java utilizando en la medida de lo posible las mejores prácticas y patrones de diseño software [25].

Esquema de ejecución de Vaadin

Este es el esquema de la arquitectura general de Vaadin, en la figura siguiente se puede observar cómo se relacionan las distintas partes del *framework* con nuestra aplicación y con la base de datos donde se incluye el servidor terminológico de BITAC.

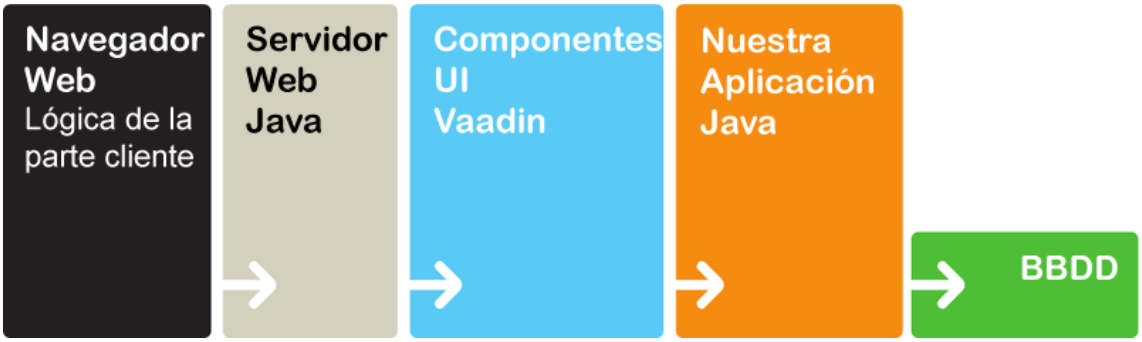

Figura 3: Arquitectura general de Vaadin

La interacción con el usuario por parte de nuestra aplicación se ejecutará en el navegador del cliente que se comunica con el servidor web (en nuestro caso Apache Tomcat 7) y con el resto de partes de la arquitectura que ya se ejecutan en la parte servidor. Las comunicaciones entre el cliente y el servidor son completamente transparentes, lo que simplifica enormemente el desarrollo.

Esquema general de SANT

A continuación se muestra el esquema de funcionamiento de nuestra herramienta de normalización terminológica que lo que busca principalmente es facilitar dicha tarea al técnico especialista.

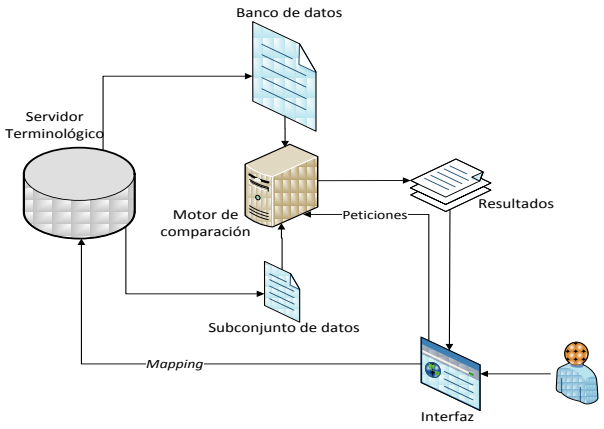

Figura 4: Esquema general de la aplicación desarrollada

Como se puede observar en la figura 4 SANT se integra en el servidor terminológico. Esta integración facilita la comunicación de retroalimentación que resulta muy importante para el crecimiento y mejora del servidor terminológico. Al mismo tiempo ahorra mucho tiempo al usuario que minimiza los cambios de contexto (cambios entre distinto *software* para la realización de tareas distintas) y los procesos de carga, modificación y guardado de datos.

SANT proporciona una interfaz de usuario a modo de aplicación web que permite el trabajo simultáneo de varios especialistas. Además el acceso se puede realizar desde multitud de dispositivos y utilizando sistemas distintos.

Uso de la aplicación

En el servidor terminológico existen pruebas de laboratorio ya enlazadas con la terminología de referencia (banco de datos en la figura 4) y otras que aún no lo están. Con estas últimas se pueden crear conjuntos de pruebas que podrán ser analizados por nuestra aplicación. Estos conjuntos de pruebas se denominan lotes (subconjunto de datos en la figura 4).

El técnico especialista puede consultar los lotes creados, su metainformación, y realizar tareas de preprocesado (descritas en la siguiente sección) sobre el lote antes de ser enviado al motor de comparación. En general, el preprocesado implica actuar sobre todas las pruebas de laboratorio dentro de un lote al mismo tiempo.

Una vez se han ejecutado todas las acciones de preproceso deseadas, el usuario puede ya mandar el lote de pruebas para su procesamiento por el motor de comparación. Una vez procesado todo el lote, aparecerá disponible para su validación.

El proceso de validación de resultados consiste en verificar que los enlaces propuestos por la herramienta de normalización son correctos. Se muestra una tabla con el conjunto de pruebas del lote. Cada una de estas pruebas no estandarizadas mostrará una prueba de la terminología de referencia asociada, es decir, el término LOINC con el que podría ser *mapeado*.

Si esta prueba candidata mostrada es correcta se puede marcar como “Validada” y continuar validando con el resto de resultados. Posteriormente, las pruebas marcadas se podrán incorporar como nuevos enlaces o *mappings* al servidor terminológico de manera que puedan formar parte de futuras búsquedas.

Si la prueba candidata mostrada no ofrece un *mapeo* correcto para nuestra prueba local es posible examinar el conjunto completo de resultados que se despliegan al seleccionar la prueba. De entre el conjunto de resultados se seleccionará la prueba correcta para que pase a ser su nueva candidata, y del mismo modo que se ha descrito anteriormente, poder ser marcada como “Validada”.

En caso de no encontrar candidato correcto entre los resultados el usuario tiene varias opciones, puede volver a lanzar el motor de comparación con una configuración menos restrictiva que devuelva un mayor número de candidatos, puede marcar una incidencia mediante un desplegable incorporado especialmente para ese caso o introducir un código de enlace manual (código Bitac).

Además, cada usuario puede trabajar por separado en lotes de pruebas distintos de manera que un usuario puede mandar a procesar un lote de pruebas al motor de comparación y otro usuario puede validar los resultados de un lote ya procesado, haya sido enviado por él o no, permitiendo paralelizar el proceso de normalización.

Mejora en el proceso de búsqueda terminológica

La aplicación proporciona unos resultados en función de la puntuación de similitud que proporciona la técnica de *string matching*, y el orden en que salen las pruebas dependerá en primera instancia de dicha puntuación.

Sin embargo se han implementado una serie de mejoras que depura este conjunto de resultados haciendo que mejore la eficacia del proceso:

El umbral y número máximo de resultados

Antes de enviar el lote de pruebas a procesar por el motor, el usuario puede modificar un umbral de similitud a partir del cual se aceptarán resultados. Es decir, si se establece un umbral del 30% sólo las pruebas que tengan una similitud mayor a este umbral se incluirán en el conjunto de resultados.

De igual manera se puede indicar un número *n* máximo de resultados a mostrar, de modo que al seleccionar una de las pruebas no estandarizadas en el proceso de validación, se muestren hasta *n* resultados.

Esta mejora hace que el técnico especialista pueda descartar automáticamente el grueso de pruebas con escaso o ningún parecido, que han dado un resultado en el motor de comparación muy bajo, y limita el número de resultados que tendrá que revisar en última instancia, permitiendo al usuario ahorrar tiempo y centrar su atención en las pruebas más relevantes.

Filtrado por ejes

El conjunto de resultados a mostrar se puede restringir según los ejes de la prueba procesada. Tal y como se explicó anteriormente, LOINC proporciona 6 ejes de información. El motor de comparación actúa sobre el eje descripción, sin embargo, quedan 5 ejes más: propiedad (directamente relacionada con la unidad), temporalidad, muestra, tipo y método.

Si la prueba analizada tiene alguno de dichos ejes informados, se puede filtrar el conjunto de pruebas del banco de datos y excluir así las pruebas que no resulten relevantes.

Por ejemplo, si la prueba local "Calcio - orina" debería mostrar sólo aquellas pruebas cuya muestra sea "orina", de este modo se limitaría de manera importante el número de pruebas a analizar. En el ejemplo siguiente se puede ver más claramente cómo actúa el filtrado por eje, filtrando aquellas pruebas irrelevantes:

Prueba analizada	Eje temporalidad	Eje muestra	Eje tipo
Calcio 24 h	**24 H**	**Orina**	**Qn**
Calcio en orina		Orina	
~~Calcio índice~~	-	~~Sangre~~	-
~~Calcio~~	-	~~Suero~~	~~Número~~
Cociente calcio/creatina		Orina	
~~Calcio en orina~~	-	~~Orina~~	~~Número~~
~~Calcio iónico~~	-	~~Sangre~~	~~Qn~~
Calcio/creatina	24H	Orina	Qn

Tabla 5: Ejemplo de funcionamiento del filtrado por ejes

Conseguimos así descartar aquellos *mappings* incorrectos por tener información distinta en los ejes, de esta forma mejoramos la eficiencia del motor de comparación ya que tendrá que hacer un menor número de comparaciones y mostramos sólo resultados relevantes al usuario.

Filtrado por ejes postproceso

Aun habiendo reducido considerablemente la lista de pruebas, es posible que la lista de resultados sea amplia, y el técnico especialista conozca más información por contexto de la que proporcionaba la prueba de laboratorio no estandarizada. Para esos casos la aplicación incorpora unos filtros postproceso que permiten filtrar la lista de resultados de acuerdo con una búsqueda por correspondencia directa de cadenas.

De esta manera se facilita la tarea de revisión de resultados, haciéndola más fácil y rápida.

Tablas de sinónimos

La información de un eje debe pertenecer al conjunto de opciones de la terminología de referencia, sin embargo, del mismo modo que ocurre con el nombre del concepto, también existe variabilidad en la forma de expresarlo debido a posibles abreviaturas, traducciones, nombres alternativos, errores tipográficos, etc.

Para que el filtrado sea efectivo, dos conceptos deben ser comparables. Para ello es necesario mantener unas tablas de sinónimos que enlacen las variaciones con su término (o incluso varios términos en algunas ocasiones) estándar. Por ejemplo, en el caso de la muestra "sangre" el término estándar en la terminología LOINC es "Bld", también lo es para "Sang", "blood", "sangre-jeringa"...

Un ejemplo del contenido de estas tablas de sinónimos es:

Muestra	Método
BLD	*Automated count*
Sangre	Contadores electrónicos
Sang	Contador celular automático
Sang TOTAL	Fluorescencia
Sang edta	Flujo y reconocimiento de imagen
SANGRE EDTA	Automatizado
Blood	Citometría de flujo
Sangre-jeringa	Citometría
…	…

Tabla 6: Ejemplo de tablas de sinónimos

La plataforma proporciona una herramienta de mantenimiento que muestra aquellos términos por eje que aún no han sido relacionados con un término estándar, y el especialista puede agregar la relación o bien editar cualquiera ya existente.

De este modo no es necesario que los ejes de las pruebas analizados estén informados en su manera estándar, con cualquier sinónimo incluido en las tablas es suficiente.

Obtención de ejes

En algunos casos la información de algunos ejes no se proporciona en el eje adecuado y, en su lugar, se encuentra junto con la descripción del concepto. Para estos casos se ha desarrollado una herramienta de preprocesado del lote que consiste en extraer dicha información de la descripción e incluirla en el eje adecuado. En todo caso el especialista puede descartar los cambios o incluso refinar el resultado mediante la edición del eje afectado.

Descripción	Prop	Tiempo	Muestra	Tipo	Método
Calcio en **orina**			***orina***		
Complemento C7 en **suero**			***suero***		
Calcio		24H		Qn	
Aminoácidos en **orina**			***orina***		
Aminoácidos **suero/plasma**			***suero/plasma***		
Hepatitis A IgG (**EIA**)				Número	***EIA***
Toxoplasma en **líquido amniótico PCR**			***líquido amniótico***		***PCR***
Albúmina en **suero**			***suero***	Qn	Nefelometría

Tabla 7: Ejemplo de algunos ejes extraídos a partir de la descripción

De esta forma ahora es posible refinar los resultados ya que podría aplicarse el filtrado por ejes descrito anteriormente.

Valores por omisión

En otros casos la información de los ejes simplemente no ha sido proporcionada, sin embargo el especialista puede conocerla gracias a su experiencia y conocimiento del contexto. Para estos casos SANT proporciona una herramienta mediante la cual es posible aplicar un valor por omisión a un eje seleccionado de entre el conjunto de términos del vocabulario estándar.

Dado el ejemplo anterior, si queremos que las pruebas para las que no se ha informado el eje muestra tomen un valor por omisión podemos seleccionarlo y el resultado sería como a continuación se indica:

Descripción	Prop	Tiempo	Muestra	Tipo	Método
Calcio en orina			*orina*		
Complemento C7 en suero			*suero*		
Calcio		24H	**orina**	Qn	
Aminoácidos en orina			*orina*		
Aminoácidos suero/plasma			*suero/plasma*		
Hepatitis A IgG (EIA)			**orina**	Número	*EIA*
Toxoplasma en líquido amniótico PCR			*líquido amniótico*		*PCR*
Albúmina en suero			*suero*	Qn	Nefelometría

Tabla 8: Ejemplo de aplicación del valor por omisión 'orina'

Del mismo modo que con la obtención de ejes, mejoramos el filtrado por eje, refinando el conjunto de resultados final.

