Facultad de Ingeniería
C.P.A.P. - Instituto de Computación

Tesis de Maestría
en Ingeniería en Computación

"Definición de una Arquitectura de Referencia para Anonimizar Documentos"

Ing. Horacio Vico
Tutor: MSc. Ing. Daniel Calegari
2013.

Agradecimientos

A mi tutor Daniel Calegari, quien ha sido un verdadero guía en este trabajo, aportando siempre oportunas ideas y comentarios sin los cuales difícilmente habría arribado a buen puerto.

A mi esposa Evelyn por su apoyo incondicional en este proyecto personal que ha sido cursar una maestría.

A mi hijo Nahuel, quien ha compartido desde sus primeros meses de vida mi tiempo libre con el desarrollo de este trabajo de tesis.

Y muy especialmente a mi hija Belén que llegó a nuestras vidas cuando ya me acercaba al final de este trabajo.

Resumen

La anonimización es un proceso que permite identificar y ocultar la información sensible contenida en los documentos, permitiendo su divulgación sin que ello implique vulnerar los derechos a la protección de datos de las personas y organizaciones que se puedan referenciar en los mismos.

La anonimización automática o semi-automática de documentos no estructurados se constituye como un desafío importante desde el punto de vista de la ingeniería de software y en particular de la arquitectura de software ya que, entre otras cosas, el proceso que se lleva a cabo no se encuentra bien definido, y para su realización se deben combinar diversas disciplinas como el procesamiento de lenguaje natural y la minería de textos.

El presente trabajo de tesis introduce una arquitectura de software de referencia para la anonimización de documentos desestructurados, basada en propuestas arquitecturales existentes en la bibliografía. Se describe dicha arquitectura en detalle así como se estudia la disponibilidad de herramientas vinculadas al procesamiento del lenguaje natural, que resultan de utilidad en un proceso de anonimización. Finalmente se lleva a la práctica la arquitectura propuesta mediante el diseño e implementación de un prototipo de sistema de anonimización concreto para un marco de aplicación específico, consistente en la anonimización de sentencias judiciales (jurisprudencia).

Palabras clave: Anonimización, despersonalización, protección de datos personales, arquitectura de software, procesamiento del lenguaje natural.

Índice

Índice de figuras

Índice de tablas

1. Introducción

El incesante avance de las tecnologías de la información en el seno de las organizaciones, ha impulsado la incorporación de la Gestión Documental[45] como una disciplina fundamental. El objetivo es optimizar la gestión y así maximizar el aprovechamiento de los grandes volúmenes de información que se encuentran en la forma de documentos. En algunos dominios de aplicación de la gestión documental tales como el gobierno electrónico o los servicios de salud, entre otros, se presenta una necesidad recurrente: la anonimización. Este y otros conceptos se describen en profundidad en la siguiente sección y en el Anexo A, pero a modo introductorio diremos que anonimización es el proceso que consiste en proteger o incluso eliminar la información sensible contenida en los documentos.

La anonimización tiene aplicación en aquellos documentos donde la información de valor contenida en ellos, es independiente de los datos personales o la información sensible. El fin es que dicha información pueda ser utilizada dentro de la propia organización o por terceros, sin que esto implique vulnerar la privacidad y la confidencialidad de los datos personales de las personas físicas o jurídicas que se referencian en el documento original. Algunos países poseen legislación muy específica vinculada con la anonimización. En Uruguay se ha aprobado normativa referente a la protección de datos personales[12], exigiendo a las organizaciones garantizar la confidencialidad de los datos personales que manejan. Este tipo de normas jurídicas han impulsado la investigación y el desarrollo de técnicas y metodologías para la anonimización automática o semiautomática de los documentos.

El problema informático de anonimizar documentos no resulta trivial, más teniendo en cuenta que muchos de ellos no siguen un formato estructurado que permita identificar fácilmente la información sensible dentro de los mismos. Disciplinas computacionales tales como el procesamiento de lenguaje natural, la minería de textos, o el aprendizaje automático por máquinas, se presentan como herramientas aplicables para la resolución de este tipo de problemas. Desde el punto de vista de la arquitectura de software, la integración de diferentes elementos tecnológicos que se pueden utilizar en un proceso de anonimización tales como los mencionados, representa un tema de investigación en sí mismo.

En el marco de este proyecto, fueron estudiadas diversas propuestas de arquitecturas de anonimización tales como ANONIMYTEXT[41], MOSTAS [15], HIDE [33], y Etiquetador ESP[29]. De dichas propuestas se identificaron características comunes de los sistemas de anonimización, y se seleccionaron aquellas que se consideran de utilidad para la definición de una arquitectura de referencia, complementándolas con definiciones específicas de la propuesta que aquí se describe.

El presente trabajo de tesis, tiene los siguientes tres grandes objetivos:

1. Realizar un relevamiento de las arquitecturas existentes para sistemas de anonimización, así como las herramientas de software que se pudieran utilizar para este fin.

2. Diseñar y documentar detalladamente una arquitectura de referencia genérica para sistemas de anonimización.

3. Llevar a la práctica dicha arquitectura, mediante el diseño e implementación de un sistema concreto de anonimización para un dominio específico.

El resto del documento se organiza de la siguiente forma:

En la Sección 2, se introducen los conceptos básicos que se manejarán a lo largo de este trabajo, fundamentalmente el concepto de anonimización y sus marcos de aplicación.

En la Sección 3 se describen distintas arquitecturas de sistemas de anonimización estudiadas, en el marco del relevamiento realizado en este trabajo. Se estudiarán aspectos específicos de cada una, así como los puntos en común que se identificaron.

En la Sección 4 se describe la arquitectura de referencia definida en el marco de este trabajo, en la forma de un extracto del Documento de Arquitectura de Software (Software Architecture Document, S.A.D.).

En la Sección 5 se introducen las herramientas y tecnologías evaluadas con el fin de instanciar los diversos componentes de la arquitectura de referencia definida en la sección anterior.

En la Sección 6 se describe el prototipo de sistema de anonimización implementado, utilizando la arquitectura de referencia descrita en la Sección 4, e integrando diversas herramientas presentadas en la Sección 5.

Finalmente en la Sección 7 se presentan las conclusiones y trabajos futuros que surgen de este trabajo.

2. Contexto

En esta sección se presentan los principales conceptos que se desarrollarán en este trabajo de tesis. Se define el concepto de anonimización, y se presentan sus principales ámbitos de aplicación. Se asume tan solo un conocimiento previo de lo que se entiende por "Gestión Documental" y lo que es una Base de Datos Documental. En virtud de que esta sección es un extracto de la información incluida en el Anexo A de éste documento, de no estar familiarizado el lector con dichos conceptos se pueden encontrar allí sus definiciones.

2.1. Anonimización

Estando consolidada la gestión documental como una actividad importante dentro de las organizaciones, es habitual que éstas tengan en su poder un importante número de bases de datos documentales.[45] Dependiendo de su naturaleza y el dominio del negocio de la organización, estas bases de datos muchas veces son fuente de conocimiento que pueden ser aprovechadas por otras organizaciones o personas, en ocasiones con un enorme beneficio científico, jurídico[24], técnico o social.

Sin embargo muchas veces los documentos almacenados en estas fuentes de información, contienen información personal o sensible de personas físicas o jurídicas, cuya privacidad debe ser garantizada por la organización que gestiona la base documental. Existen fundamentos normativos y legislativos que hacen que la protección de los datos personales se vuelva una responsabilidad plausible de sanciones civiles o penales.[36, 8, 12]

La anonimización y la despersonalización de documentos, resulta una tarea tediosa, que implica un enorme esfuerzo cuando se trata de bases documentales muy grandes, y es allí donde aportan un enorme valor técnicas que permitan identificar en forma automática o asistida éstos datos sensibles.

En nuestro país existe legislación reciente vinculada a la protección de datos personales.[12] También se están realizando esfuerzos importantes en incorporar tecnologías de la información en los procesos de gobierno, con una creciente disponibilidad de documentos públicos hacia los ciudadanos. Un ejemplo claro de esta iniciativa fue la creación de la Agencia de Gobierno Electrónico y la Sociedad de la Información (AGESIC) [10]. Esta tendencia hace que la anonimización automática de documentos pueda ser una tecnología aplicable en el medio local en diferentes ámbitos, y resulta por tanto un tópico interesante para desarrollar en profundidad.

El verbo anonimizar es de reciente aceptación en el idioma español. Tal es así que el último diccionario de la Real Academia Española publicado (22a edición, año 2001) no lo define. Sin embargo en su diccionario en línea ya se lo ha incorporado aclarando que será incluido en la siguiente edición del diccionario (23a edición). La R.A.E. define (o definirá) anonimizar como "expresar un dato relativo a entidades o personas, eliminando la referencia a su identidad." [14]

Una muy buena definición de anonimización, se da en una ley española [8] que regula la investigación biomédica, dominio en el cual se estudiará ésta temática

más adelante. La ley 14/2007 del Reino de España, define en su artículo tercero la anonimización como el "proceso por el cual deja de ser posible establecer por medios razonables el nexo entre un dato y el sujeto al que se refiere".

Existen además dos niveles de "anonimización". La despersonalización (de-identification) y la anonimización propiamente dicha (anonimization)[8]. La anonimización implica la quita irreversible de toda información que permita identificar a un individuo u organización. La despersonalización sin embargo añade la posibilidad de que se guarde algún registro referencial que permita a una entidad autorizada o de confianza acceder a los datos personales eliminados.[8] La ley española citada, también da dos definiciones en relación a los datos que son anonimizados o despersonalizados respectivamente.

Se define como dato anonimizado o irreversiblemente disociado, a aquel "dato que no puede asociarse a una persona identificada o identificable por haberse destruido el nexo con toda información que identifique al sujeto, o porque dicha asociación exige un esfuerzo no razonable, entendiendo por tal el empleo de una cantidad de tiempo, gastos y trabajo desproporcionados."[8]

Como dato codificado o reversiblemente disociado, se entiende a aquel "dato no asociado a una persona identificada o identificable por haberse sustituido o desligado la información que identifica a esa persona, utilizando un código que permita la operación inversa." Este dato fue procesado en algún procedimiento de despersonalización.

Anonimizar un documento, puede entenderse como un proceso que contempla, entre otras, las siguientes tareas[27]:

- Eliminar o sustituir algunos nombres de personas (físicas o jurídicas), direcciones y demás información de contacto, números identificativos, apodos o cargos.
- Eliminar o sustituir algunos lugares mencionados (ciudades, barrios, regiones, instalaciones, monumentos, áreas naturales, etc.)
- Mantener otros nombres de entidades (personas, organizaciones o lugares) cuando aportan información relevante para el caso y no facilitan la identificación.
- En ocasiones es necesario también filtrar fechas o cantidades monetarias.
- Si se sustituyen referencias a entidades por etiquetas, es necesario mantener la consistencia a lo largo de un mismo documento, a pesar de que existan variaciones en la denominación (por ejemplo, si no se usa el nombre completo en todo el texto, si se usan alias, o si existen variantes debido a errores ortográficos).

2.2. Marcos de Aplicación

Existen diversos marcos de aplicación de la anonimización. Como ejemplos se pueden citar las organizaciones gubernamentales, la aplicación en las ciencias biomédicas (p.e. historia clínica), o en ámbitos judiciales.

En las organizaciones gubernamentales, por motivos de transparencia sus documentos deben estar accesibles al ciudadano, pero a su vez se deben proteger datos personales contenidos en ellos. De hecho, existen fundamentos normativos y legislativos que hacen que la protección de los datos personales se vuelva una responsabilidad plausible de sanciones civiles o penales. Particularmente en Uruguay existe una Ley de Protección de Datos personales [12] que regula la responsabilidad en la protección de los datos personales en poder de personas físicas o jurídicas en el territorio nacional. Entre otras cosas, la ley determina qué información es pública (p.e. nombres, apellidos, documentos de identidad) y qué información es definida como un dato sensible (p.e. datos que revelen origen racial y étnico, convicciones religiosas o morales o información referente a la salud o a la vida sexual) y por ende se necesita del consentimiento de la persona para que sea publicado.