Criterios de ordenación: método y ranking

Dentro del conjunto de resultados puede haber pruebas más relevantes que otras, incluso cuando han obtenido la misma puntuación de similitud. Para reflejar esto, se han incluido los siguientes criterios de ordenación:

- Por método: generalmente si el método ha sido especificado en la prueba de laboratorio local es porque es relevante y necesaria para diferenciar el término LOINC. El conjunto de resultados se ordenará por tanto siguiendo este criterio. Dada una misma similitud tendrá una mayor preferencia aquella prueba que tenga el eje método especificado.

- Por ranking: El ranking es un campo de LOINC, nuestra terminología de referencia. LOINC ha escogido las 2000 pruebas más comunes y las ha ordenado por frecuencia de uso. De modo que la prueba LOINC más utilizada es aquella con ranking igual a 1. Dada una misma similitud las pruebas del conjunto de resultados se ordenan, como segundo criterio, según el ranking.

Casos de uso

A continuación se describirán las principales acciones que puede realizar el usuario en nuestra aplicación, los cuales conforman los distintos casos de uso.

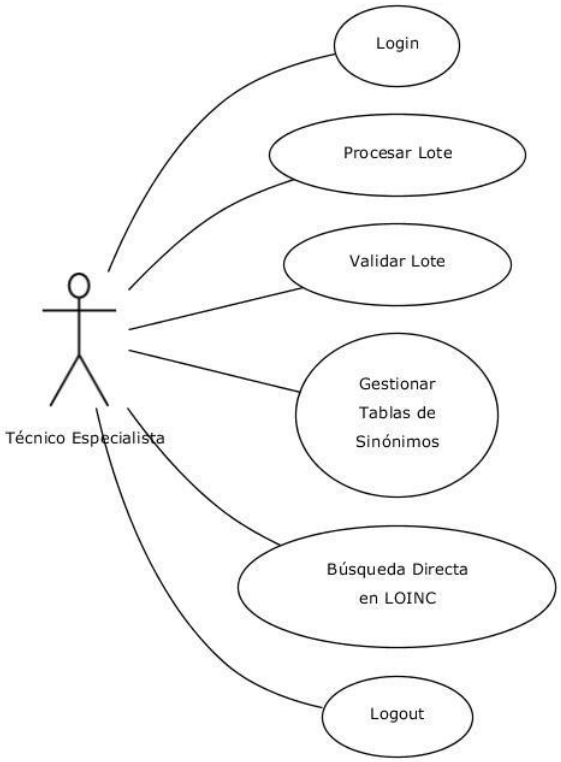

Figura 5: Diagrama de los Principales Casos de Uso

Login

La pantalla de *login* es la pantalla inicial. Se muestra los campos de textos para introducir usuario y contraseña y el botón "Entrar". Tras pulsar el botón "Entrar" el sistema comprueba los datos en la base de datos, si los datos son correctos la aplicación llevará a la pantalla de trabajo, si los datos son erróneos muestra un mensaje de error.

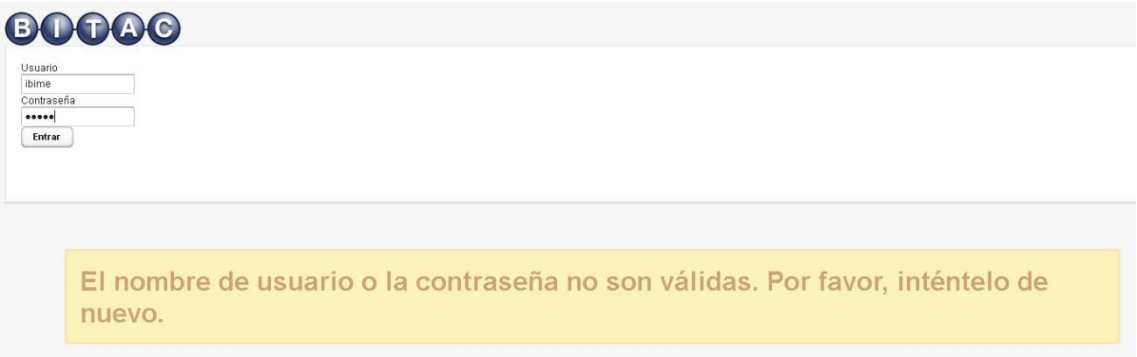

Figura 6: Pantalla inicial que muestra un error de *login*

Procesar lote

La pestaña Lotes permite por una parte realizar las tareas de preprocesado de las pruebas de laboratorio de los lotes antes de ser enviados al motor de comparación y por otra parte enviar a procesar el lote con una configuración determinada. A continuación se muestra en el diagrama de casos de uso las distintas acciones posibles:

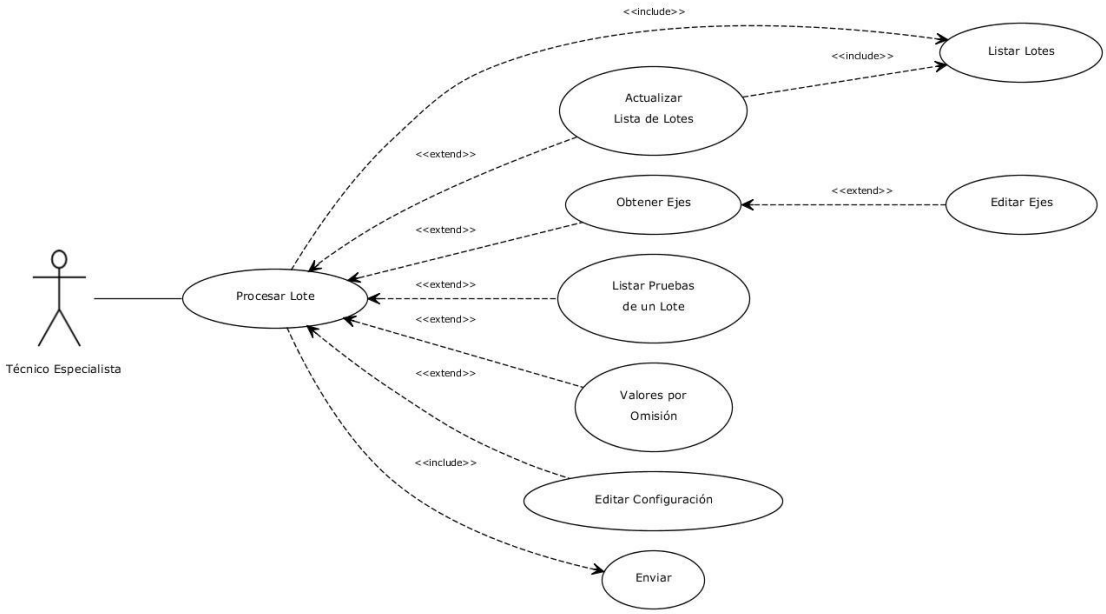

Figura 7: Caso de Uso "Procesar Lote"

Listar Lotes

Al entrar en la pestaña de Lotes se listan los lotes que el usuario ha preparado con antelación (creados desde una aplicación externa, pero gestionados por la nuestra), y que están ya disponibles en la base de datos.

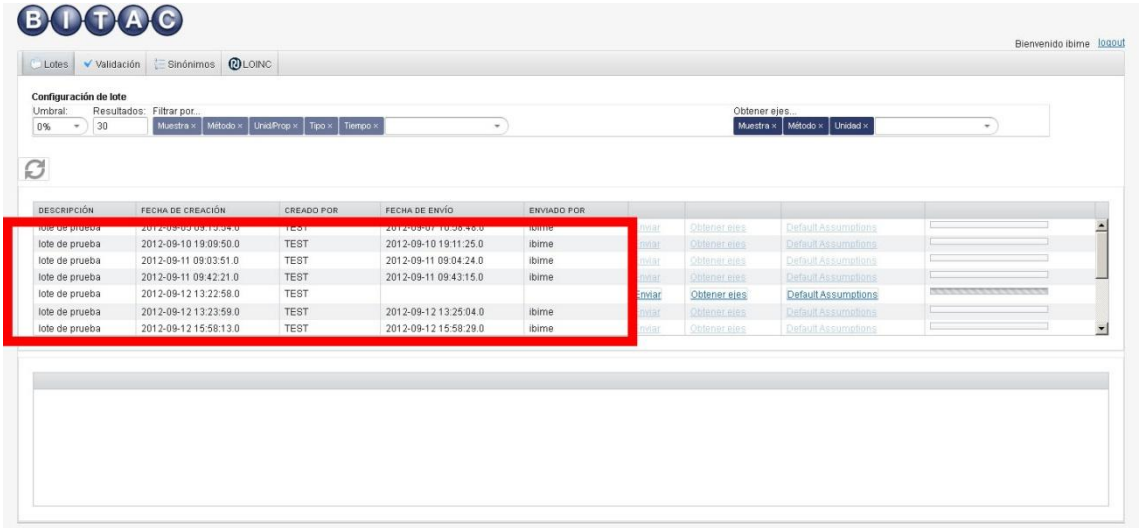

Figura 8: Lista de lotes

Actualizar Lista de Lotes

Si otro lote es creado mientras el usuario está en la pestaña de Lotes, es posible refrescar la vista de lotes mediante un botón.

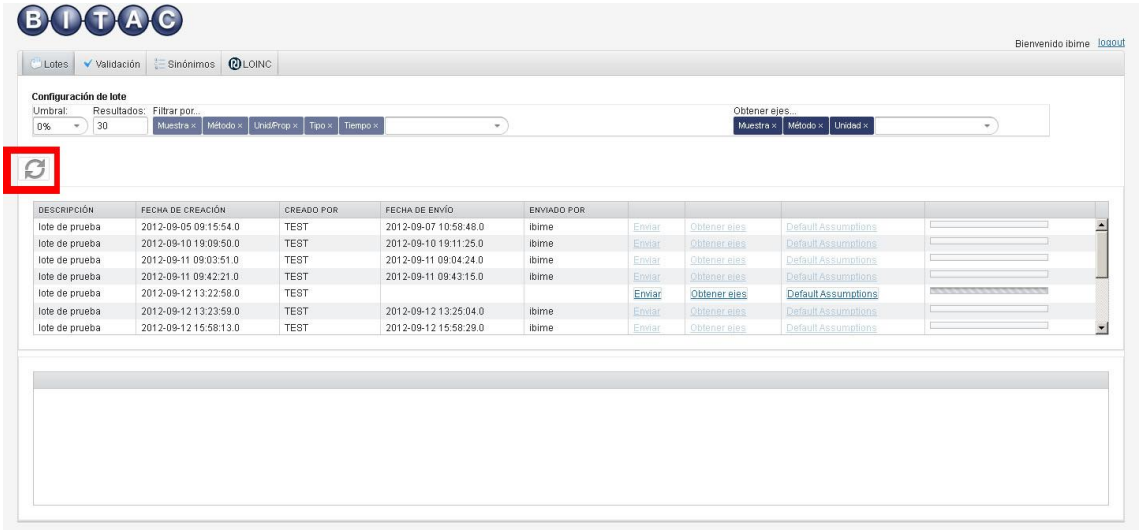

Figura 9: Botón para actualizar la lista de lotes

Listar Pruebas de un Lote

Seleccionando la fila correspondiente a un lote es posible listar todas las pruebas que lo componen.

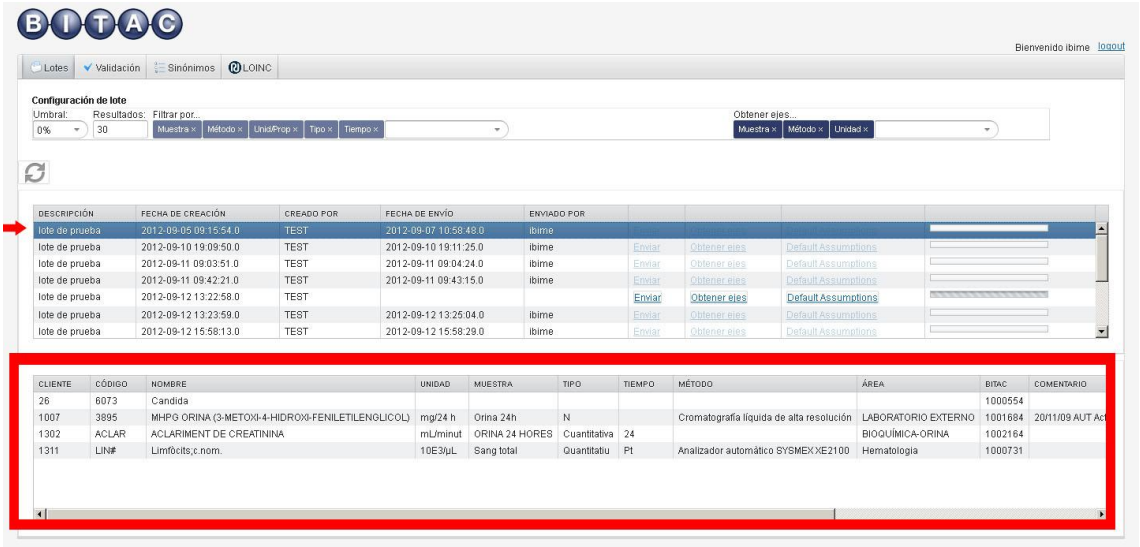

Figura 10: L lista de pruebas que contiene el lote seleccionado

Obtener Ejes

Una parte de preprocesado del lote es la obtención de ejes, esto es un proceso mediante el cual se buscan muestras, métodos y unidades válidas dentro de la descripción de cada una de las pruebas de laboratorio que componen el lote y se extraen al eje correspondiente para que pueda ser aplicado el filtrado correctamente. Para ejecutar la herramienta se dispone de un botón "Obtener Ejes" por cada uno de los lotes listados.

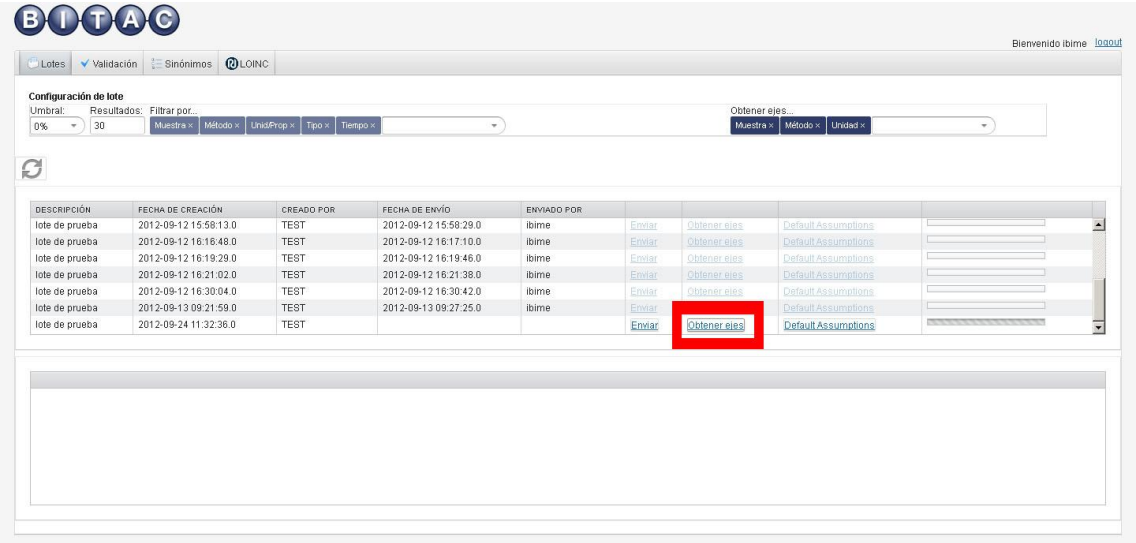

Figura 11: Botón para ejecutar la función de obtención de ejes

Editar Ejes

Es posible editar los datos extraídos de la descripción si el usuario considera que no son correctos. Pulsando el botón "Editar Ejes" se accede a una ventana de edición de los ejes que han sido modificados en el proceso de obtención de ejes.

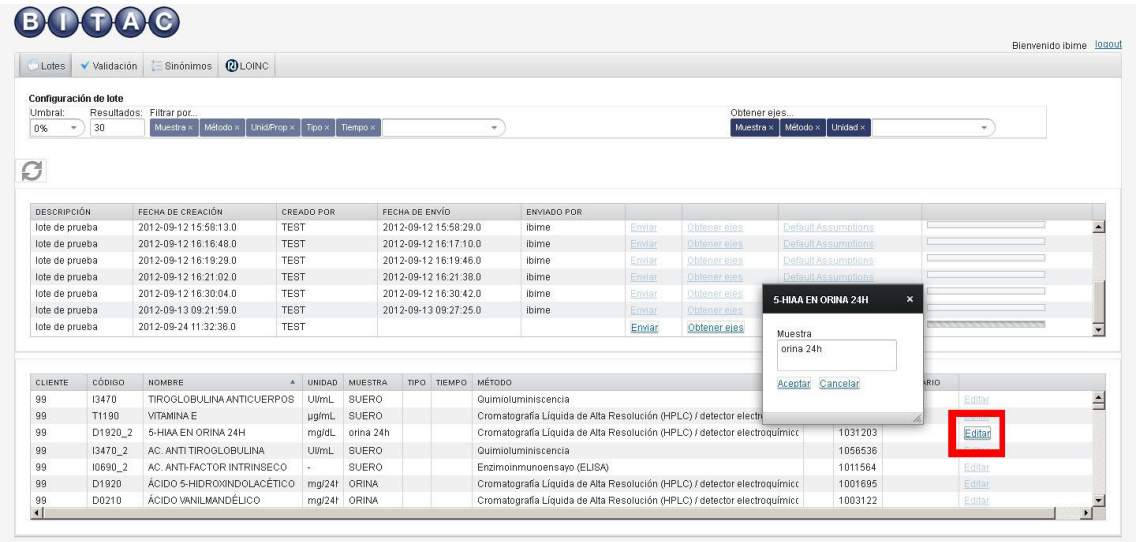

Figura 12: Ventana de edición del eje muestra obtenido a partir de la descripción

Valores por Omisión

Otra herramienta para el preprocesado del lote que puede utilizar el usuario es la asignación de valores por omisión (*default assumptions*) a ejes no informados en la prueba de laboratorio evaluada. Al pulsar el botón "Default Assumptions" accedemos a una ventana donde se puede seleccionar para los ejes muestra, método y propiedad un término del estándar LOINC a aplicar durante la fase de filtrado. La selección se realiza mediante un desplegable, y se aplicará a aquellas pruebas del lote que tengan el campo correspondiente al eje seleccionado vacío.

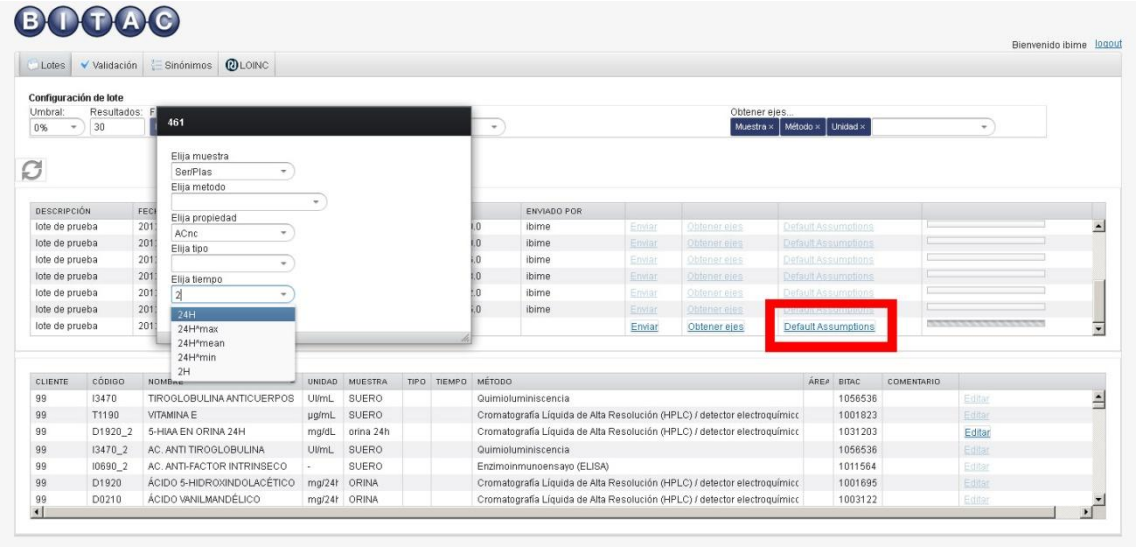

Figura 13: Ventana de *Default Assumptions*

Editar Configuración

Desde esta misma pestaña es posible cambiar la configuración de envío del lote. Una barra superior facilita esta tarea desde la misma pestaña que se mandan los lotes al motor de comparación. En esta barra se puede seleccionar mediante un desplegable el umbral de similitud a aplicar (por debajo del cual no se mostrarán resultados), el número máximo de resultados a mostrar y qué filtros aplicar durante el procesado del lote.

Además es posible aquí también seleccionar sobre qué ejes actuará la función de obtención de ejes anteriormente descrita.

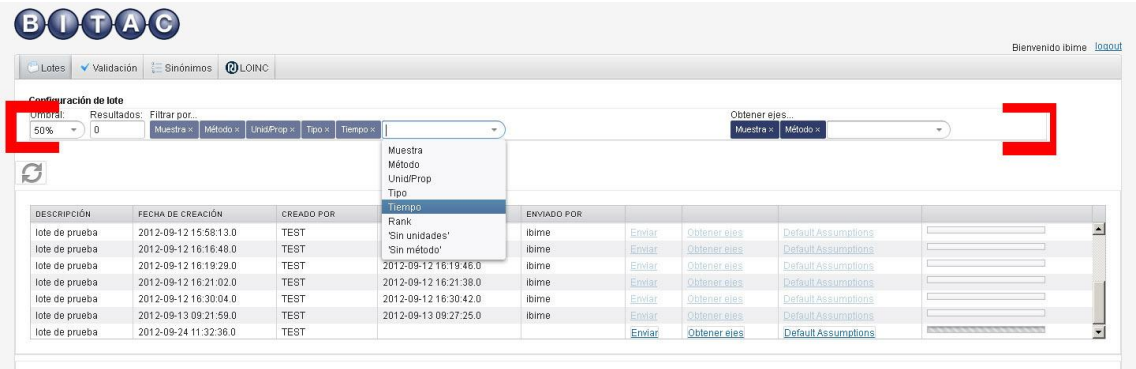

Figura 14: Barra de configuración

Enviar

El caso de uso principal en la pestaña de Lotes es la función de envío del mismo. Se dispone de un botón “Enviar” para cada lote. Además se puede observar el progreso de análisis del mismo mediante una barra de progreso habilitada en la misma línea.

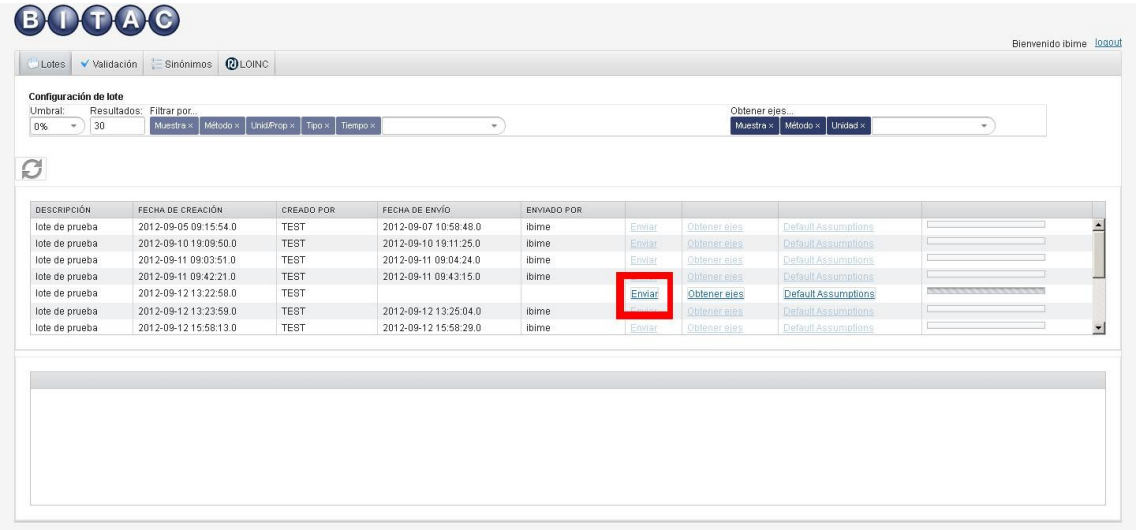

Figura 15: Botón para enviar el lote al motor de comparación

Validar lote

A continuación se describen los casos de uso correspondientes a la pestaña de validación. En este punto los lotes ya han sido analizados por el motor de comparación y por tanto se pueden revisar los resultados.

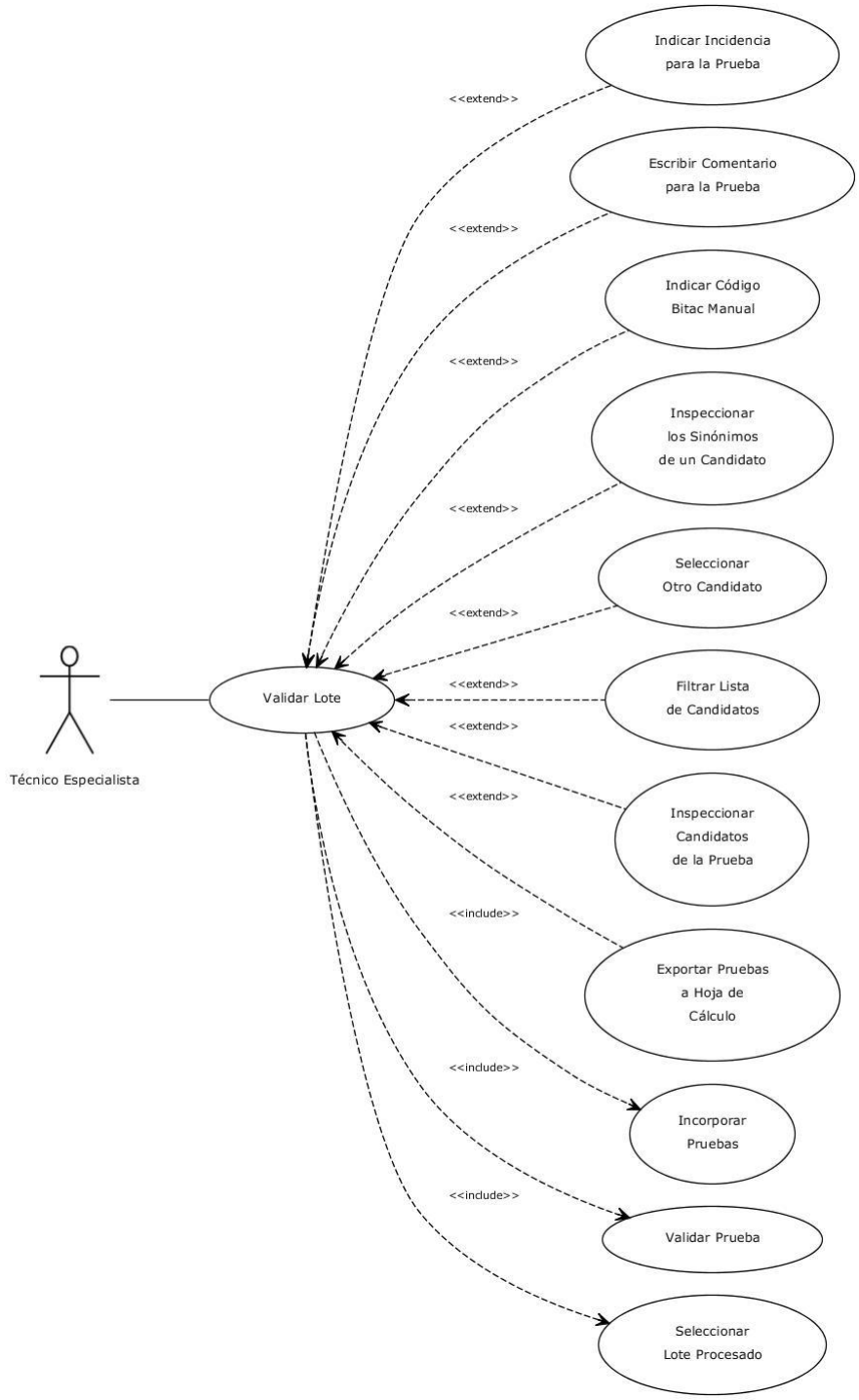

Figura 16: Caso de Uso "Validar Lote"

Seleccionar Lote Procesado

Una vez procesados los lotes, estarán disponibles para seleccionar en el desplegable que encabeza la sección. Además es posible actualizar la lista con el botón de refresco que aparece junto al desplegable.