Más allá de esta normativa particular, la definición de dato público o sensible depende del contexto y es variable a nivel mundial. Por ejemplo, la Ley 18.335 uruguaya [11] establece que "la historia clínica es de propiedad del paciente, será reservada y sólo podrán acceder a la misma los responsables de la atención médica y el personal administrativo vinculado con éstos, el paciente o en su caso la familia el el Ministerio de Salud Pública cuando lo considere pertinente." Sin embargo, la Ley de Portabilidad y Contabilidad de los Seguros de Salud (Health Insurance Portability and Accountability Act, HIPAA) de Estados Unidos[36] que regula aspectos similares define que un centro asistencial puede utilizar los datos de un paciente para investigación, sin su consentimiento, si se realiza un proceso de despersonalización de la información.

Un contexto de especial interés para este trabajo es el ámbito judicial ya que la administración de justicia por su alcance universal a la ciudadanía, muchas veces implica indagar en aspectos privados de las personas o las organizaciones, y ésta información puede quedar registrada en distintos documentos (expedientes, sumarios, presumarios, sentencias, etc).

El Poder Judicial uruguayo cuenta con una base de jurisprudencia que es accesible al público general a través de la web [24]. El proceso de anonimización de sentencias hoy en día se realiza en forma manual, y constituye un proceso bastante tedioso, y que insume numerosos recursos humanos. El usuario debe leer la sentencia contenida en un documento no estructurado, ubicar los datos sensibles, e irlos sustituyendo por identificadores genéricos para guardar cierta consistencia para la lectura del documento. En este contexto, el uso de una solución semi-automática puede resultar de gran valor.

3. Arquitecturas de Anonimización

Definidos los conceptos angulares de este trabajo, en esta sección se describirán las propuestas arquitecturales que fueron estudiadas y el estado del arte en arquitecturas de sistemas de anonimización. Las propuestas estudiadas surgen de algunos trabajos académicos, y herramientas de software disponibles enfocadas en la anonimización de documentos.

3.1. Propuestas existentes

Una primera iniciativa evaluada, enfocada en anonimización de documentos en idioma castellano, es la denominada ANONIMYTEXT[41]. Los autores proponen una arquitectura completa para un software de anonimización de información médica no estructurada. Para ello se utiliza una combinación de técnicas para lograr la identificación de la información sensible en el texto. El proceso propuesto, cuyo esquema se presenta en la Figura 1, comienza con un experto del negocio (un médico) que define una lista de conceptos "sensibles" que aparecen en los documentos como ser nombres o direcciones. Esta información es utilizada para generar un diccionario de términos que se utiliza en una segunda fase para realizar un análisis semántico del texto y etiquetar los conceptos de interés. Una vez etiquetado, se detecta la información sensible del documento basándose en la configuración que se lo provea de acuerdo a la normativa legal a aplicar. Seguidamente se propone que el documento sea revisado por un experto del negocio que aprueba, corrige o rechaza el texto, proveyendo en este último caso información que permita retroalimentar al sistema. Finalmente el documento se anonimiza siendo cifrada la información sensible con un algoritmo de clave pública o una función de hash.

Otra iniciativa es la arquitectura MOSTAS[15], vinculada a la identificación de términos biomédicos en documentos no estructurados en idioma español. El proceso propuesto se ilustra en la Figura 2. El sistema recibe las notas clínicas en formato no estructurado y se realiza un análisis morfo-semántico utilizando palabras en un diccionario general del lenguaje español. De esta manera se identifican los términos generales que no tienen valor desde el punto de vista biomédico. Las palabras no reconocidas por el analizador morfo-semántico se buscan en otros diccionarios más específicos, de siglas, abreviaturas y acrónimos biomédicos. Alimentado por los términos que aún no han sido reconocidos por el motor de búsqueda anterior, se procesa el texto anterior en la búsqueda de conceptos de interés para la anonimización. En MOSTAS el corrector ortográfico forma parte de un componente más grande junto con el anonimizador, y el analizador morfo-semántico se agrupa con otros componentes como se puede apreciar en la Figura 2.

Existen otras propuestas que proveen una solución total o parcial al problema siguiendo básicamente el mismo enfoque que las anteriores, aunque con algunas diferencias a destacar. Por ejemplo, el software de dominio público HIDE [25] utiliza un proceso de etiquetado que permite vincular toda la información sensible a una entidad (p.e. edad, dirección y nombre de una persona), así como

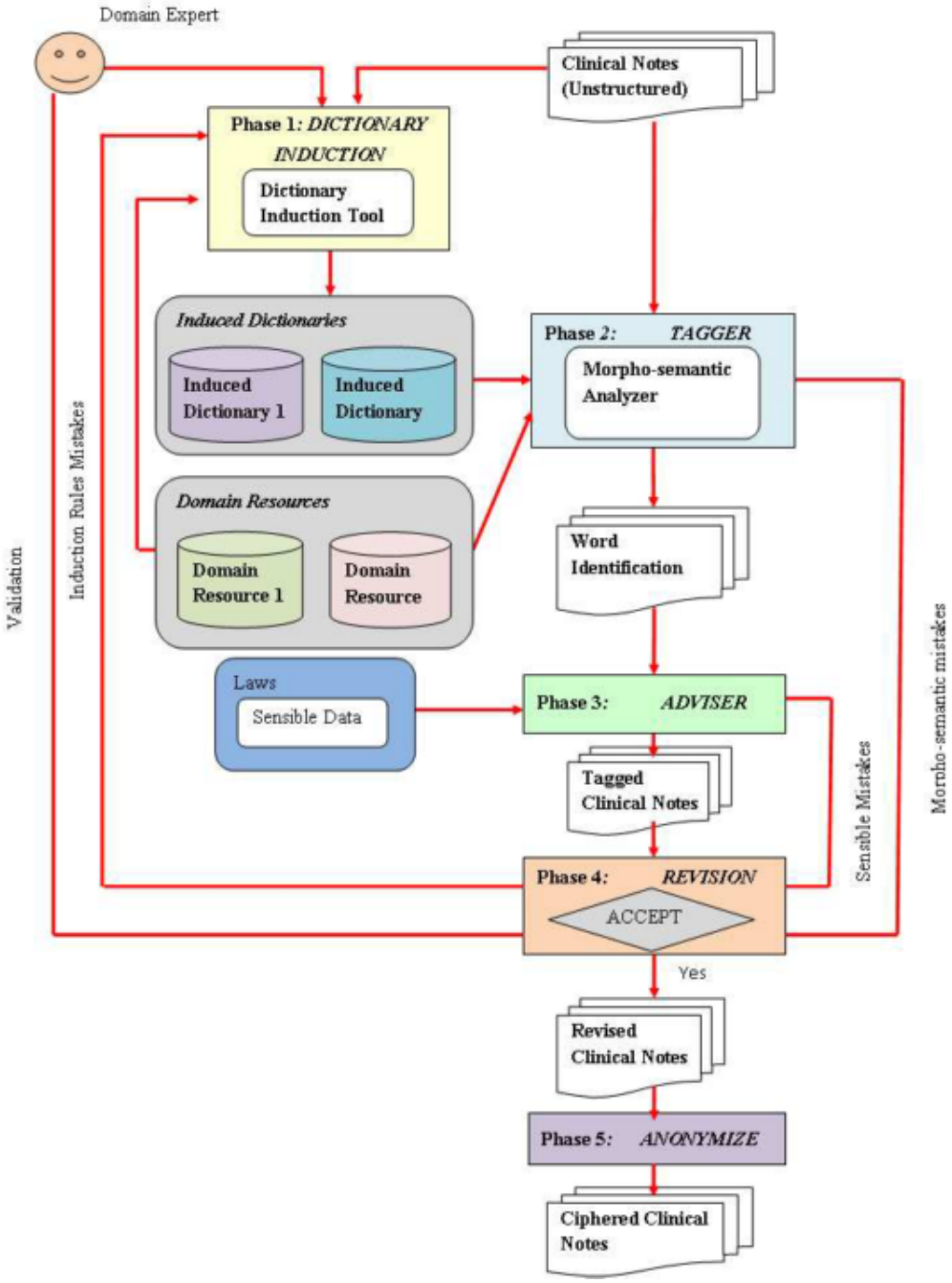

Figura 1: Arquitectura ANONIMYTEXT

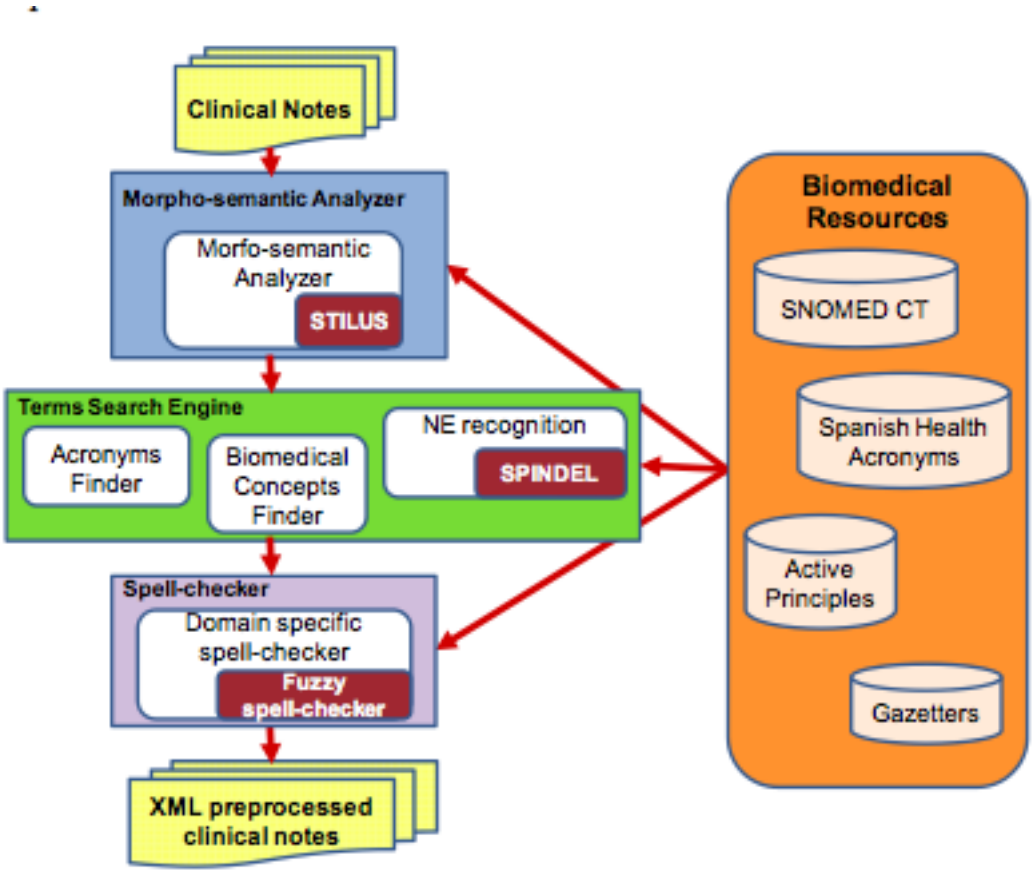

Figura 2: MOSTAS

permite seleccionar una estrategia de anonimización completa o parcial (de algunos atributos críticos). En la Figura 3 se puede apreciar una visión a alto nivel de los principales componentes de HIDE, así como su estrategia para realizar el etiquetado de texto.