Figura 17: Desplegable de selección de lote

Tras seleccionar un lote se despliegan las pruebas analizadas en una tabla como se muestra en la figura siguiente.

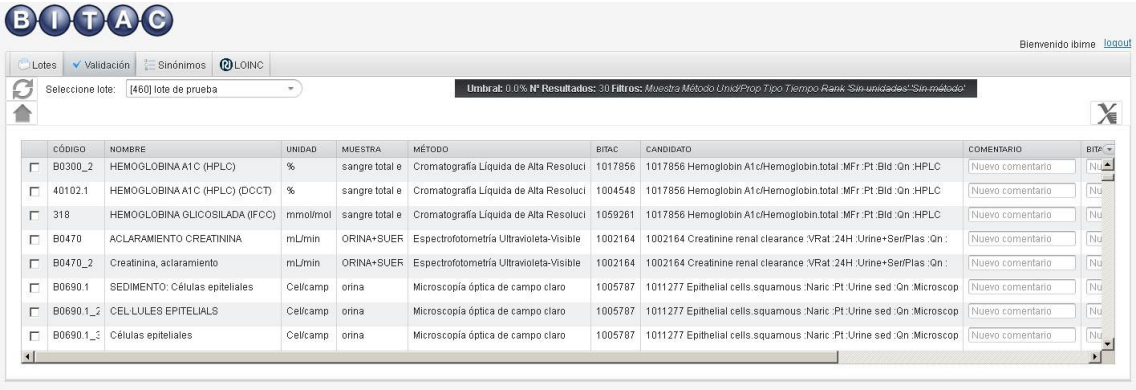

Figura 18: Lista de pruebas asociadas a su candidato del lote seleccionado

Aparece una barra superior que ofrece información de qué procesos de filtrado se han llevado a cabo en el motor de comparación. Se muestran también las distintas pruebas del lote ya asociadas a una prueba candidata.

Inspeccionar Candidatos de la Prueba y Seleccionar Otro Candidato

Al seleccionar una prueba de las desplegadas se muestra una tabla inferior con todos los resultados para dicha prueba. Es posible seleccionar un candidato de éstos mediante el doble *click* para que se actualice el campo superior de la prueba con el nuevo candidato seleccionado.

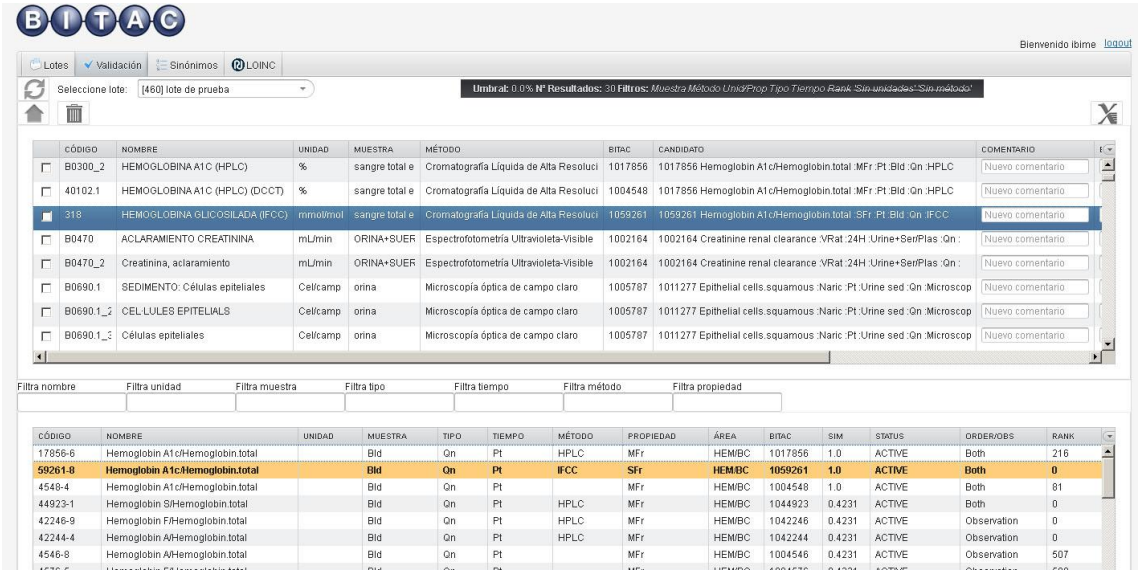

Figura 19: Lista de candidatos desplegada para la prueba seleccionada

Filtrar Lista de Candidatos

Es posible que la lista de candidatos sea extensa si se han establecido unos parámetros de configuración poco restrictivos. Se han habilitado unos cuadros de texto para buscar dentro de la lista de resultados. Se trata de unos filtros de contenido basados en búsqueda exacta de cadenas.

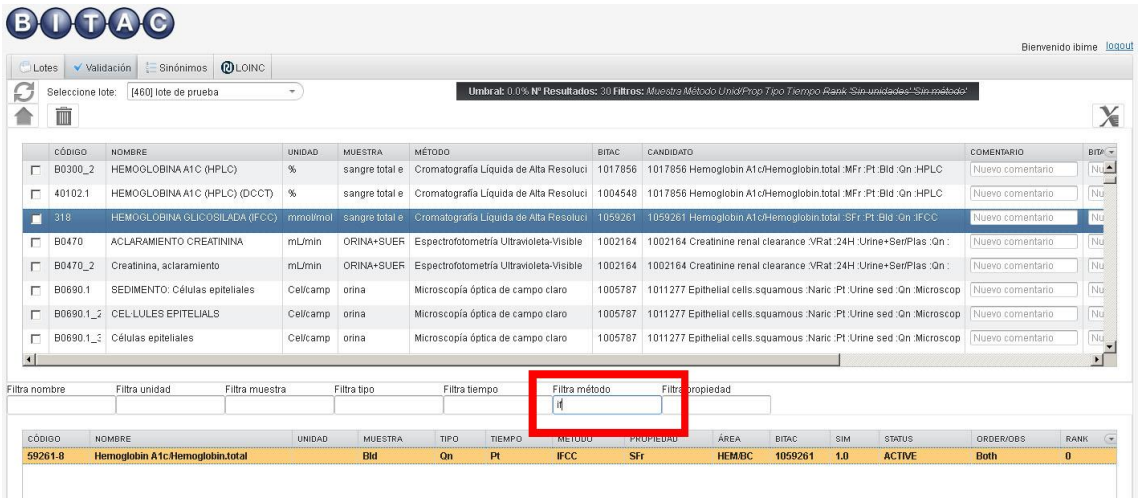

Figura 20: Ejemplo de uso del filtro postproceso para la columna método

Inspeccionar los Sinónimos de un Candidato

Mediante el uso del segundo botón del ratón sobre la prueba candidata deseada desplegamos una nueva ventana que muestra todas las pruebas del servidor terminológico que están relacionadas con ella mediante el código Bitac.

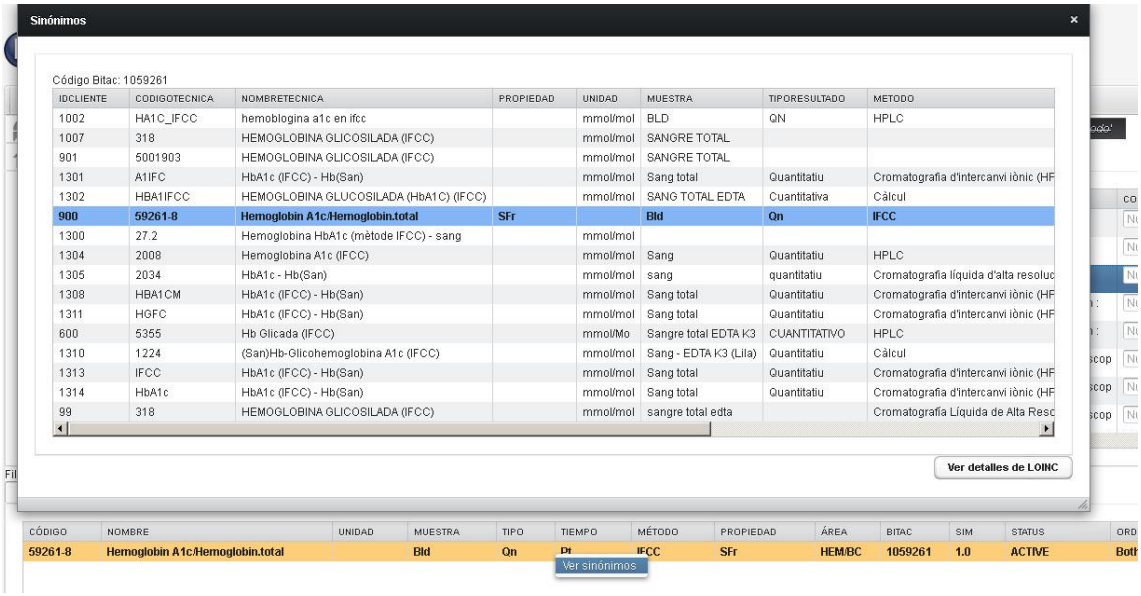

Figura 21: Ventana de sinónimos de una prueba candidata

Escribir Comentario para la Prueba, Indicar Código Bitac Manual e Indicar Incidencia para la Prueba

Tanto si el técnico especialista desea agregar algún comentario a la prueba analizada sobre el *mapping* realizado, como si no desea utilizar ninguno de los resultados y proponer un código Bitac manualmente, o si se trata de un caso especial y debe especificar una incidencia para la prueba, se dispone de dos cuadros de texto y un desplegable para dichas tareas:

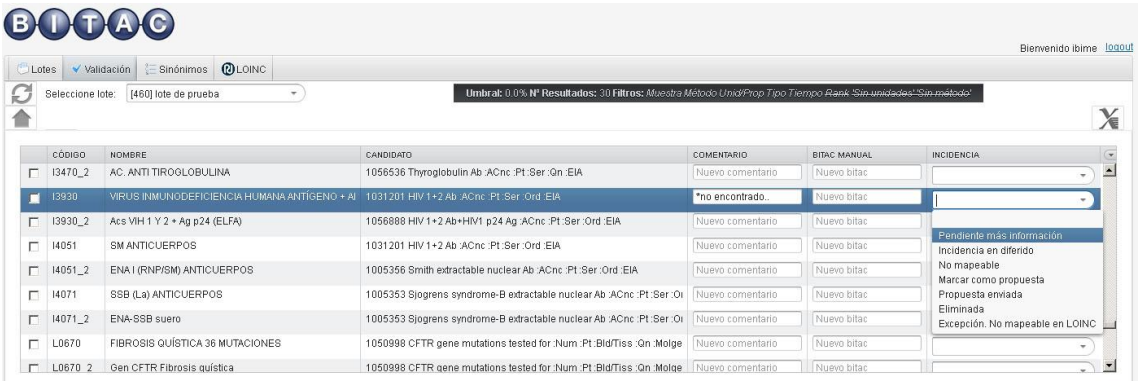

Figura 22: Ejemplo de uso del campo comentario y el desplegable de incidencias

Validar Prueba

Una vez el usuario ha comprobado que el candidato para la prueba es correcto puede marcarla como validada mediante un *checkbox* habilitado en el margen izquierdo de la tabla.

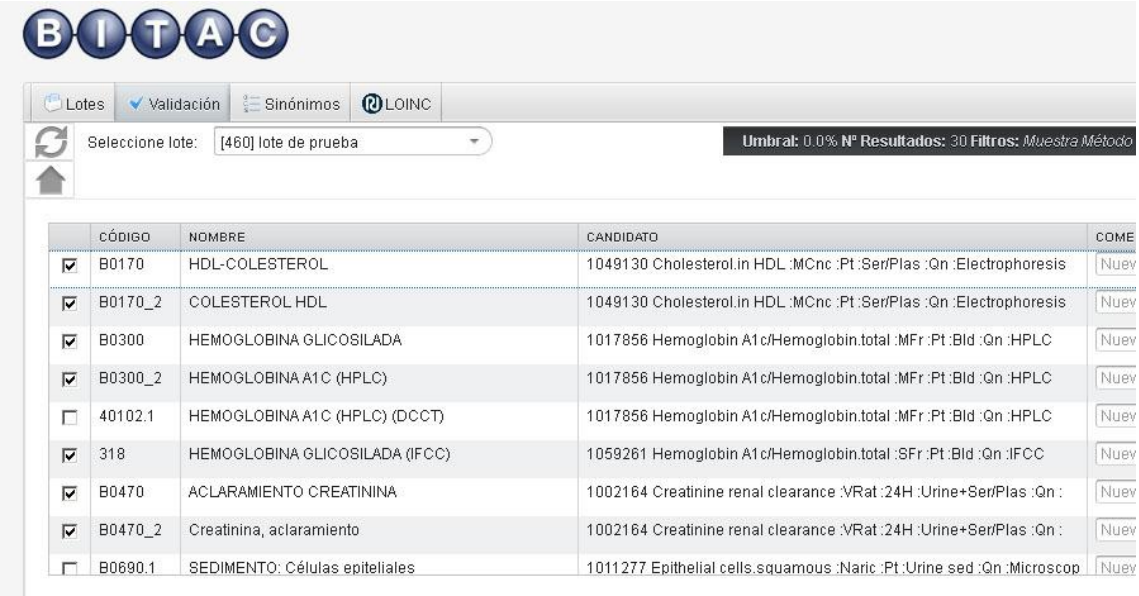

Figura 23: Conjunto de pruebas marcadas como validadas

Exportar Pruebas a Hoja de Cálculo

Una funcionalidad especial que se incluye en la aplicación es la opción de exportar la tabla de pruebas con sus candidatos a una hoja de cálculo, de forma que se pueda revisar, enviar, compartir con otros usuarios en cualquier momento sin necesitar acceder a la aplicación.

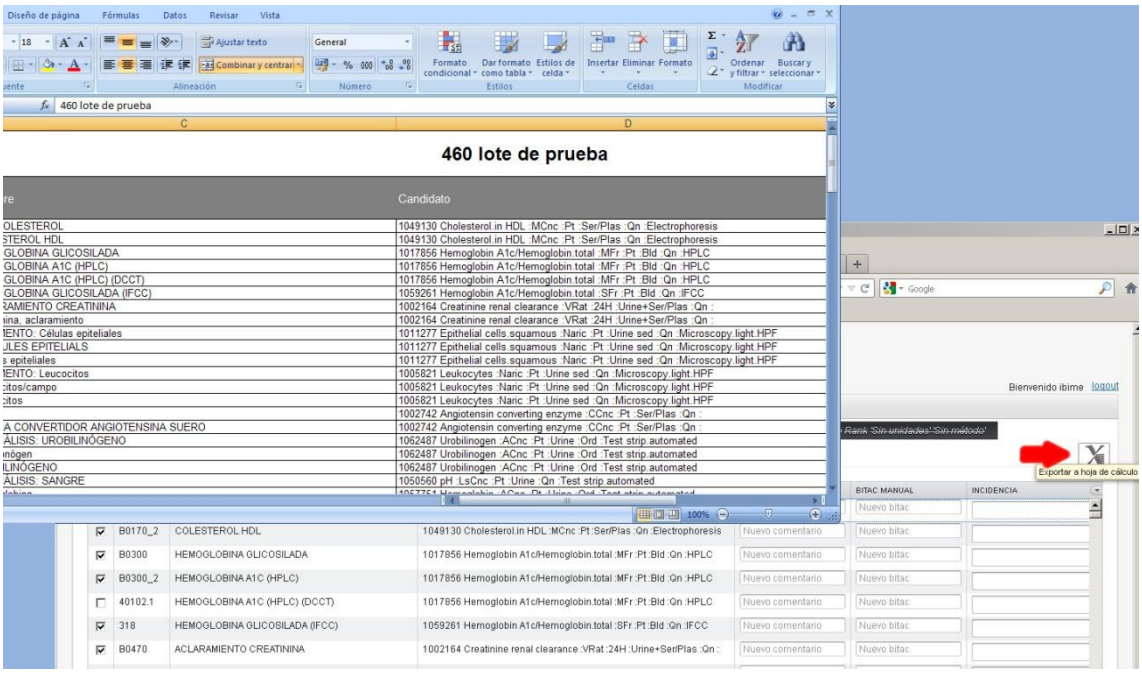

Figura 24: Exportación de resultados a una hoja de cálculo

Incorporar Pruebas

Una vez comprobadas y marcadas las pruebas por el técnico especialista ya están listas para incorporarse al servidor terminológico. Pulsando un botón sobre la tabla de pruebas es posible realizar la incorporación de todas las pruebas marcadas.

Figura 25: Botón para incorporar las pruebas marcadas al servidor terminológico

Gestionar tablas de sinónimos

La tercera pestaña de la aplicación se denomina “Sinónimos” y hace referencia a la gestión de las tablas de sinónimos para los ejes. Estos son los distintos casos de uso que encontramos en esta sección:

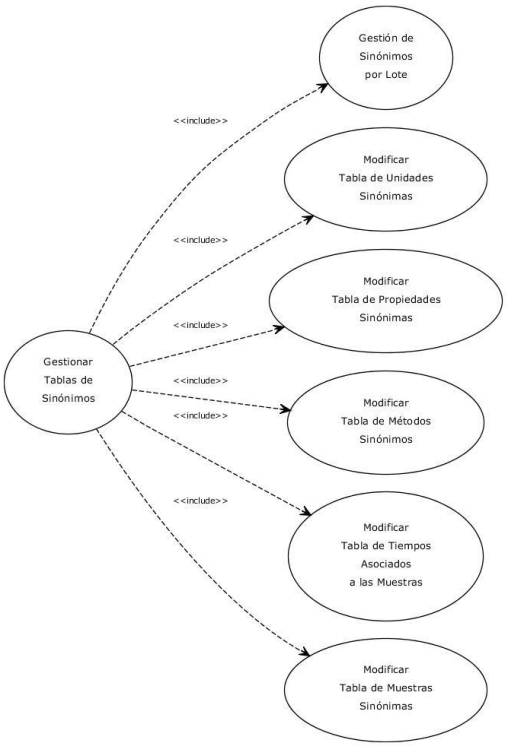

Figura 26: Caso de Uso "Gestionar Tablas de Sinónimos"

Para acceder a cada una de estas funcionalidades disponemos de un panel de botones que ser pulsados harán abrir la ventana de gestión correspondiente.

Figura 27: Pestaña de sinónimos

Modificar Tablas de Muestras, Métodos, Propiedades y Unidades Sinónimas

En todos los casos es posible modificar las tablas de sinónimos de cada eje, y el proceso es muy similar. Podemos acceder a la ventana habilitada para esta función a través del botón correspondiente "Sinónimos". Se abre una ventana donde podremos elegir el término (muestra, método o propiedad) estándar. Al seleccionar el término se despliega una tabla de opciones disponibles a la izquierda con un campo de texto para filtrar los resultados y a la derecha la lista de sinónimos existentes para el término seleccionado.

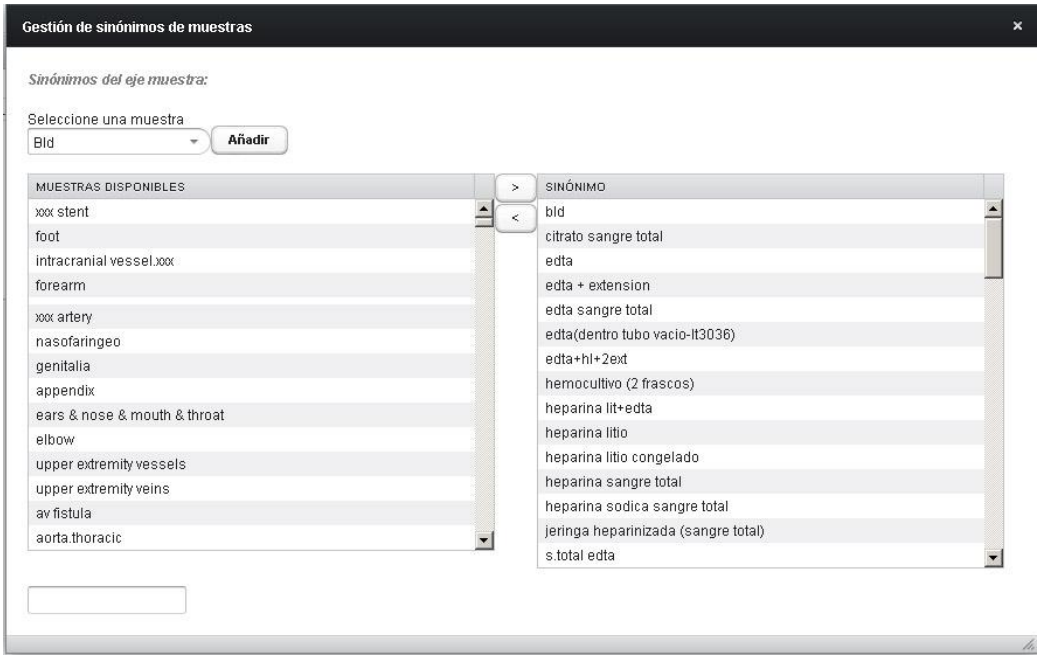

Figura 28: Ventana de gestión de sinónimos para el eje muestra

Modificar Tabla de Tiempos Asociados a la Muestras

Una característica común de las pruebas sin normalizar es que en un único eje de la prueba no estandarizada puede contener información de varios ejes estándar. Esto ocurre con mucha frecuencia en el eje muestra, donde se suele informar también el eje tiempo (o temporalidad).

Para poder aplicar los filtros adecuadamente se han creado unas tablas que dividen la información de la muestra proporcionada por la prueba no estandarizada. Para gestionar la tabla se dispone de la siguiente funcionalidad:

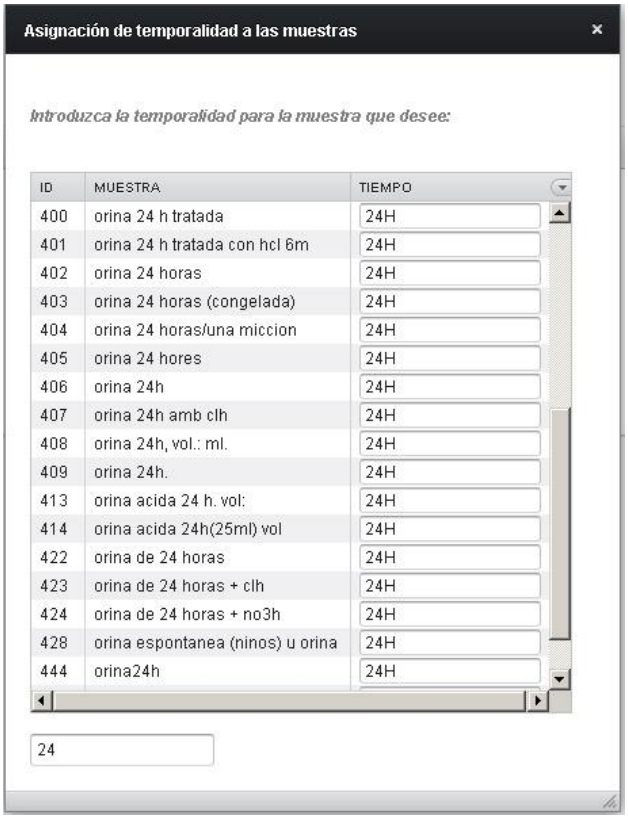

Figura 29: Ventana de gestión de la tabla de tiempos asociados a muestras

Gestión de Sinónimos por Lote

Mediante la función de gestión de sinónimos por lote podemos asignar nuevos sinónimos a los ejes de LOINC a partir de nuevas muestras, métodos o unidades que aparezcan en las pruebas de un lote.

Pulsando el botón correspondiente se abre una nueva ventana para la gestión de estos sinónimos. Aparece un desplegable donde podemos seleccionar sobre qué eje actuar, posteriormente podremos seleccionar el lote y el término estándar al que queremos asociar un nuevo sinónimo de los listados en la tabla izquierda. En la tabla de la derecha se muestran los sinónimos del término seleccionado.