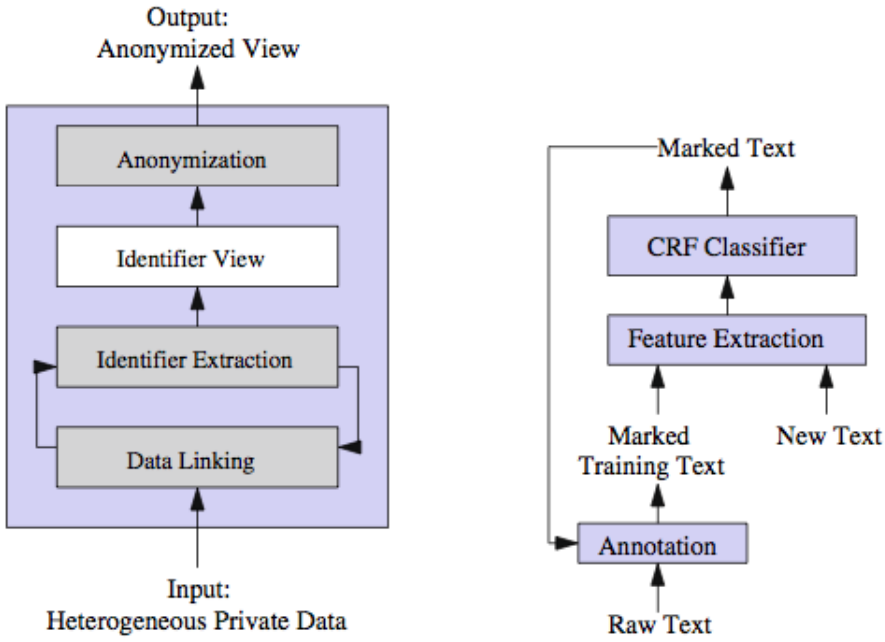

Figura 3: Clasificador HIDE

Por otro lado, se pueden utilizar otras herramientas de procesamiento de lenguaje natural como por ejemplo un etiquetador morfológico, el cual básicamente etiqueta los términos en un texto libre asignándole una lista de posibles categorías gramaticales. En un trabajo académico estudiado [29], se presenta un etiquetador morfosintáctico para el español que incorpora el uso de heurísticas para determinar con mayor exactitud la naturaleza de las palabras procesadas por el analizador morfológico. Una heurística concreta que se propone, de gran interés para un proceso de anonimización es la identificación de nombres propios utilizando un listado extenso de nombres y apellidos de personas, nombres de localidades, ciudades y países. Para la identificación del nombre propio se considera la aparición de la palabra en alguno de los listados mencionados, la presencia de una letra mayúscula al inicio de la palabra, la posición de la palabra dentro de la oración, el haber sido reconocida o no por el analizador morfológico, etc. La arquitectura de este etiquetador morfosintático, que si bien no es un sistema de anonimización presenta, algunas características similares desde el punto de vista de la arquitectura, se puede visualizar en la Figura 4.

Un elemento que llama la atención, es que se encuentran con facilidad nume-

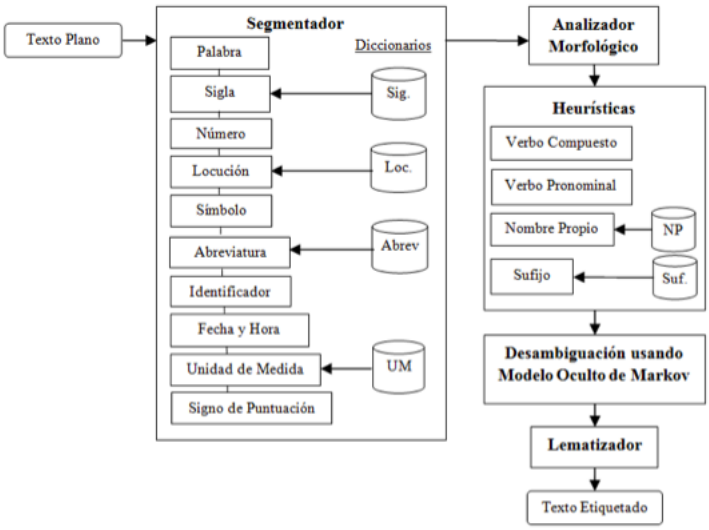

Figura 4: Etiquetador morfosintáctico para el español

rosas publicaciones académicas relacionadas a la temática de la anonimización automática de documentos, pero en general trabajando sobre documentos en el idioma inglés. Son abundantes las referencias en relación al ámbito médico en Estados Unidos, en cuanto a los requerimientos impuestos por la ley HIPAA.

3.2. Aspectos a destacar

Del análisis de las arquitecturas estudiadas, se recogen una serie de elementos en común, y otros específicos que se consideran de interés para ser incorporados en una posible arquitectura de referencia. Por otra parte, se concluye que existen distintos mecanismos automáticos que se pueden utilizar para implementar un proceso de anonimización de documentos. Son frecuentemente utilizadas tecnologías basadas en estadística y procesamiento de lenguaje natural (PLN), como también se aplica reconocimiento de patrones, expresiones regulares y reglas. Se encuentran propuesta que se basan en diccionarios para identificar la información sensible, así como complejos algoritmos basados en aprendizaje de máquinas y/o inteligencia artificial. Pero también es posible utilizar combinaciones de las técnicas anteriores, siendo frecuente la combinación de expresiones regulares con los diccionarios.

Se presentan los aspectos comunes y específicos anteriormente mencionados en forma tabular en el cuadro 1.

Tabla 1: Aspectos comunes y específicos

Módulos	Anonimytext	MOSTAS	HIDE	Etiquetador ESP
NER	Si (Tagger)	Si (NE recognition)	Si (Ident. Extract)	Si (Analizador)
Corrector Ortográf.	No	Si (Spell-checker)	No	No
Heurísticas	Si (Laws)	Si (Terms S. Eng.)	No	Si (Heurísticas)
NE Clustering	No	No	No	No
Revisor	Si (Revision)	No	Si (Identifier View)	Si (Ident. View)
Anonimizador	Si (Anonymize)	Si (NE recognition)	Si (Anonymization)	No

3.2.1. Aspectos comunes

Una de las primeras conclusiones que surgen, es que en la totalidad de las propuestas estudiadas la piedra angular de la anonimización es un módulo o componente NER que aparece en todas las arquitecturas vistas, es decir la identificación de Entidades con Nombre (Named Entities, N.E.). Las entidades con nombre o "named entities" son justamente lo que se busca anonimizar, no siendo éstas otras cosa que los datos sensibles del documentos tales como "nombres de personas, organizaciones, localizaciones, expresiones de horas, cantidades, valores monetarios, porcentajes, etc."[3]. Este componente debe recibir como insumo un texto no estructurado, procesarlo, y brindar como salida el mismo texto pero con sus Named Entities identificadas mediante algún tipo de marca o etiqueta. Un sistema encargado del Reconocimiento de Entidades con Nombre (Named Entity Recognition, NER) persigue delimitar en un texto arbitrario aquellas frases simples que responden de forma directa a preguntas del tipo ¿quién?, ¿dónde?, ¿cuándo? o ¿cuánto?.[16]

De acuerdo a lo investigado, existen diversas herramientas que permiten realizar N.E.R., algunas con mayor o menor precisión en los resultados, y algunas proveen características adicionales como la clasificación de entidades con nombre. Cabe pensar entonces en una propuesta arquitectural que permita adaptar o intercambiar fácilmente estos "motores" NER, es decir que podría pensarse en contar con alguna capa de abstracción, de forma de adaptar las interfaces a distintas herramientas y tecnologías.

En todas las propuestas específicas de arquitecturas de anonimización (ANONIMITEXT, MOSTAS y HIDE), se identifica un componente que realiza el procesamiento final del texto que es nada menos que el anonimizador. También se visualizan especializaciones de este módulo, dado que la anonimización puede ser reversible o irreversible, parcial o total (en cuanto a los atributos que se anonimizan), de acuerdo a los requerimientos que se planteen cada caso.

3.2.2. Aspectos específicos

Luego del procesamiento de Named Entities, distintos enfoques arquitecturales proponen el concepto de retroalimentación del sistema de alguna forma. Uno de los enfoques que resultan interesantes en este sentido, propuesto por MOSTAS, es el de procesar los documentos utilizando algún corrector ortográ-

fico cuando quedan elementos sin identificar. En la propuesta de [29], se aplican una serie de heurísticas luego de que el texto pasa por el analizador morfológico. Sería deseable en una arquitectura de referencia, que éstas posibles etapas de postprocesado del texto tuvieran interfaces de entrada y salida equivalentes, de forma que distintos componentes pudieran agregarse o quitarse tal como si fueran eslabones en una cadena.

Otro enfoque vinculado a la retroalimentación, en este caso asistida, es el de permitir a un experto identificar el tipo de errores cometidos en el proceso NER mediante alguna interfaz de usuario acorde, y luego retroalimentar el sistema con esta información.

En las distintas propuestas vistas, en algunos casos se utilizan analizadores morfológicos para identificar y clasificar las palabras, en otros casos herramientas basadas en diccionarios y motores de reglas, y también se vieron propuestas híbridas.

Finalmente, en relación al clustering de Named Entities, se puede pensar en un aspecto adicional a considerar en una arquitectura de referencia, que permita tener en cuenta este importante factor al anonimizar.

4. Arquitectura de Referencia

En esta sección se describe de forma resumida la Arquitectura de Referencia definida una vez fue completada la investigación del estado del arte en arquitecturas de anonimización. La descripción detallada de la arquitectura se encuentra documentada en el Anexo B, mediante el correspondiente Documento de Arquitectura de Software, del cual esta sección es un extracto.

4.1. Contexto y Análisis Funcional del Sistema

El alcance de la propuesta abarca la descripción de la arquitectura en forma genérica. No se presenta un sistema en particular sino la arquitectura en la cual podrán basarse implementaciones concretas. Por tal motivo se seleccionará un conjunto de vistas tendientes a describir aspectos generales de la arquitectura, y no aquellos específicos que serán de particular interés a la hora de implementar un sistema de anonimización concreto.

Durante el estudio del estado del arte en anonimización, se determinaron una serie de requerimientos que guían y condicionan la arquitectura de referencia que se propone (Architectural Drivers o simplemente Drivers).

Se resumirán dichos drivers a continuación:

1. **Adaptabilidad**: En distintas propuestas de anonimización estudiadas, se pudo visualizar al proceso de anonimización como una cadena de subprocesos donde se aplican distintas herramientas sobre el texto del documento. A su vez, se estudiaron distintas instanciaciones tecnológicas para cada uno de esos subprocesos, y se encontró gran variedad de herramientas que permiten realizar cada una de estas tareas con distintas ventajas y desventajas (términos de licenciamiento, efectividad, nivel de configuración, etc). Una arquitectura de referencia podría beneficiarse mucho si permitiera abstraer de alguna manera la interacción (interfaces) entre estos subprocesos, de forma que se pudiera optar por uno u otro en base a configuración.

2. **Proceso**: El proceso es otro conductor de la arquitectura. En el contexto del proyecto fueron estudiadas diversas propuestas de arquitecturas para sistemas de anonimización, donde se pudo ver un común denominador: un proceso en etapas, donde el texto va pasando de un módulo hacia otro hasta que se logra determinar las entidades con nombre (Named Entities), y las mismas pueden finalmente ser anonimizadas. No se encuentran elementos que permitan pensar en una propuesta diferente para una arquitectura de referencia, por tanto la misma deberá soportar un proceso similar al visto en las arquitecturas preexistentes. El proceso descrito será detallado en la vista funcional más adelante en éste documento.

3. **Configuración**: Del driver anterior se desprende otro aspecto conductor de la arquitectura. Se busca una propuesta de arquitectura que permita

optar por configuración por diferentes herramientas que se puedan utilizar para realizar los distintos procesamientos que se realizan sobre el texto. Además es posible que incluso alguna de éstas etapas pudiera directamente activarse o desactivarse, de acuerdo a los requerimientos del usuario en cada caso.

En la Figura 5 se presenta un diagrama donde se pueden ver los "actores" típicos que interactúan con un sistema de anonimización. El software de anonimización presenta interfaces hacia otros sistemas, típicamente sistemas de gestión documental, mediante las cuales recibe la información (documentos) a anonimizar, y luego de procesarlos los devuelve al sistema de origen. No se planteo como un objetivo de un sistema de anonimización almacenar la información anonimizada ni presentarla a usuarios finales, la idea es que la anonimización sea un servicio semiautomático que puede ser utilizado por otros sistemas de la organización. Como único actor humano del sistema de anonimización, se presenta un experto del dominio, que en caso de que el sistema de anonimización contemple una etapa de validación será quien deberá aprobar o rechazar el texto procesado por el sistema, y retroalimentar si así corresponde al propio software.

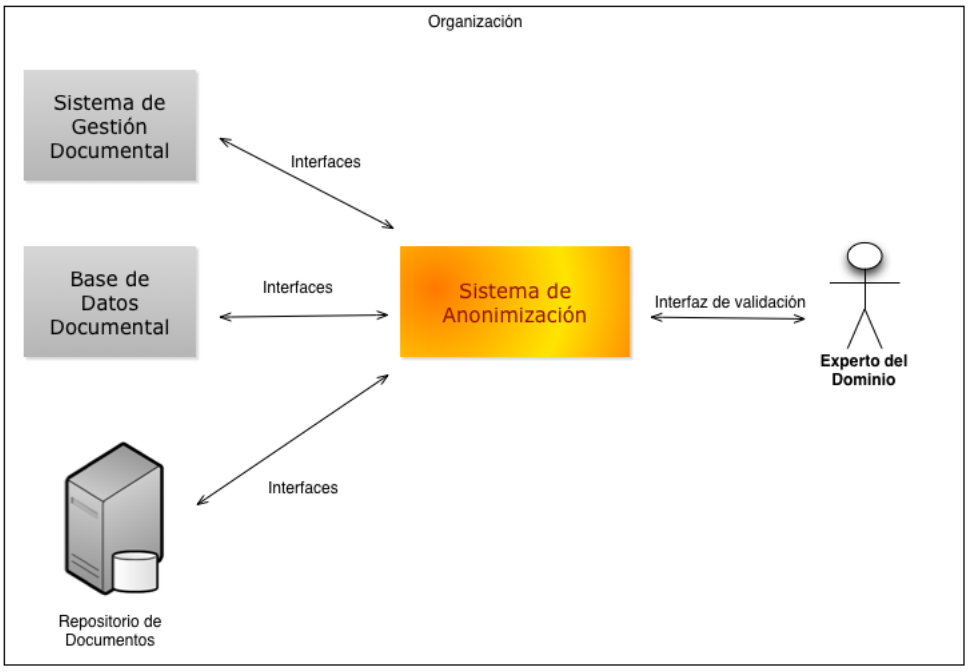

Figura 5: Diagrama de Contexto

Identificados los principales drivers y presentado el contexto del sistema, se deben especificar los stakeholders (interesados) de la arquitectura, cuya definición condicionará fuertemente los aspectos a describir de la propuesta. Para la

Arquitectura de Referencia se identificaron los siguientes stakeholders:

- **Arquitectos de Software**: Por tratarse de una arquitectura genérica "de referencia", los principales interesados en la presente arquitectura son arquitectos de software que pretendan diseñar sistemas para la anonimización de documentos de texto.
- **Desarrolladores:** Programadores o analistas interesados en implementar alguna "instanciación" de ésta arquitectura de referencia en un sistema concreto. Puede ser de interés conocer los conceptos de alto nivel y los drivers que llevaron a definir la estructura general de la arquitectura.
- **Estudiantes o personal del ámbito académico:** Interesados en la temática de la anonimización o las arquitecturas de software, así como lógicamente los directamente interesados en el presente trabajo de tesis (tutor, tribunal, etc.)

Para el diseño y la descripción de ésta arquitectura, se consideraron diferentes propuestas para documentarla, tales como la de BassL. et al. [28], KrutchenP. [39] y Rozansky/Woods N. and E. [35].

La propuesta de Bass propone documentar la arquitectura basándose en un modelo de vistas (Views). Las vistas son una representación de la arquitectura de un sistema, enfocada en un conjunto de intereses concretos, como pueden ser elementos de desarrollo, de despliegue, de flujo de información, etc. Las Vistas son seleccionadas cuidadosamente teniendo en cuenta especialmente a los interesados (stakeholders) del software a documentar, así como otras características específicas del mismo.

Por su parte, la propuesta de Krutchen, el conocido modelo 4+1, utiliza este mismo concepto pero predefine o preselecciona un conjunto de cuatro vistas a modo de estándar (Lógica, de Desarrollo, de Procesos y Física), agregando además los escenarios de casos de uso como elemento o "vista" adicional para describir la arquitectura del software.

Finalmente, la propuesta de Rozanski/Woods, también incluye el concepto de vistas, pero lo complementa introduciendo dos nuevos elementos denominados "Puntos de Vista" (ViewPoints) y "Perspectivas" (Perspectives). Un ViewPoint es un conjunto de buenas prácticas, patrones y plantillas para guiar en la definición de las vistas arquitecturales. La idea detrás de esto es fomentar la reutilización del conocimiento adquirido al documentar una arquitectura. Por otra parte, el concepto de Perspective podría decirse que aporta una nueva "dimensión" para documentar la arquitectura, ya que se orienta especialmente a documentar aquellos clásicos atributos de calidad del software que afectan al sistema de forma transversal, impactando las diferentes vistas y no siendo fácilmente representables en una sola o en un subconjunto de éstas. Ejemplos de tales atributos de calidad o requerimientos no funcionales pueden ser la seguridad, la performance, la usabilidad, etc.

Para la documentación de esta arquitectura de referencia se definió optar por la propuesta de Rozanski y Woods, ya que como se puede apreciar dicha

propuesta aporta una mayor flexibilidad y profundidad a la hora de describir una arquitectura, con la adición de las Perspectives, pero además provee un conjunto de herramientas o plantillas concretas que facilitan la documentación a través de los ViewPoints. Basado en dichos ViewPoints se definió un conjunto de Vistas, teniendo en cuenta el objetivo de reflejar una arquitectura genérica como la propuesta, y teniendo en cuenta los intereses de los stakeholders especificados oportunamente, a saber:

1. **Vista Funcional**: Mediante esta vista se describirá la responsabilidad de cada componente, sus interfaces e interacciones con el resto del sistema, y fundamentalmente sus responsabilidades.

2. **Vista de Información**: Describirá la forma en que se fluye la información en el sistema. En el caso de la arquitectura existirá un flujo constante de información proveniente del texto que se está procesando. Se entiende que mediante ésta vista se podrá describir el mismo en detalle, incluyendo el formato y estructura de la información, su flujo dentro del sistema, etc.

3. **Vista de Desarrollo**: Describirá aspectos de interés para los interesados (stakeholders) involucrados en la implementación o instanciación de la arquitectura propuesta, tales como la organización de los módulos y su estructura.
   Con estas definiciones, se está en condiciones de presentar las metas del sistemas, así como especificar los principales requerimientos funcionales y no funcionales que la arquitectura deberá resolver.

El proceso de anonimización implica analizar un texto identificando cada referencia a persona, lugar u organización cuyos datos se desean proteger, sustituyéndolos por una versión cifrada de los mismos (cifrado reversible), o directamente por nombres genéricos que no permitan identificarlos. El sistema de anonimización tiene como meta principal implementar un mecanismo automático o semiautomático, que permita eliminar la información sensible en documentos no estructurados.

- Como estímulo de negocio (business driver*)* se presenta el avance de legislación específica en lo que refiere a la protección de datos personales. De acuerdo al estudio del estado del arte realizado, se verificó la existencia en otros países de normas específicas orientadas a la anonimización de información de pacientes en historias clínicas. En Uruguay se ha comenzado a legislar en la materia[12], y si bien aún la normativa es algo genérica, es de esperarse que a medida que se vayan reglamentando las leyes surjan requerimientos específicos que puedan ser atendidos por procesos de anonimización.

- La definición de una arquitectura de referencia además, pretende tomar las mejores ideas de variadas arquitecturas estudiadas, y proponer un diseño de alto nivel que por sobre todo sea adaptable e interoperable con distintas herramientas de software preexistentes. Debe proveer la documentación

necesaria y un diseño que permita integrar nuevas herramientas, sin que ello implique una reingeniería del sistema.

A continuación se enumeran los requerimientos funcionales y no funcionales identificados para un sistema de anonimización.

En el Anexo B (SAD), se describen una serie de escenarios funcionales y no funcionales para ilustrar cómo el sistema deberá responder ante los distintos estímulos que surgen de los requerimientos.

Se presentarán a continuación las vistas que fueron seleccionadas, de acuerdo al criterio especificado previamente, de forma resumida.

4.2. Vista Funcional

En primer lugar se presenta la Vista Funcional de la arquitectura. A través de la misma, se introducen los diferentes elementos funcionales del sistema de anonimización, sus responsabilidades y la forma en que se comunican entre sí. Asimismo se presentan los parámetros de configuración utilizados por los componentes funcionales.

El proceso fue definido como uno de los principales drivers de la arquitectura. En el contexto del proyecto fueron estudiadas diversas propuestas de arquitecturas para sistemas de anonimización, donde se pudo identificar un común denominador: un proceso en etapas, donde el texto va pasando de un módulo hacia otro hasta que se logra reconocer las entidades con nombre, y las mismas pueden finalmente ser anonimizadas.

En primer lugar el sistema de anonimización deberá tener algún tipo de interfaz de usuario (UI), donde presentará los documentos en sus estados iniciales y anonimizados. Los documentos podrían ser ingresados directamente por los usuarios de alguna manera, o provendrían de fuentes externas de documentos, tales como un sistema de gestión documental o una base de datos documental.

La interfaz de usuario deberá permitir además parametrizar y configurar al sistema de acuerdo a las necesidades específicas del usuario. Esta interfaz de usuario además deberá permitir manejar distintos niveles de seguridad, es decir que se deberá proveer alguna interfaz de autenticación de usuarios, para contemplar la diferenciación entre los usuarios comunes y los expertos/validadores que se mencionaron en la especificación de requerimientos.

Por otra parte, la lógica de negocios y el motor del sistema encargado de gestionar el proceso, se puede abstraer en una capa del sistema donde encontraremos diversos módulos o componentes que se describirán a continuación.