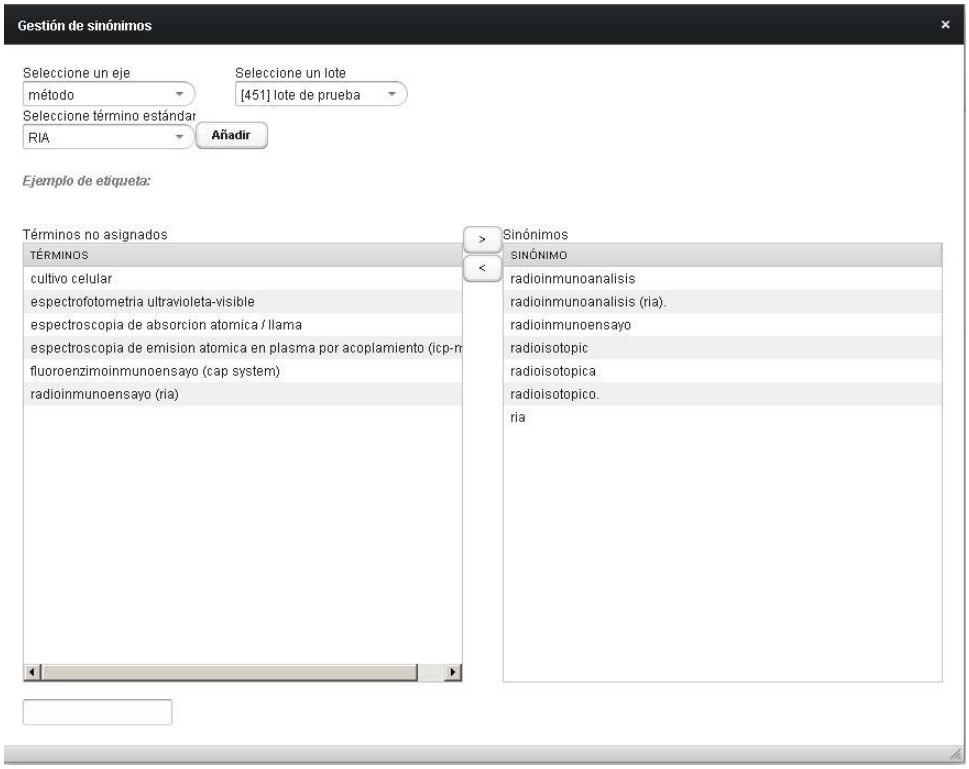

Figura 30: Ventana de gestión de sinónimos por lote

Búsqueda directa en LOINC

La pestaña LOINC permite acceder a una parte de la página web externa http://loinc.org que ofrece una herramienta de búsqueda por palabras de términos LOINC. Se trata de una función de apoyo y consulta para el técnico.

Al pulsar la pestaña LOINC se carga en el marco principal de la aplicación la página de búsqueda mencionada.

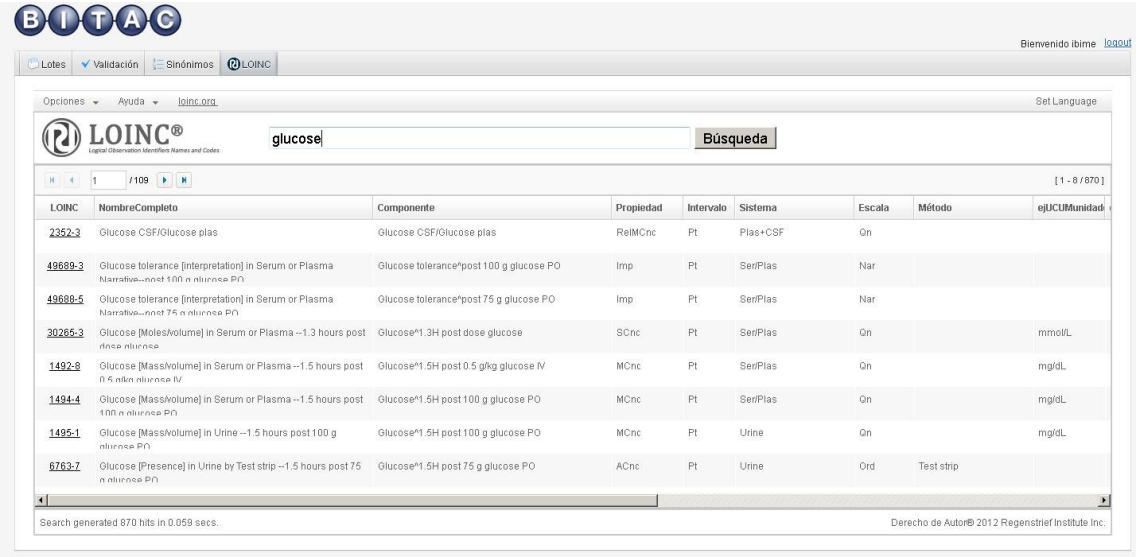

Figura 31: Pestaña LOINC para la búsqueda externa de términos

Logout

Sobre la barra de pestañas, en el margen derecho, se muestra el mensaje de bienvenida al usuario y el botón para cerrar sesión. Pulsando el botón *logout* la aplicación vuelve a la pantalla inicial.

Figura 32: Botón *logout*

7. Resultados

Para evaluar los resultados del proyecto se muestra el ciclo del proceso de análisis y validación completo para un lote de pruebas. Este lote de pruebas ha sido compuesto por la empresa BITAC especialmente para reflejar la casuística del problema.

El lote se compone de 157 pruebas reales pertenecientes a distintas áreas de laboratorio para las cuales ya se conoce su código Bitac correcto. Lo que se pretende es, ignorando el código Bitac ya existente, buscar *mappings* con el resto de pruebas del servidor terminológico y verificar si se encuentra o no el resultado correcto.

El primer paso la obtención de ejes no informados a partir de la descripción, para ello ajustamos la configuración, indicando que intente extraer la muestra, el método y la unidad, y pulsamos el botón "Obtener Ejes". Posteriormente revisamos los resultados:

Descripción	Unidades	Muestra	Método
ENZIMA CONVERTIDOR ANGIOTENSINA **SUERO**	U/L	**suero**	Espectrofotometría Ultravioleta-Visible
GASTRINA EN **SUERO**	pg/mL	**suero**	Radioinmunoensayo (RIA)
CALCITONINA HUMANA / **SUERO**	pg/mL	**suero**	Radioinmunoensayo (RIA)
5-HIAA EN **ORINA 24H**	mg/dL	**orina 24h**	Cromatografía Líquida de Alta Resolución (HPLC) / detector electroquímico
ANTICUERPOS IgM ANTI EPSTEIN-BARR [EARLY] EN **SUERO**	TITULO	**suero**	Inmunofluorescencia indirecta.
Acs VIH **1** Y 2 + Ag p24 (ELFA)	**1**	SUERO	Quimioluminiscencia
ZINC EN **SERUM**	µg/dL	**serum**	Espectroscopía de Absorción Atómica / Llama
COBRE EN **SUERO**	µg/dL	**suero**	Espectroscopía de emisión atómica en plasma por acoplamiento (ICP-MS)

Tabla 9: Pruebas que incluyen información de alguno de sus ejes en su descripción

Como se puede observar en la tabla 9, se ha extraído información a partir de la descripción para 8 pruebas. En un caso se ha realizado incorrectamente, es el caso de la prueba "Acs VIH 1 Y 2 + Ag p24 (ELFA)". En este caso se ha extraído incorrectamente el "1" en el eje unidad. Es posible corregir este error mediante la herramienta de edición:

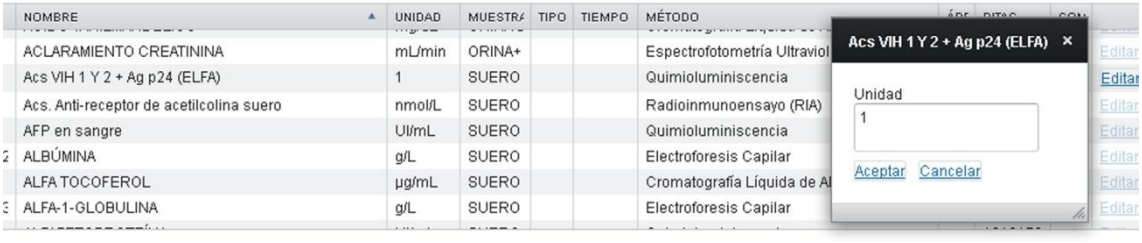

Figura 33: Ventana de edición de ejes obtenidos

Una vez corregido el error, se comprueba si es necesario aplicar algún *Default Assumption,* recordemos que al seleccionar un valor por omisión, éste se aplicará sobre todos los campos del eje al que corresponda que se encuentren vacíos, por ejemplo, al seleccionar "Ser/Plas" en nuestro lote, se aplicaría a todas aquellas pruebas que posean el campo muestra vacío. Sin embargo, en nuestro caso no es posible aplicar un valor por omisión debido a la heterogeneidad del lote.

Llegados a este punto ya se puede enviar el lote al motor de comparación. Ajustaremos la configuración de envío para que utilice los filtros de muestra, método y unidad(o propiedad). Indicaremos un número máximo de resultados de 10, para que los resultados sean comparables con los experimentos realizados para la técnica de *string matching* utilizada (Jaccard).

Figura 34: Configuración aplicada

Una vez completado el análisis del motor de comparación se puede proceder a la validación de las pruebas.

En la pestaña correspondiente seleccionamos nuestro lote y procedemos a validar los resultados, marcando aquellas pruebas para las que hemos encontrado al candidato correcto.

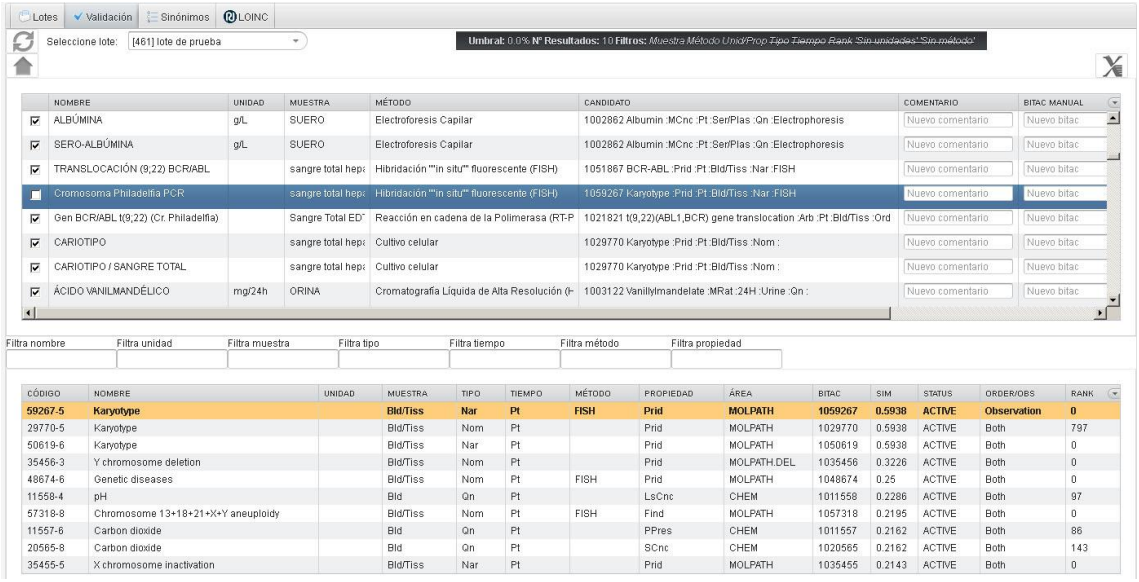

Figura 35: Marcado de pruebas que han encontrado un candidato correcto

De las 157 pruebas del lote se han conseguido validar 146. Esto supone haber encontrado candidato para el 93% de las pruebas. Si comparamos los resultados con los que obteníamos utilizando la técnica Jaccard sin aplicar ninguna de las mejoras que aporta la aplicación observamos una mejora considerable que supone elevar la tasa de acierto en un 30%.

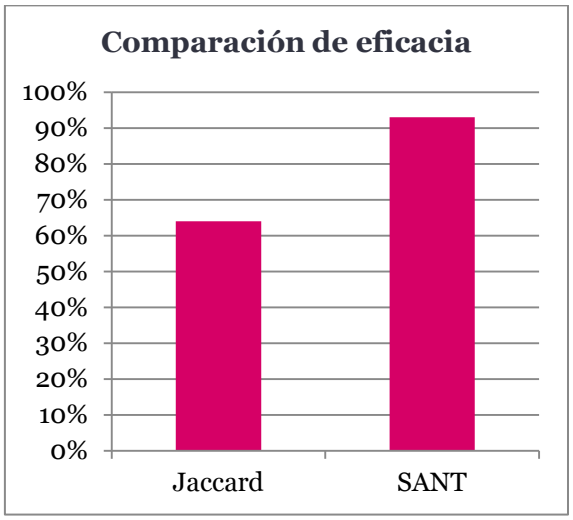

Figura 36: Gráfica comparativa de la eficacia en cada caso

En definitiva se ha conseguido desarrollar una plataforma de trabajo para la normalización terminológica con LOINC que se integra dentro del servidor terminológico del cliente de forma que se facilita la comunicación con él, tanto para la exportación de pruebas no estandarizadas como para la importación de resultados, y que automatiza el proceso de búsqueda de candidatos mediante un motor de comparación basado en técnicas de correspondencia aproximada de cadenas.

Gracias a que SANT se integra dentro del servidor terminológico y a su motor de comparación es posible realizar la búsqueda de términos estandarizados sin necesidad de preprocesado del término no estandarizado (lo que implicaría entre otras cosas la traducción del término al lenguaje de la terminología estándar, en este caso al inglés). De esta forma se consigue ahorrar una cantidad de tiempo muy importante al usuario ya que se reduce sustancialmente la carga de trabajo manual.

Por otra parte, el usuario también ve facilitada su labor de búsqueda del candidato correcto mediante las herramientas proporcionadas por SANT que reúne los resultados en una sola ventana, y proporciona una interfaz sencilla y práctica.

Una vez el especialista encuentra el *mapping* apropiado, es posible incorporar este resultado al servidor terminológico para que pase a formar parte de futuras búsquedas. Esto en definitiva es un proceso de realimentación de la base de conocimiento.

LOINC es un lenguaje en pleno desarrollo, y por tanto es posible encontrar pruebas de laboratorio que no han sido incorporadas aún a la terminología. Para facilitar este proceso SANT ofrece una herramienta que facilita que estas pruebas se marquen de una manera especial en el servidor terminológico para su posterior envío al organismo responsable del mantenimiento de LOINC.