El núcleo del proceso, y por ende de esta capa lógica del sistema, se centra en el reconocimiento de las entidades con nombre, que se puede definir como "una subtarea de la recuperación de información cuyo objetivo es localizar y clasificar los elementos atómicos en el texto sobre categorías predefinidas como nombres de personas, organizaciones, localizaciones, expresiones de horas, cantidades, valores monetarios, porcentajes, etc."[3]

El componente que llamaremos NER, tiene la responsabilidad de procesar el reconocimiento primario de entidades con nombre, y debe recibir como insumo

Tabla 2: Requerimientos

Referencia	**Descripción del Requerimiento**
RF1	El sistema debe poder procesar documentos no estructurados, en formato texto plano.
RF2	Se deben poder utilizar distintas herramientas de software para el procesamiento del texto indistintamente. La herramienta a utilizar en cada etapa debe ser un elemento más de configuración.
RF3	El sistema debe permitir a un experto del dominio validar la salida anonimizada, y permitir aprobar el documento o reprobarlo brindando feedback que retroalimente al sistema.
RF4	Se debe poder configurar el alcance de la anonimización, la cual puede ser total (todos los atributos que identifican a personas u organizaciones), o parcial (un subconjunto de los atributos).
RF5	El sistema debe permitir por configuración, realizar una anonimización irreversible (se elimina por completo la información sensible), o reversible (se sustituye la información sensible por una referencia cifrada a la misma).
RF6	El sistema debe permitir la introducción de reglas o patrones (heurísticas) definidas por un usuario experto, para contemplar características específicas de los documentos de un dominio en particular.
RF7	El sistema debe contemplar la posibilidad de interactuar con fuentes externas, mediante adaptadores, que provean información de términos específicos de un dominio (diccionarios, tesauros, gacetillas, acrónimos, etc.)
RNF1	Adaptabilidad: Para dar soporte al RF2.
RNF2	Configurabilidad: Para dar soporte a RF2 y RF5
RNF3	Interoperabilidad: El sistema deberá interoperar con distintas fuentes externas (RF7)
RNF4	Documentación: Las interfaces del sistema deberán estar debidamente documentadas, de forma que un usuario avanzado pueda agregar nuevas fuentes de conocimiento o herramientas de procesamiento de lenguaje natural.
RNF5	Extensibilidad: El añadir nuevas herramientas de procesamiento de lenguaje (RF2 y RNF1) deberá ser un proceso sencillo, que no implique modificaciones mayores al sistema.
RNF6	Auditoría: El sistema deberá registrar las interacciones de usuarios con los documentos.
RNF7	Seguridad: Los documentos no anonimizados no deben ser accedidos por usuarios no autorizados.

un texto no estructurado, procesarlo, y brindar como salida el mismo texto pero con sus entidades con nombre identificadas mediante algún tipo de marca o etiqueta. De acuerdo a lo investigado, existen diversas herramientas de procesamiento del lenguaje natural, que permiten realizar NER, algunas con mayor o menor precisión en los resultados. Estas herramientas utilizan técnicas basadas en estadísticas, aprendizaje de máquinas, inteligencia artificial, diccionarios o combinaciones de dichas técnicas, para identificar las entidades con nombre en el texto.

Algunas herramientas proveen características adicionales como la clasificación de entidades con nombre. La clasificación no es otra cosa que categorizar las entidades con nombre de acuerdo a su naturaleza (por ejemplo determinar si la entidad con nombre se corresponde a un nombre propio, una localización geográfica, un identificador único de una persona, etc.).

Se pensó entonces en un proceso que permita adaptar o intercambiar fácilmente estos "motores" NER, contando a su vez con una capa de abstracción, de forma de adaptar las interfaces a distintas herramientas y tecnologías.

El proceso contempla además la integración de herramientas que permitan agrupar las entidades con nombre en clusters (módulo "Clustering"). Esto permite por ejemplo que conceptos equivalentes (por ejemplo una sigla y su significado, "O.N.U. = Organización de las Naciones Unidas"), puedan ser identificados como una misma entidad con nombre, y en consecuencia la anonimización aplique el mismo criterio para las equivalencias, aportando a conservar de mejor manera la coherencia del documento anonimizado.

También se incorpora la identificación de entidades con nombre adicionales o específicas mediante el uso de heurísticas, tales como la identificación de reglas, expresiones regulares o patrones específicos, que puedan ser personalizados de acuerdo al dominio del cuerpo de texto a analizar. En el proceso se definen tareas específicas en este sentido, y se deja la puerta para integrar herramientas adicionales que se pudieran adaptar para determinado dominio específico (diccionarios o tesauros, categorización mediante servicios web, etc).

En todas las propuestas específicas de arquitecturas de anonimización, se identifica otro componente mediante el cual se realiza el procesamiento final del texto, que es concretamente el módulo anonimizador. Se visualizan especializaciones de este módulo, dado que la anonimización puede ser reversible o irreversible, parcial o total (en cuanto a los atributos que se anonimizan), de acuerdo a los requerimientos que se planteen cada caso.

Como se mencionó anteriormente, la lógica del sistema deberá interoperar con diversas fuentes y herramientas externas, tales como las herramientas NER o las herramientas de clustering.

Toda la interoperabilidad del sistema, ya sea con herramientas externas, fuentes de conocimiento, o las propias fuentes de documentos, se gestionará con una capa de integración que denominaremos "Conectores".

Finalmente, los requerimientos del sistema especifican la necesidad de contar con auditoría en el proceso, así como seguridad para los documentos que se procesan. Estos dos componentes son afectan al sistema en distintos niveles, y los podemos visualizar como aspectos o capas transversales.

Vinculado a la seguridad, en el sistema se deberían definir los siguientes roles, de acuerdo a los requerimientos RF3 y RNF7:

1. Rol "Usuario": Rol genérico a asignar a los usuarios del sistema. Permite procesar documentos en el sistema de anonimización con la configuración predeterminada.

2. Rol "Revisor": Este rol se asigna a los expertos del dominio de los documentos que se procesan. Tiene la potestad de validar un documento procesado por el sistema de anonimización. El usuario revisor podrá además retroalimentar al sistema de manera de que se enriquezca el proceso de anonimización con su conocimiento, y además podrá reconfigurar el sistema de acuerdo a las necesidades.

En base a todos los elementos mencionados, en la Figura 6 se ilustra el sistema y los componentes descritos en base a un modelo de capas clásico. Allí se puede visualizar:

1. La interfaz de usuario gestionada por la capa UI.

2. La capa lógica donde residen los componentes centrales del sistema descritos anteriormente (NER, clustering, reglas y patrones), así como el propio motor de la aplicación.

3. Una subcapa de persistencia, que gestiona los datos y el acceso a las bases de datos del propio sistema.

4. Una capa de conectores, para interoperar con servicios y fuentes externas.

5. Las capas de seguridad y auditoría transversales.

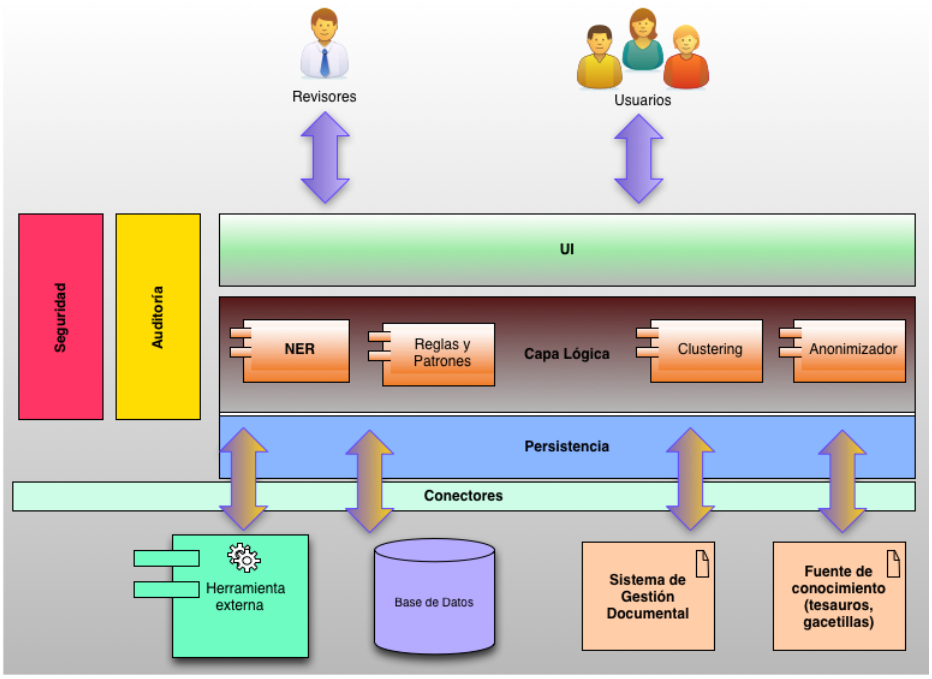

Figura 6: Modelo en capas del sistema

4.2.1. Modelado como proceso de negocios utilizando BPMN

A partir de la especificación del sistema descrita anteriormente, se visualizó que esta arquitectura particular permitía modelar el sistema como un proceso de negocios, mediante notación BPMN2[37].

El proceso es uno de los conductores de la arquitectura, y se presentan algunos requerimientos donde los motores de procesos y los Sistemas de Gestión de Procesos de Negocio (Business Process Management Systems, BPMS) son especialmente útiles. Por BPMS nos referimos a un sistema integral para la implementación de procesos de negocio, que permite además de modelar, implementar y ejecutar los procesos, el modelado de la GUI, aportando infraestructura de seguridad y auditoría embebidas entre otros elementos.

En primer lugar vemos que el proceso de anonimización descrito, puede ser perfectamente modelado como un proceso de negocios, con tareas y transiciones bien definidas. De esta manera, se aporta entonces la flexibilidad y la potencia del modelado de procesos de negocios que permite adaptar con facilidad el proceso frente a distintos escenarios, como se puede apreciar en la Figura 7.

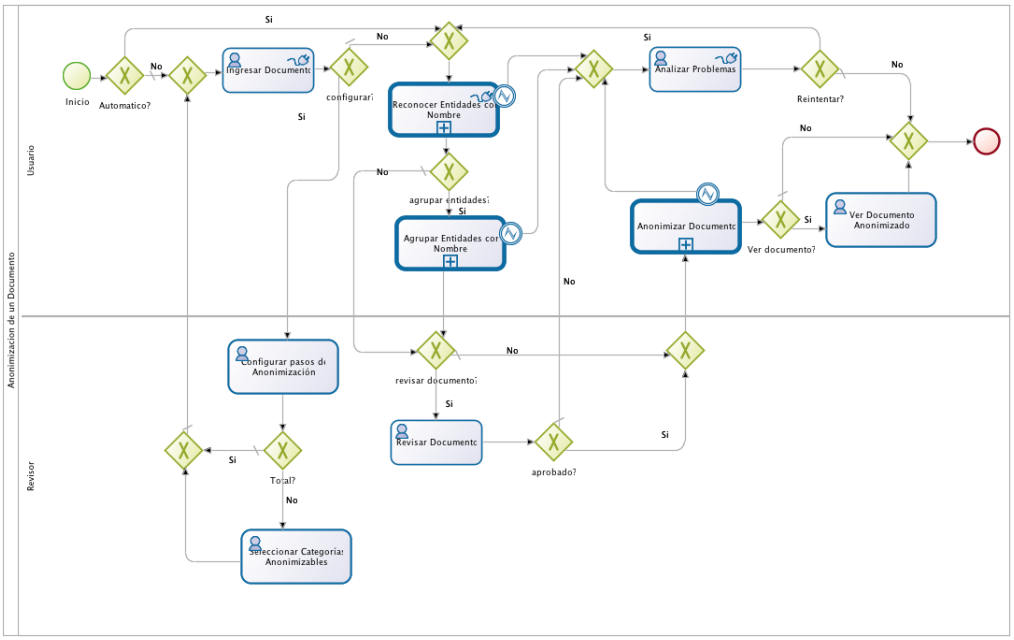

Figura 7: Proceso modelado mediante BPMN2

La propuesta de modelado como proceso de negocios[2] constituye una característica específica de la arquitectura propuesta, que la diferencia con las arquitecturas estudiadas.

- El modelado del proceso mediante BPMN2[37], permite que se pueda personalizar el mismo, por ejemplo habilitando o deshabilitando ciertas tareas de acuerdo a la necesidad, sin que ello derive finalmente en una implementación o programación. Por otra parte definir el proceso mediante este estándar de la industria, está alineado con el principio de interoperabilidad definido como lineamiento general para ésta arquitectura.

- Tratándose de una arquitectura genérica, y teniendo en cuenta el requerimiento de adaptabilidad definido (RNF1), es necesario proveer mecanismos para que el proceso aquí descrito pueda adaptarse a dominios específicos y sistemas concretos. Una vez más la flexibilidad de BPMN2 permite atender esta necesidad.

- Se presentan requerimientos de auditoría y seguridad transversales, con necesidad de asignación de roles para distintas tareas. Estos elementos son parte integral de las BPMS, y por ende se obtiene una notable ganancia en productividad a la hora de desarrollar un sistema de anonimización, ya que se podrá utilizar la infraestructura de seguridad y auditoría provista por el BPMS.

Pero quizás la mayor ganancia de utilizar BPMN2 es que una vez definido en este lenguaje, el proceso se puede pasar rápidamente del diseño a la ejecución sobre alguno de los motores de procesos de las BPMS existentes compatibles con este estándar. De esta manera toda la arquitectura presentada en la Figura 6 queda inmersa en el BPMS, a excepción de los componentes y sistemas externos que se pretendan integrar, tales como las que figuran en la zona inferior de dicha figura (herramientas externas, base de datos, sistemas de gestión documental y fuentes de conocimiento).

Finalmente, cabe destacar que de esta manera el proceso se vuelve extensible con facilidad, ya que se pueden incorporar nuevos elementos (tareas, subprocesos) nuevamente sin impacto en el diseño.

A continuación se describen las entradas, salidas y responsabilidades de los subprocesos y tareas del proceso.

Tareas

Nombre	**Ingresar Documento**
Responsabilidades	Esta tarea inicializa el documento a ser anonimizado.
Entrada	Un documento a anonimizar
Salida	Documento a ser anonimizado inicializado en el sistema.

Nombre	**Configurar pasos de la anonimización**
Responsabilidades	Esta tarea gestionará toda la configuración del sistema. Establece los parámetros de configuración que son utilizados por los distintos componentes.
Entrada	Documentos seleccionados para anonimizar
Salida	Variables y parámetros de configuración establecidos.

Nombre	**Seleccionar Categorías Anonimizables**
Responsabilidades	Configuración complementaria. De tratarse de un proceso de anonimización parcial, permite seleccionar las categorías de Entidades con Nombre a ser anonimizadas.
Entrada	Variable “anonimización parcial” establecida.
Salida	Lista de categorías de Entidades con Nombre a anonimizar.

Nombre	**Reconocer Entidades Con Nombre (Subproceso)**
Responsabilidades	Este componente es el subproceso encargado de identificar las Entidades con Nombre en el texto.
Entrada	Un documento a anonimizar.
Salida	Lista de Entidades con Nombre identificadas en el documento.

Nombre	**Agrupar Entidades con Nombre (Subproceso)**
Responsabilidades	Este componente agrupa las entidades con nombre equivalentes en clusters.
Entrada	Documento con sus Entidades con Nombre identificadas.
Salida	Las entidades con nombre equivalentes agrupadas en clusters.

Nombre	**Revisar Documento**
Responsabilidades	Aprobar o rechazar la identificación de Entidades con Nombre realizada.
Entrada	Documento con sus Entidades con Nombre identificadas y agrupadas.
Salida	Documento aprobado o rechazado.

Nombre	**Anonimizar Documento (Subproceso)**
Responsabilidades	Este componente tiene la responsabilidad de remplazar la información sensible con datos genéricos o versiones cifradas de los datos, según la configuración del sistema.
Entrada	Documento aprobado, con sus Entidades con Nombre identificadas y agrupadas.
Salida	Documento anonimizado.

Nombre	**Ver Documento Anonimizado**
Responsabilidades	Mostrar el documento anonimizado al usuario.
Entrada	Documento anonimizado.
Salida	Fin del proceso.

Parámetros de Configuración

El proceso principal estará condicionado por una serie de variables y parámetros para representar las distintas configuraciones que son admitidas por el sistema. En el Cuadro 4 se detallan dichos parámetros.

Tabla 4: Parámetros

Nombre	Tipo	Parámetro	Método de Introducción
Texto	model.Text	El objeto documento a anonimizar. El tipo de datos model.Text será detallado en la Vista de Información más adelante en éste documento.	Interfaz con otro sistema, base de datos o introducción directa por parte del usuario.
Aprobado	Booleano	Variable para representar la aprobación o rechazo del documento	Establecida por el usuario con un checkbox en la tarea "Revisar Documento"
Automático	Booleano	Define si el proceso de anonimización será automático o será supervisado.	Establecida por el usuario con un checkbox en la tarea "Configurar pasos de la anonimización"
Agrupar	Booleano	Define si se deberán agrupar (clustering) las entidades con nombre.	Establecida por el usuario con un checkbox en la tarea "Configurar pasos de la anonimización"
Reversible	Booleano	Define si la anonimización será reversible o irreversible.	Establecida por el usuario con un checkbox en la tarea "Configurar pasos de la anonimización"
herramientaNER	Texto	Nombre de la herramienta NER que se utilizará.	Seleccionable por el usuario en un combo box en la tarea "Configurar pasos de la anonimización"
herramientaClustering	Texto	Nombre de la herramienta de clustering que se utilizará.	Seleccionable por el usuario en un combo box en la tarea "Configurar pasos de la anonimización"
heuristicas	Booleano	Define si se procesará el documento mediante reglas heurísticas, reglas y patrones.	Establecida por el usuario con un checkbox en la tarea "Configurar pasos de la anonimización"
Total	Booleano	Define si la anonimización será parcial (false) o total (true).	Establecida por el usuario con un checkbox en la tarea "Configurar pasos de la anonimización"
Otra_herramienta	Booleano	Define si se deberán invocar herramientas externas adicionales.	Establecida por el usuario con un checkbox en la tarea "Configurar pasos de la anonimización"

Dentro del proceso de anonimización, se identificaron los subprocesos que se describen a continuación.

Subproceso Reconocer Entidades con Nombre

Este subproceso se ilustra en la Figura 8, y tiene como cometido identificar las entidades con nombre. Como tareas del proceso destacan la invocación de herramientas NER, herramienta externas (por ejemplo tesauros, gacetillas, etc), y la invocación de motores de reglas y patrones.

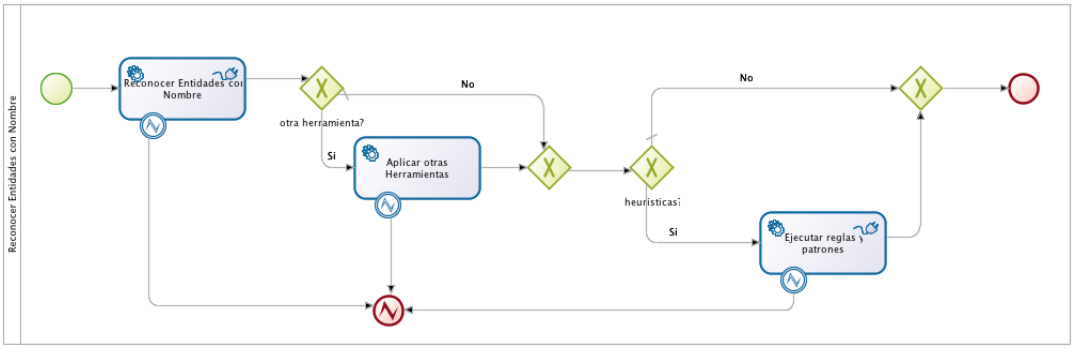

Figura 8: Subproceso Reconocer Entidades con Nombre

Dentro de este subproceso, se encuentra una primera tarea que es la que efectivamente invoca alguna herramienta con capacidades NER concretas. Esta tarea se puede encontrar en las arquitecturas estudiadas. Por ejemplo, es identificada como NE Recognition en MOSTAS[15] y como Tagger en ANONIMYTEXT[41].

Las tareas adicionales del subproceso posibilitan la ejecución de herramientas complementarias para mejorar la identificación de entidades con nombre. En las propuestas estudiadas también se contemplaba la integración de herramientas y servicios adicionales, tales como gacetillas, diccionarios de acrónimos, correctores ortográficos, etc, por tanto esta propuesta deja abierta esta posibilidad.

Tareas

Nombre	**Reconocer Entidades con Nombre**
Responsabilidades	Invoca a la herramienta NER
Entrada	Documento
Salida	Documento con Entidades con Nombre identificadas.

Nombre	Aplicar otras herramientas
Responsabilidades	Este servicio tiene la responsabilidad de manejar la interoperabilidad del sistema con fuentes externas de conocimiento.
Entrada	Documento con Entidades con Nombre identificadas.
Salida	Documento con Entidades con Nombre identificadas (con posibles mejoras en la identificación)

Nombre	Ejecutar reglas y patrones
Responsabilidades	Se aplican reglas heurísticas y patrones para identificar Entidades con Nombre adicionales.
Entrada	Documento con Entidades con Nombre identificadas.
Salida	Documento con Entidades con Nombre identificadas (con posibles mejoras en la identificación)

Subproceso Agrupar Entidades con Nombre

Las entidades con nombre identificadas son clasificadas y agrupadas en clusters, mediante algún algoritmo de agrupamiento. El objetivo es que referencias a una misma entidad sean agrupadas. Por ejemplo, "Naciones Unidas" es lo mismo que "O.N.U." El subproceso se muestra en la Figura 9.

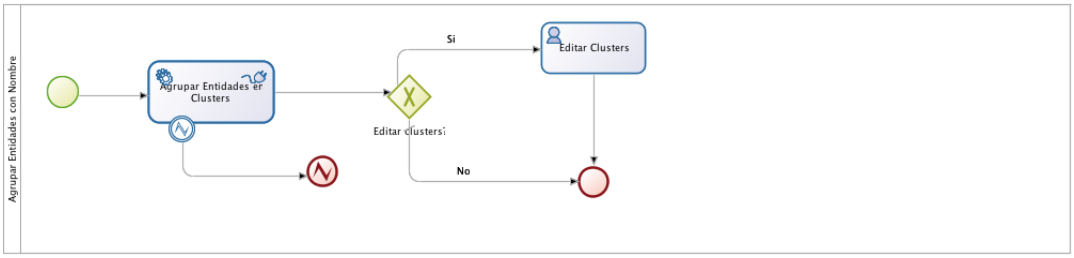

Figura 9: Subproceso Agrupar Entidades con Nombre

Tareas

Nombre	Agrupar Entidades en Clusters
Responsabilidades	Invoca a herramientas de clustering, agrupando las Entidades con Nombre equivalentes.
Entrada	Documento con Entidades con Nombre identificadas.
Salida	Documento con Entidades con Nombre identificadas y agrupadas.

Nombre	Editar Clusters
Responsabilidades	Esta tarea permite al usuario corregir y editar los grupos (clusters) identificados.
Entrada	Documento con Entidades con Nombre identificadas y agrupadas.
Salida	Documento con Entidades con Nombre identificadas y agrupadas (con posibles mejoras en la agrupación)

Subproceso Anonimizar Documento

Estando identificadas las entidades con nombre, este subproceso se encarga de la anonimización en si misma. Se identificaron distintos tipos de anonimización, parcial o total según si se anonimizan todos los tipos de entidades con nombre, reversible o irreversible si un documento anonimizado puede revertirse al original o no respectivamente. El subproceso se ilustra en la Figura 10.