8. Conclusiones

Una vez analizados los inconvenientes actuales de la normalización terminológica con LOINC se plantea la implementación de SANT, un sistema de ayuda a la normalización terminológica integrado en el servidor terminológico de BITAC.

Para implementar el motor de comparación de nuestro sistema se han analizado distintas técnicas de correspondencia aproximada de cadenas. Tras la realización de experimentos para comprobar la eficacia de cada una de las técnicas consideradas se escoge Jaccard por resultar la más efectiva y ser además de complejidad baja lo que repercute en un mayor rendimiento del motor de comparación.

Para la implementación de la interfaz gráfica de usuario se han comparado distintos *frameworks* de desarrollo web, decantándonos por Vaadin, un *framework* de última generación que ofrece gran cantidad de documentación, contenido pregenerado en forma de componentes y *addons*, y un rendimiento óptimo.

Hemos aportado mejoras a nuestro sistema para incrementar la eficacia del motor de comparación en forma herramientas de preprocesado de pruebas (obtención de ejes a partir de la descripción, aplicación de valores por omisión), uso de filtros por ejes de LOINC durante el proceso de análisis dentro del motor de comparación y herramientas de búsqueda en el proceso de validación para facilitar el trabajo del usuario.

Se han demostrado los beneficios de SANT como sistema de ayuda a la normalización terminológica, ahorrando una sustancial carga de trabajo al técnico especialista y facilitando las tareas de creación de *mappings* y propuesta de incorporación de nuevos términos LOINC.

9. Trabajo futuro

Actualmente se está trabajando en una nueva versión del sistema que incluye nuevas mejoras. Una de las nuevas técnicas consiste en mejorar la explotación de la información que nos proporciona el servidor terminológico mediante la extracción de, lo que hemos llamado, los sinónimos por componente. Mediante este proceso se consigue que pruebas ya estandarizadas que tenían pocos sinónimos y de bajo parecidos a la prueba analizada, ahora puedan mostrarse en los resultados, reduciendo así el número de falsos negativos.

Otra herramienta que se quiere incorporar próximamente es la validación automática de pruebas. Esto es, según unos criterios a determinar, se podrán validar de forma automática las pruebas analizadas, sin intervención del técnico especialista. A esta nueva versión del proyecto se le ha denominado SAM (*Automatic Mapping System*).

Como trabajo futuro adicional se plantea el uso del sistema desarrollado con otras terminologías clínicas como por ejemplo para SNOMED-CT, terminología clínica de gran aceptación dentro de la comunidad experta en normalización e interoperabilidad semántica que actualmente se está implantando en gran parte de los centros hospitalarios a nivel internacional y recientemente adoptada por el sistema nacional de salud.

10. Agradecimientos

Me gustaría que estas líneas sirvieran para expresar mi más profundo y sincero agradecimiento a todas aquellas personas que con su ayuda han colaborado en la realización del presente trabajo, en especial a la Dra. Montserrat Robles Viejo por darme la oportunidad de trabajar en algo que me apasiona, a mi codirector el Dr. Jose Alberto Maldonado Segura por la orientación, el seguimiento y la supervisión continua del proyecto, a ambos por la motivación y el esfuerzo por que este proyecto llegara a buen término.

Expresar mi profundo agradecimiento a BITAC, por darme la oportunidad de trabajar en este proyecto tan interesante y por dedicarle un tiempo precioso, Toni, Mireia, Vicky, por empujarme a mejorar más y más.

Gracias a todos mis compañeros en IBIME por toda la ayuda prestada, por el buen ambiente de trabajo que generan y por enriquecer en general mi vida profesional y personal.

Un agradecimiento muy especial merece la comprensión, paciencia y el ánimo recibidos de mi familia y amigos.

A todos ellos, muchas gracias.

11. Bibliografía

1. Cimino JJ. Saying what you mean and meaning what you say: coupling biomedical terminology and knowledge. Academic medicine : journal of the Association of American Medical Colleges. 1993 Apr;68(4):257-260.

2. Rector AL. Clinical terminology: Why is it so hard. In: Methods of Information in Medicine; 1999. p. 239-252.

3. Chute CG. Clinical Classification and Terminology. Journal of the American Medical Informatics Association. 2000 May;7(3):298-303.

4. Forrey AW, McDonald CJ, DeMoor G, Huff SM, Leavelle D, Leland D, et al. Logical observation identifier names and codes (LOINC) database: a public use set of codes and names for electronic reporting of clinical laboratory test results. Clinical Chemistry. 1996 Jan;42(1):81-90.

5. Committee on Data Standards for Patient Safety. "Front Matter." Patient Safety: Achieving a New Standard for Care. Washington, DC: The National Academies Press, 2004.

6. Sellers PH. On the theory and computation of evolutionary distances. SIAM Journal on Applied Mathematics. 1974 Junio; 26(4): p. 787-793.

7. Levenshtein VI. Binary codes capable of correcting spurious insertions and deletions of ones. Problems of information transmission. 1965; 1(1): p. 8.

8. Masui T. An efficient text input method for pen-based computers. In Co. APWP, editor. Proceedings of the SIGCHI conference on Human factors in computing systems; 1998; New York. p. 328-335.

9. Lasko TA, Hauser SE. Approximate string matching algorithms for limited-vocabulary OCR output correction. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series. 2000 December; 4307: p. 232-240.

10. Angell RC, Freund GE, Willet P. Automatic spelling correction using a trigram similarity measure. Information Processing & Management. 1983; 19(4): p.255-261.

11. Rahm E, Do HH. Data Cleaning: Problems and Current Approaches. In: IEEE Data Engineering Bulletin. vol. 23; 2000.

12. Kreuzthaler M, Bloice MD, Faulstich L, Simonic KM, Holzinger A. A Comparison of Different Retrieval Strategies Working on Medical Free Texts. Journal of Universal Computer Science. 2011 Apr;17(7).

13. Vreeman DJ, McDonald CJ. A comparison of Intelligent Mapper and Document Similarity Scores for Mapping Local Radiology Terms to LOINC. In AMIA Annual Symposium proceedings; 2006. p. 809-813.

14. Lau LM, Johnson K, Monson K, Lam SH, Huff SM. A method for the automated mapping of laboratory results to LOINC. In Proceedings / AMIA. Annual Symposium; 2000. p. 472-476.

15. Chandel A, Hassanzadeh O, Koudas N, Sadoghi M, Srivastava D. Benchmarking declarative approximate selection predicates. In Proceedings of the 2007 ACM SIGMOD international conference on Management of data; 2007; New York. p. 353-364.

16. Ukkonen E. Approximate string-matching with q-grams and maximal matches. Theoretical Computer Science. 1992 January ; 92(1): p. 191-211.

17. Sarawagi S, Kirpal A. Efficient set joins on similarity predicates. In Proceedings of the 2004 ACM SIGMOD international conference on Management of data; 2004; New York. p. 743-754.

18. Salton G, Wong A, Yang CS. A vector space model for automatic indexing. In Commun. ACM.; 1975; New York. p. 613-620.

19. Gusfield D. Algorithms on Strings, Trees, and sequences: Computer Science and Computational Biology. Cambridge: Cambridge University Press; 1997.

20. Monge A, Elkan C. The field matching problem: Algorithms and applications. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining; 1996. p. 267-270.

21. Cohen WW, Ravikumar P, Fienberg SE. A comparison of string distance metrics for name-matching tasks. In IJCAI-03 Workshop on Information Integration; 2003. p. 73-78.

22. Manning CD, Raghavan P, Schütze H. Introduction to Information Retrieval. Cambridge: Cambridge University Press; 2008.

23. Marco Ruiz L. Automatización del proceso de estandarización de pruebas clínicas de laboratorio mediante técnicas de record linkage. 2012. Tesis Final de Máster en Ingeniería de Análisis de Datos, Mejora de Procesos y Toma de Decisiones. DEIOAC. Universitat Politècnica de València.

24. Grönroos M. Book of Vaadin: 4th Edition. Turku: Vaadin Ltd; 2012.

25. Gamma E, Helm R, Johnson R, Vlissides JM. Design Patterns: Elements of Reusable Object-Oriented Software: Addison-Wesley Professional; 1994.

12. Anexo: Artículo Inforsalud'12

NORMALIZACIÓN TERMINOLÓGICA MEDIANTE TÉCNICAS DE BÚSQUEDA APROXIMADA

E. PARCERO[1], L. MARCO[1], J.A. MALDONADO[1], V. BÉREZ[2], T. MAS[2], M. RODRÍGUEZ[2], M. ROBLES[1]

[1]*Grupo IBIME. Instituto ITACA. Universitat Politècnica de València. 46022-Valencia. España.*
[2]*BITAC MAP S.L. 08008-Barcelona. España.*

En este artículo se presenta la plataforma de normalización terminológica SANT (Sistema de Ayuda a la Normalización Terminológica). SANT utiliza técnicas de búsqueda aproximada de cadenas (string matching), filtros configurables y reglas semánticas para encontrar correspondencias entre los términos locales no estandarizados y los términos contenidos en terminologías estándar o de referencia como LOINC. SANT se ha integrado con el sistema de información de gestión terminológica de la empresa BITAC con el propósito de facilitar la normalización de vocabulario médico, su principal ámbito de trabajo. El presente artículo describe la metodología seguida, idoneidad de las técnicas de búsqueda aproximada y el sistema informático resultante de la colaboración entre BITAC y el grupo IBIME de la Universitat Politècnica de València.

1. Introducción y motivación

La necesidad de utilizar vocabularios normalizados es vital para la comunicación e integración de la información sanitaria. Sin embargo, los centros sanitarios utilizan generalmente una terminología y codificación propia, haciendo difícil la comunicación entre ellos. El uso de una terminología común permite el intercambio de datos clínicos entre sistemas independientes en pos de mejorar la atención al paciente y facilitar las actividades de investigación.

No obstante, la adopción de un sistema terminológico estándar no está exenta de dificultades. Además de existir múltiples sistemas terminológicos y de codificación propia, nos encontramos con la inmensidad y continua evolución del lenguaje clínico y las sutiles diferencias existentes entre los conceptos, aparentemente muy parecidos, pero esencialmente diferentes. Por tanto, la normalización de la terminología propietaria consume mucho tiempo y es un proceso muy especializado que requiere personal cualificado.

Por una parte, las herramientas de alineación existentes no están integradas en el sistema, de manera que se necesita un procesado manual previo de una batería de términos para su posterior análisis y búsqueda de posibles candidatos en la terminología objetivo. La búsqueda del candidato correcto es realizada por un especialista que evalúa la correspondencia entre términos basándose en la información que le ha proporcionado el centro de origen. Y, por último, la integración de los resultados necesitará un procesado posterior para su almacenamiento, generalmente también manual.

Por otra parte, la terminología objetivo puede que se encuentre en otro idioma que la local e incluso es posible que se disponga de términos de entrada en varios idiomas.

2. Objetivos

Describir la plataforma SANT, un sistema de normalización terminológica, que además de unificar los procesos de extracción, comparación e incorporación de términos del servidor de terminología, automatiza el proceso de comparación.

Reducir el tiempo que el especialista debe dedicar a la tarea de búsqueda del candidato mediante la selección y aplicación automática de reglas establecidas para la obtención de listas de términos ya normalizados, ordenados según un coeficiente de similitud, proporcionado por una técnica basada en la correspondencia por cadenas.

Realimentar la base de conocimiento de correspondencia semántica entre términos locales y estandarizados de manera que la aportación de los datos sea sencilla, asista en la elección del término candidato, y facilite la incorporación de los resultados.

Facilitar la propuesta de incorporación de nuevos términos que actualmente no tienen correspondencia en la terminología estándar.

3. Materiales y métodos

Como punto de partida se dispone de un banco de datos de términos de pruebas clínicas. Este banco de datos reúne términos de distintos centros codificados según su sistema local, contiene también una terminología de referencia y códigos que relacionan términos locales con los términos de dicha terminología de referencia.

Así pues, es posible agrupar los términos por código, obteniendo un conjunto de sinónimos para un concepto dado. Este grupo es interesante por la información que aporta ya que puede incluir distintas formas de referirse a un concepto, términos con mayor o menor información relevante, y equivalencias de conceptos entre los idiomas utilizados en el banco de datos.

En concreto, el banco de datos utilizado pertenece a la empresa BITAC, especialistas en normalización de datos de laboratorio, y hace uso de la terminología LOINC como terminología estándar. LOINC es un sistema de codificación de pruebas de laboratorio y otras observaciones clínicas en el que un término se representa fundamentalmente por un código único y 6 ejes que describen el término: nombre del componente medido, propiedad, tiempo de la medición, tipo de muestra, escala de medida y método utilizado si es relevante [1]. Además BITAC es miembro de la comunidad LOINC y colaboran en el desarrollo y traducción de su vocabulario.

Los términos de referencia del banco de datos se representan conforme al estándar definido por LOINC, y el resto de términos correspondientes a distintos centros se almacenan según su formato original, de manera que es habitual encontrar términos con ejes no informados, ejes con información incorrecta o ejes con información que corresponde a un eje, o incluso a varios ejes distintos.

Para calcular la similitud entre términos se hace uso de técnicas de búsqueda aproximada de cadenas, más comúnmente conocidas por su término en inglés *string matching* [2]. Consisten en aplicar una función de similitud a dos cadenas *simil(c1, c2)*, y obtener a partir de ella una puntuación. Si dicha puntuación supera un umbral μ, ajustado según conveniencia, entonces se habrá encontrado un término candidato a ser su equivalente.