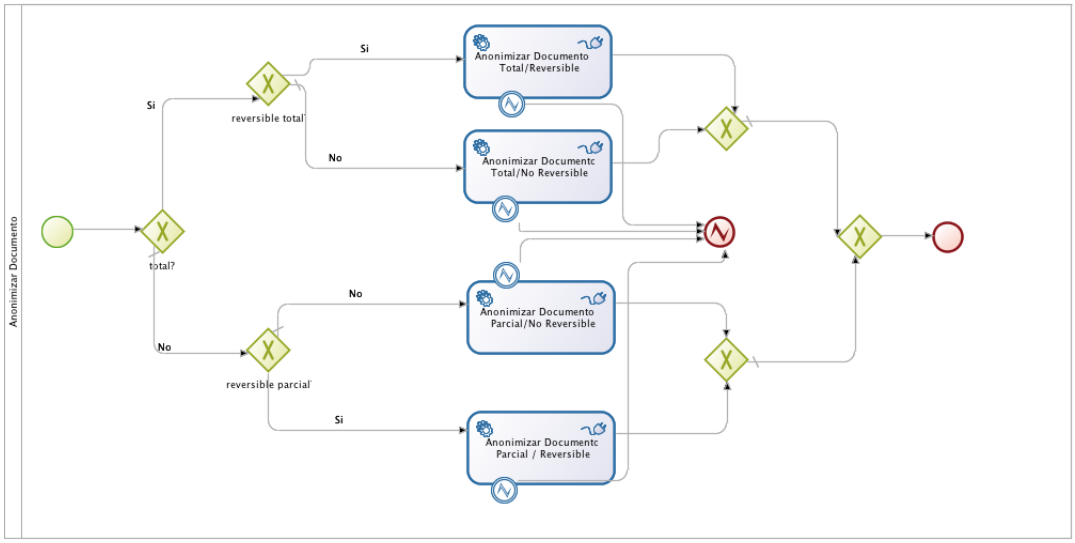

Figura 10: Subproceso Anonimizar Documento

Tareas

Nombre	Anonimizar Documento Total/Reversible
Responsabilidades	Sustituye todas las Entidades con Nombre identificadas en el documento por una referencia cifrada reversible.
Entrada	Documento con sus Entidades con Nombre identificadas.
Salida	Documento anonimizado

Nombre	Anonimizar Documento Total/No Reversible
Responsabilidades	Sustituye todas las Entidades con Nombre identificadas en el documento por un texto genérico.
Entrada	Documento con sus Entidades con Nombre identificadas y agrupadas.
Salida	Documento anonimizado

Nombre	Anonimizar Documento Parcial/No Reversible
Responsabilidades	Sustituye las Entidades con Nombre que pertenezcan a las categorías configuradas, por un texto genérico.
Entrada	Documento con sus Entidades con Nombre identificadas y agrupadas.
Salida	Documento anonimizado.

Nombre	Anonimizar Documento Parcial / Reversible
Responsabilidades	Sustituye las Entidades con Nombre que pertenezcan a las categorías configuradas, por una referencia cifrada reversible.
Entrada	Documento con sus Entidades con Nombre identificadas y agrupadas.
Salida	Documento anonimizado.

4.3. Vista de Información

Esta sección presenta la Vista de Información, mediante la cual se describe cómo se representa el modelo de datos del sistema de anonimización, y el formato en el cual fluye la información entre los distintos componentes.

4.3.1. Estructura de Datos

Para el flujo del texto a ser anonimizado dentro del sistema, se propone una estructura de datos sencilla que se aprecia en la Figura 11, consistente en una entidad que denominaremos Text que contendrá la cadena (String) conteniendo el documento en su estado actual (texto original o anonimizado), y una estructura conteniendo el conjunto de Entidades con Nombre identificadas para el documento.
La entidad con nombre se modela con una clase específica denominada NamedEntity, también especifica en la Figura 11, que tiene tres atributos:

1. term: Cadena que contiene el término o nombre. Ejemplo: “Juan Pérez”.

2. neClass: Cadena que contiene el tipo de entidad con nombre clasificada: Ejemplo: “ORGANIZATION”, “GEO_LOCATION”, “PERSON”

3. aliases: Lista de términos equivalentes a la entidad con nombre definida en term. Aquí se guardan las entidades equivalentes al agrupar las entidades (clustering)

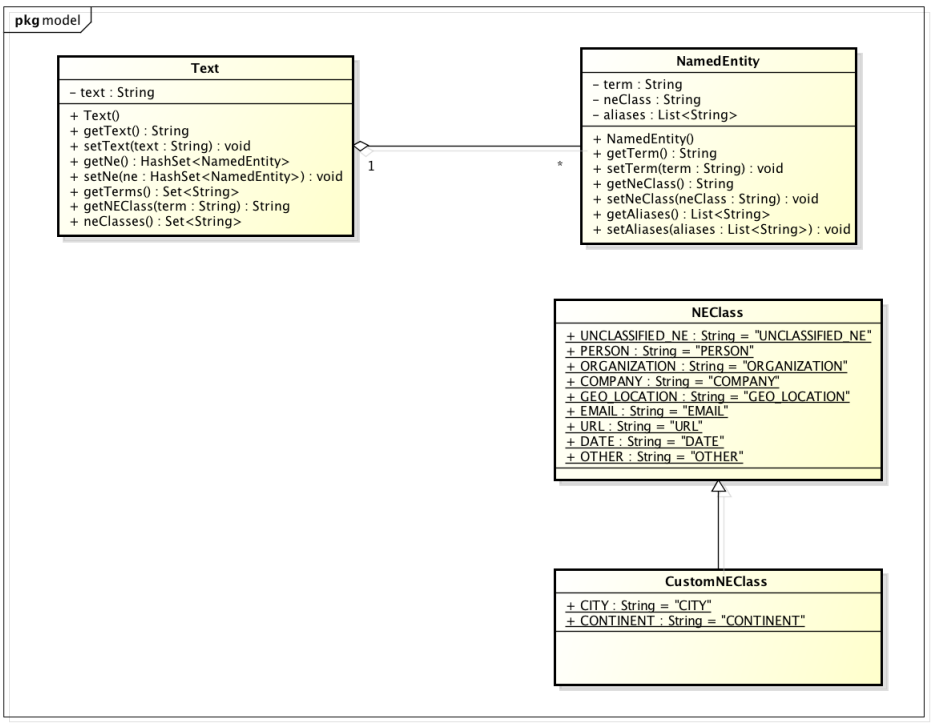

Figura 11: Modelo de Datos

4.3.2. Flujo de Datos

La información fluye en el sistema en el formato de un documento de texto presentada en la sección anterior. Si bien al haberse modelado el proceso utilizando notación BPMN2 el flujo resulta autoexplicativo, en la Figura 12 se ilustra el flujo de información mediante notación propia de dicho estándar. Allí se puede apreciar como el documento pasa por los distintos componentes del sistema de manera secuencial.

1. En primer lugar se procesa el documento (texto no estructurado) mediante el componente NER, el cual realiza la identificación de entidades con nombre primaria.

2. Seguidamente el documento es procesado mediante las reglas y patrones heurísticos que complementan al proceso NER anterior.

3. Se agrupan las entidades con nombre equivalente mediante técnicas de clustering. Se presenta el agrupamiento, y en caso de existir errores se retroalimenta al sistema y se reinicia el proceso.

4. El documento pasa al módulo Revisor, que permite a un experto validar el resultado de la identificación de entidades con nombre. En caso de no aprobación se retroalimenta al sistema y se reinicia desde el módulo NER.

5. Finalmente el documento es procesado por el módulo de Anonimización, el cual sustituye o cifra las entidades con nombre en el documento que será enviado como salida del proceso.

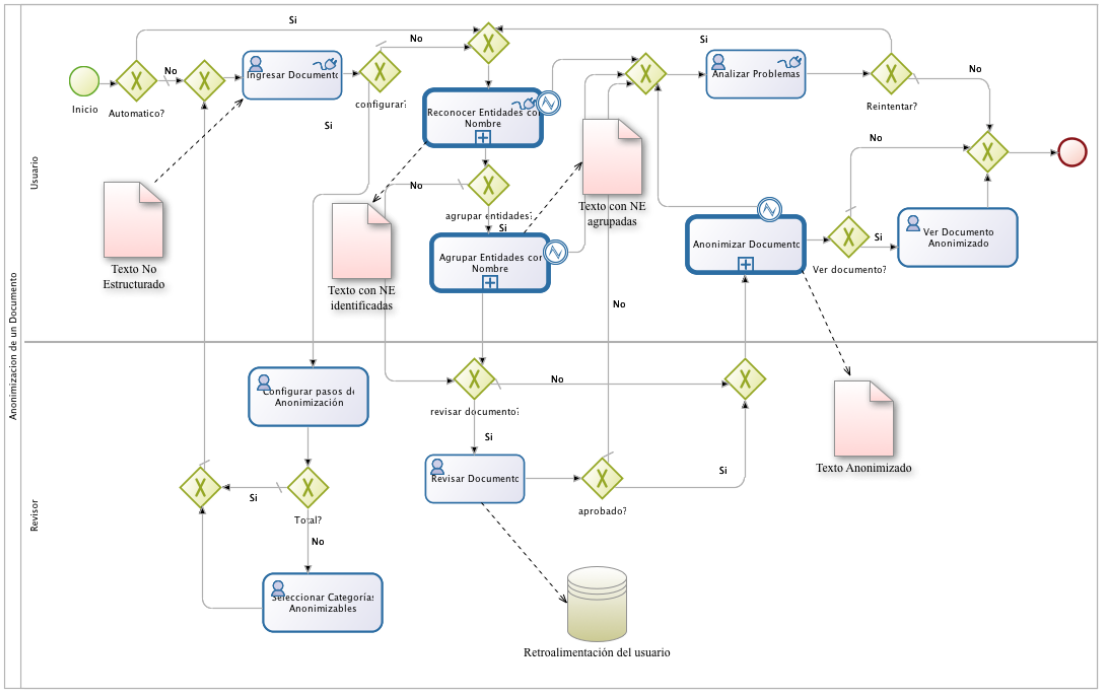

Figura 12: Flujo de información

4.4. Vista de Desarrollo

La presente Vista de Desarrollo, describe organización modular del sistema (estructura de paquetes), y los estándares de diseño que se utilizan en el sistema de anonimización.

4.4.1. Estructura de Paquetes

En la figura Estructura de Paquetes se describe una estructura de paquetes genérica para organizar los servicios que deberán ser implementados e invocados

desde el flujo de trabajo del sistema.

Como se aprecia en la figura, se proponen tres paquetes para organizar los diferentes módulos del sistema:

1. **Paquete anonimization.externaltoolwrapper:** Este paquete contiene los módulos NER, heurísticas y clustering, cada uno en un subpaquete específico. El común denominador es la interacción con herramientas externas que realizan cada uno de estos procesamientos de texto, mediante la utilización de adaptadores.

2. **Paquete anonimization.model:** En este paquete se concentra el modelo de datos que se maneja en el sistema, las clases Text, NamedEntity y NEClass, que fueran descritas en la Vista de Información precedente en este documento.

3. **Paquete anonimization.anonymizer:** Aquí se define el módulo Anonimizador del sistema, el cual será el encargado de cifrar o sustituir las entidades con nombre en el documento. Se define una clase Anonymizer, especializada por dos clases, una para representar un anonimizador reversible (ReversibleAnonimizer), que sustituye las entidades con nombre por el propio dato pero luego de procesarlo con un cifrado reversible, y un anonimizador no reversible (NonReversibleAnonymizer), el cual simplemente sustituye las entidades con nombre por información genérica.