Las funciones implementadas en el motor de comparación son las funciones de *string matching*, una clase de funciones de popularidad creciente, basada en la *tokenización* de las cadenas en *Q-grams*, es decir, en subcadenas de tamaño *Q* [3]. De esta forma es posible tratar las cadenas como conjuntos de *tokens* y usar las operaciones de teoría de conjuntos sobre los conjuntos de *tokens* para calcular la similitud entre cadenas. La técnica Jaccard[4] hace uso de esto. Otro tipo de funciones son las basadas en distancias, como las distancias de edición de Levenshtein [5]. Existen métodos que combinan estas dos ideas, denominados métodos híbridos, como Monge Elkan [6] y SoftTFIDF [7].

Jaccard se trata de una técnica sencilla y barata desde el punto de vista computacional. El coeficiente de similitud de Jaccard entre dos cadenas *c1* y *c2* es el cociente de la cantidad de *tokens* presentes tanto en *c1* como en *c2* entre la suma de ambas cantidades:

$$similJaccard(c1,c2) = \frac{|tokens(c1) \cap tokens(c2)|}{|tokens(c1) \cup tokens(c2)|}$$

Levenshtein consiste básicamente en calcular el coste de transformar una cadena en otra:

$$Dist_{Levenshtein}(c1, c2) = min(s_1, s_2, s_3 \ldots)$$

Donde s_1, s_2, s_3... definen el coste de secuencias de operaciones de edición tales como *copia*, *inserción*, *substitución* y *borrado* a realizar para convertir la cadena *c1* en *c2*. El resultado es peor cuanto mayor sea este coste, es decir, las cadenas son más diferentes.

En Monge Elkan el cálculo de la función de similitud se hace a partir del cálculo de las distancias mínimas de edición entre *tokens*, y normalizando con respecto al tamaño en *tokens* de la cadena origen:

$$similMongeElkan(c1,c2) = \frac{1}{K}\sum_{i=1}^{K} {}_{j=1}^{L}\max(simil'(c1_i, c2_j))$$

Donde *simil'* es una función secundaria basada en distancias, una variante de la métrica de Levenshtein, *K* es la cantidad de *tokens* de c1 y *L* es la cantidad de *tokens* de c2.

En SoftTFIDF el *token* se pondera según una función entre las frecuencias de aparición del *token* en la cadena a comparar y en el banco de datos general, de manera intuitiva cuanto más frecuente sea el *token* menos significativo resulta.

4. Resultados y discusión

Se ha desarrollado la plataforma SANT para facilitar la tarea de la normalización terminológica, de modo que, como muestra la figura 1, se integra dentro del sistema objetivo, enlazándose a su base de datos.

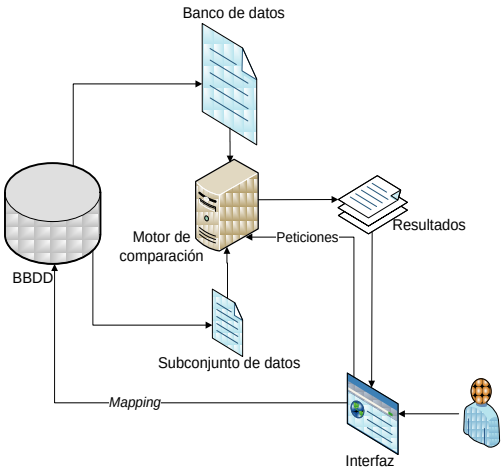

Figura 1: *Perspectiva general de funcionamiento de la plataforma SANT.*

SANT proporciona una interfaz web que permite el trabajo simultáneo de varios especialistas, permitiendo componer distintos subconjuntos de datos o lotes que serán procesados por el motor de comparación y realizar el tratamiento de los resultados arrojados por la plataforma. Además el acceso se puede realizar desde multitud de dispositivos y/o con sistemas distintos.

4.1. Funciones de string matching: comparación

SANT soporta cualquier método de comparación de cadenas que proporcione un resultado normalizado en forma de puntuación.

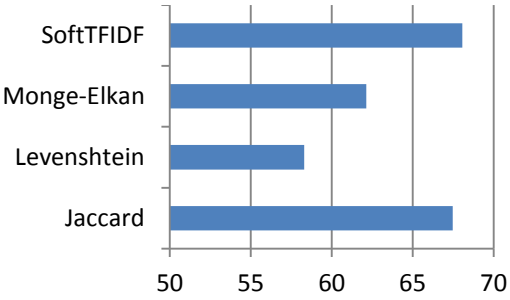

Figura 2: *Porcentaje de acierto por método de comparación por string matching considerado.*

En la plataforma han sido implementadas las técnicas basadas en *string matching* anteriormente descritas. Para su evaluación se ha utilizado un subconjunto de datos de 3.000 términos LOINC escogidos al azar de entre un banco de datos de aproximadamente 150.000 términos. Se ha realizado la comparación de todos estos términos, por el primer eje, nombre del componente, obteniendo el conjunto de los 10 mejores resultados para cada término del subconjunto. Se considera acierto en la comparación si entre estos 10 términos se encuentra el término correcto.

Como se puede observar en la figura 2, de entre todos los métodos de comparación considerados, aquellos que se comportan mejor son Jaccard (con una *tokenización* en *Q-grams* de dos caracteres) y SoftTFIDF (con una *tokenización* en *Q-grams* de tres caracteres), ambos métodos con resultados cercanos al 70% de acierto. Debido a la sencillez de Jaccard, se ha escogido éste para su implantación.

4.2. Mejoras en el proceso de búsqueda terminológica

La plataforma SANT admite la creación de conjuntos de conceptos para los cuales se desea establecer nuevas relaciones. El especialista se encarga de componer estos conjuntos o lotes utilizando las herramientas suministradas. A parte del nombre del componente, se incluyen el resto de ejes. Los ejes aportan una información muy valiosa a la hora de decidir con qué concepto se puede establecer o no una relación, y en base a esto se introducen en la plataforma las siguientes mejoras: (a) Filtrado por eje. (b) Mantenimiento de sinónimos. (c) Extracción de ejes a partir de la descripción. (d) Asunción de valores por omisión.

El filtrado por eje consiste en proporcionar candidatos que contengan una información semánticamente igual en los ejes seleccionados a filtrar. Esta selección de ejes será fija para todo un lote. Para cada concepto se consulta qué información contiene en cada uno de los ejes relevantes, y SANT busca en el banco de datos las posibles correspondencias que cumplan la restricción de poseer la misma información en sus propios ejes, eliminando así conceptos irrelevantes. Con esto se consigue mejorar la eficiencia temporal, porque el banco de datos con el que comparar se ve drásticamente reducido, a la vez que disminuye el conjunto de resultados que el usuario debe revisar.

La información de un eje debe pertenecer al conjunto de opciones de la terminología de referencia, sin embargo, del mismo modo que ocurre con el nombre del concepto, también existe variabilidad en la forma de expresarlo debido a posibles abreviaturas, traducciones, nombres alternativos, errores tipográficos, etc. Para que el filtrado sea efectivo, dos conceptos deben ser comparables. Para ello es necesario mantener unas tablas de sinónimos que enlacen las variaciones con su término (o incluso varios términos en algunas ocasiones) estándar. La plataforma proporciona una herramienta de mantenimiento que muestra aquellos términos por eje que aún no han sido relacionados con un término estándar, y el especialista puede agregar la relación o bien editar cualquiera ya existente.

En algunos casos la información de algunos ejes no se proporciona en el eje adecuado y, en su lugar, se encuentra junto con la descripción del concepto. Para estos casos SANT dispone de una herramienta de preprocesado del lote que consiste extraer dicha información de la descripción e incluirla en el eje adecuado. En todo caso el especialista puede descartar los cambios o incluso refinar el resultado mediante la edición del eje afectado.

En otros casos la información de los ejes simplemente no ha sido proporcionada, sin embargo el especialista puede conocerla gracias a su experiencia y conocimiento del contexto. Para estos casos SANT proporciona una herramienta que permite aplicar un valor por omisión a un eje, seleccionado de entre el conjunto de términos del vocabulario estándar. Otros parámetros de configuración que el usuario puede modificar son: el umbral de similitud a aplicar en la técnica de *string matching* y el número máximo de candidatos a mostrar.

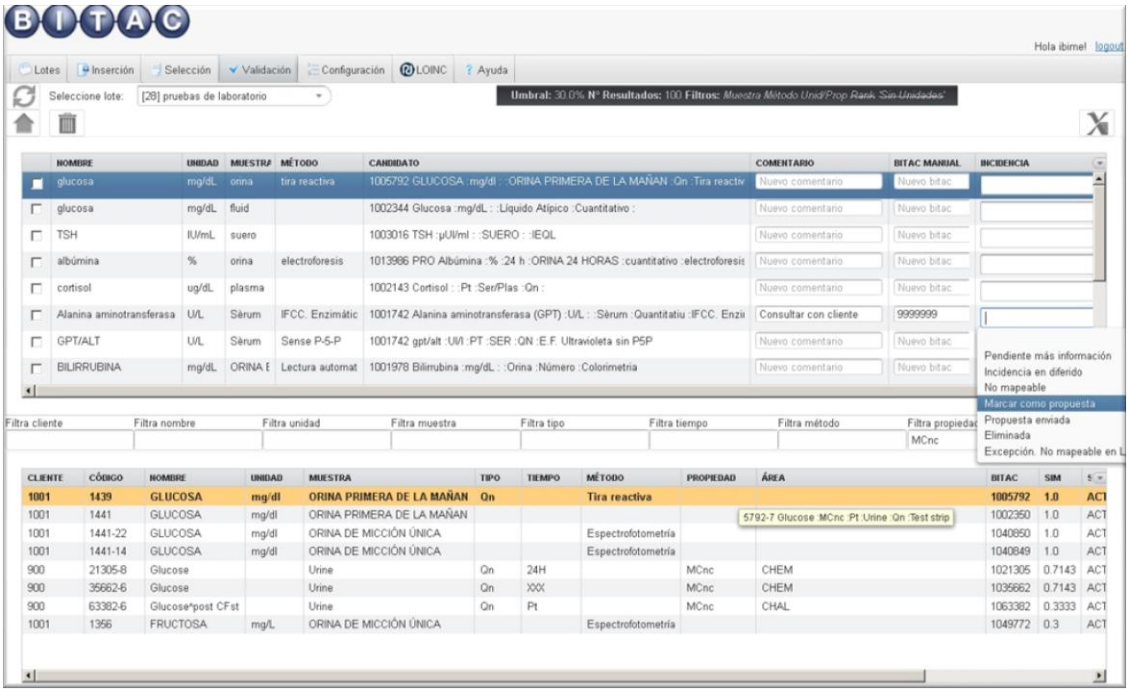

Figura 3: *Captura de pantalla de la plataforma SANT*

4.3. Tratamiento de los resultados

Una vez el motor de comparación ha volcado los resultados, el especialista comienza el proceso de comprobación (ver figura 3) que consiste en verificar si el candidato propuesto es acertado. Si el primer candidato no es correcto se puede seleccionar otro entre un conjunto de candidatos que se despliegan al seleccionar el concepto.

SANT proporciona una herramienta de búsqueda por cadenas entre los resultados en el caso de preferir la búsqueda directa a la navegación. Consiste en un filtro por caracteres, y se han incluido para los ejes más comúnmente utilizados.

En caso de encontrar un candidato correcto para el concepto éste puede ser seleccionado para posteriormente ser incorporado a la base de datos local con el código de enlace correspondiente que lo relaciona con la terminología de referencia.

En caso de no encontrar candidato correcto el usuario tiene varias opciones, puede volver a lanzar el motor de comparación con una configuración menos restrictiva que devuelva un mayor número de candidatos, puede marcar una incidencia mediante un

desplegable incorporado especialmente para ese caso o incluso introducir un código de enlace manual.

5. Conclusiones

Se ha presentado la plataforma SANT, se ha conseguido un entorno integrado y completo, independiente de herramientas externas para la importación de términos locales y exportación de términos ya normalizados. Como indican los datos, las técnicas de búsqueda aproximada consiguen un acierto en el 70% de los casos, esto, unido a las herramientas proporcionadas por la plataforma, facilita la tarea de la normalización terminológica, reduciendo la complejidad del proceso y el tiempo que un especialista debe emplear en él.

Referencias

[1] A. Forrey, C. McDonald, G. DeMoor, S. Huff, et al. Logical observation identifier names and codes (LOINC) database: a public use set of codes and names for electronic reporting of clinical laboratory test results. Clinical Chemistry, 42 (1), 81-90 (1996).

[2] A. Chandel, O. Hassanzadeh, N. Koudas, M. Sadoghi y D. Srivastava. Benchmarking declarative approximate selection predicates. SIGMOD '07. 353-364 (2007).

[3] E. Ukkonen. Approximate String Matching with q-Grams and Maximal Matches. Theorical Computer Science, 92 (1), 191-211 (1992).

[4] S. Sarawagi y A. Kirpal. Efficient set joins on similarity predicates. SIGMOD'04. 743-754 (2004).

[5] D. Gusfield. Algorithms on strings, trees, and sequences: computer science and computational biology. Cambridge University Press (1997).

[6] A. Monge y C. Elkan. The Field-Matching Problem: Algorithm and Applications. Proc. 2nd ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, AAAI Press. 267–270 (1996).

[7] W. W. Cohen, P. Ravikumar, and S. E. Fienberg. A comparison of string distance metrics for Name-Matching tasks. IJCAI-03 Workshop on Information Integration. 73-78 (2003)