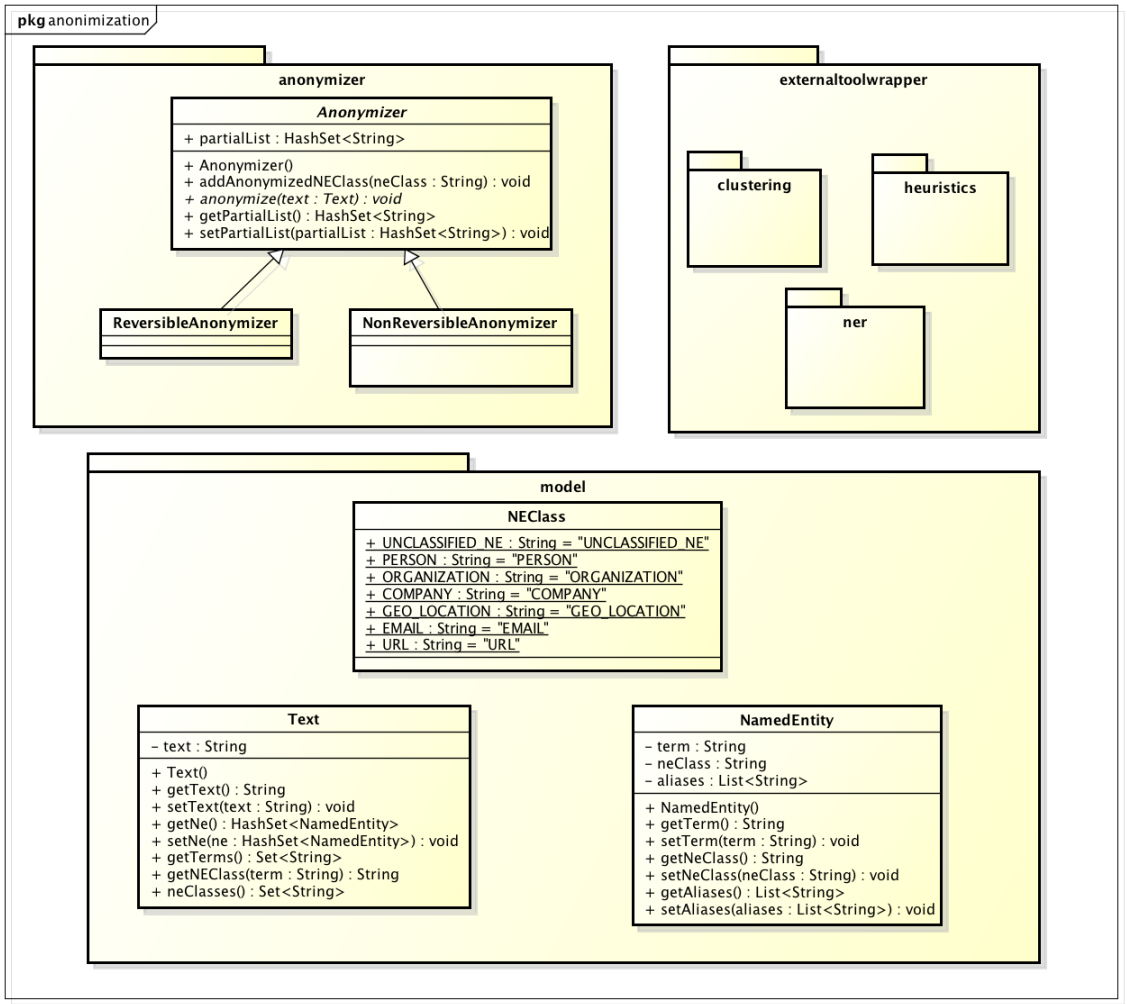

Figura 13: Estructura de Paquetes

4.4.2. Estándares de diseño

El sistema presenta algunos requerimientos clásicos, que son resueltos por aplicación de patrones de diseño. Tal estrategia es aplicada en los componentes del paquete "externaltoolwrapper.ner", donde se utiliza el patrón de diseño Adapter, como se ilustra en la Figura 14, para modelar la interacción del sistema con herramientas externas, cuyas interfaces se desea abstraer. Para ello se define una superclase con métodos abstractos que deberán ser implementados por las especializaciones.

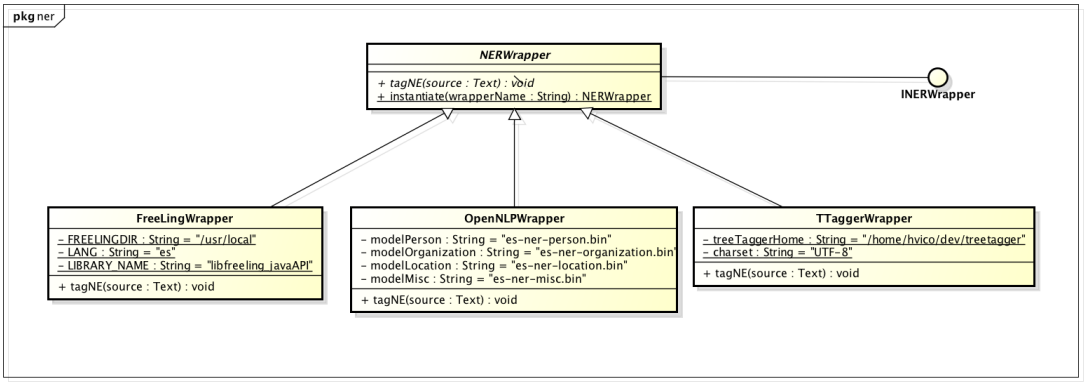

Figura 14: Patrón Adapter

Se ha resumido hasta aquí la arquitectura de referencia, cuya definición es el objetivo principal de éste trabajo de tesis. Como se explicó al comienzo del presente documento se definieron metas adicionales, y en particular un objetivo específico que consiste en poner a prueba la arquitectura definida, mediante la implementación de un sistema basado en ella para un dominio concreto (que en adelante llamaremos "Aplicación DEMO").

Antes de pasar a describir dicho sistema, se entiende pertinente resumir el parte del trabajo de investigación realizado durante el estudio del estado del arte, en relación a la instanciación tecnológica de los distintos módulos que se pueden encontrar en un sistema de anonimización. De las herramientas estudiadas se presentarán a continuación aquellas que luego fueron utilizadas para la implementación del sistema de anonimización.

5. Instanciación tecnológica de los módulos

Como se indicaba previamente, se resumirán a continuación las características que presentan algunas de las herramientas evaluadas durante el proceso de investigación del presente trabajo.

5.1. TreeTagger

TreeTagger [22] es una herramienta de uso libre para investigación, educación y evaluación. Utiliza técnicas inductivas para etiquetar texto en numerosos idiomas luego de un proceso de entrenamiento. Se proveen archivos de parámetros con los resultados de entrenamiento para una gran cantidad de idiomas, entre ellos el castellano. El etiquetador permite identificar sustantivos propios, números, verbos, códigos alfanuméricos, y decenas de otras formas lingüísticas, así como identificar el lema de las palabras. Existen numerosos "envoltorios" (wrappers) disponibles para integrar TreeTagger a software desarrollado en distintos lenguajes/plataformas como JAVA, Perl, Python, R, y Ruby.

Internamente TreeTagger utiliza técnicas basadas en árboles de decisión binarios para identificar las palabras [SCHMID1994].

Exista una interfaz gráfica (GUI) para Windows para interoperar fácilmente con TreeTagger, mediante la cual se configuran los numerosos parámetros que admite la herramienta.

5.2. FreeLing

FreeLing [30] es una suite de herramientas de análisis del lenguaje natural, de código abierto (open source) y de uso libre bajo licencia GNU. También provee archivos de datos pre-entrenados para múltiples idiomas, entre ellos el español. Algunas de las funcionalidades que provee en particular para textos en idioma español:

1. Detección de oraciones.
2. Detección de números.
3. Identificación de fechas
4. Análisis morfológico
5. Detección de frases multi-palabra.
6. Detección de entidades nombradas (NE).
7. Clasificación de entidades nombradas (NE).
8. Etiquetado PoS (part-of-speech).

FreeLing se distribuye como una biblioteca C++, y se puede integrar fácilmente a software desarrollado en otros lenguajes como JAVA a través de la API JNI. Sin embargo también se distribuye una utilidad de línea de comandos para ejecutarlo como utilitario independiente.

5.3. Apache OpenNLP

OpenNLP [20] es un proyecto de la Apache Software Foundation, y por tanto naturalmente es de uso libre y open source, que consiste en una suite de herramientas para el procesamiento del lenguaje natural basadas en el aprendizaje de máquinas. Como la mayoría de las herramientas basadas en técnicas inductivas, se requiere entrenar los distintos componentes para luego utilizarlos sobre texto nuevo. Se pueden descargar sets de datos pre-entrenados para diversos idiomas, entre ellos el español.

Algunas de las herramientas que provee OpenNLP:

1. Detección de oraciones
2. Detección de entidades nombradas (NE)
3. Etiquetado PoS (part-of-speech)

5.4. OpenCalais

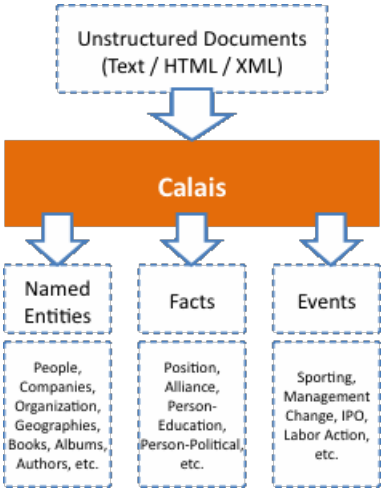

Figura 15: OpenCalais

OpenCalais [42] es una API y un procesador semántico de documentos no estructurados. Es un producto propietario parte de la línea de productos ClearForest desarrollados por la Coporación Thomson Reuters, pero para éste servicio en particular se provee acceso libre al mismo tanto para uso personal como comercial. Provee una interfaz de web services mediante la cual es posible procesar documentos no estructurados (Texto / HTML / XML), y entre otras cosas, identificar y clasificar Named Entities. Identifica personas, organizaciones, ubicaciones geográficas, libros, álbumes musicales, autores, etc. La única limitante es que la API acepta hasta 50.000 transacciones por día para cada usuario registrado, y hasta 4 transacciones por segundo. En la figura 15 se presenta un

esquema básico del funcionamiento del servicio:

Para poder probar el servicio sin necesidad de consumir el web service directamente desde una aplicación propia, se provee una utilidad llamada Calais Submission Tool que permite mediante su interfaz de usuario seleccionar documentos y enviarlos a Calais para su procesamiento. En pruebas realizadas con esta utilidad, se pudo observar una muy buena efectividad en la identificación de Named Entities. Coincide esta observación empírica con los resultados de un estudio de efectividad de sistemas NER, realizado en 2009 [34].

5.5. LingPipe

LingPipe[4] es un framework comercial desarrollado en JAVA puro. La empresa que lo desarrolla (alias-i) lo describe como "un kit de herramientas de procesamiento de texto usando computación lingúistica".

Se presenta como una biblioteca integrable a cualquier aplicación, que provee una amplia gama de servicios, entre otros:

1. Tokenización
2. Etiquetado gramatical
3. Detección de Entidades con Nombre
4. Clustering
5. Identificación de frases importantes
6. Clasificación de tópicos
7. Minería de textos para bases de datos
8. Corrector ortográfico

Como se destaca, este framework provee dos características de interés para un sistema de anonimización. Tienen capacidad de identificar Entidades con Nombre, y además de agrupar texto en clusters.

LingPipe se puede utilizar de forma gratuita con fines académicos.

Introducidas estas herramientas, se describirá a continuación el prototipo diseñado e implementado en base a la arquitectura de referencia.

6. Prototipo Aplicación DEMO: Anonimización de Jurisprudencia

En el prototipo Aplicación DEMO implementado, se puso a prueba la arquitectura de referencia presentada en la sección 4, y se integraron las herramientas introducidas en la sección 5. Para documentar este sistema se desarrolló un S.A.D. complementario a la arquitectura de referencia, el cual se incluye en el Anexo C. La presente sección resumirá algunos aspectos medulares del sistema, extraídos y resumidos a partir de dicha documentación. El prototipo además se modela además sobre la base de un dominio concreto: La anonimización de sentencias para un sistema de gestión de jurisprudencia utilizando en el Poder Judicial uruguayo. Se introducirá brevemente a continuación este contexto.

Tal como se mencionó en la subsección 2.2 al comienzo del presente documento, el Poder Judicial uruguayo cuenta con un sistema de gestión documental denominado "Base de Jurisprudencia Nacional" (BJN), el cual almacena las sentencias judiciales de Tribunales de Apelaciones y la Suprema Corte de Justicia de la República Oriental del Uruguay. Dichas sentencias son procesadas por un equipo de funcionarios que, entre otras tareas, anonimizan las sentencias para que las mismas puedan ser publicadas para acceso libre por parte de la ciudadanía. En el proceso de anonimización se oculta la información personal de las personas que se citan en la sentencia. Aquí es donde surge un requerimiento concreto de anonimización, que fue utilizado para diseñar el prototipo que se describe en la presente sección.
Mediante una interfaz el sistema Aplicación DEMO, deberá interoperar con el sistema BJN para obtener los documentos a anonimizar, y luego de procesarlos almacenarlos en la propia base de datos del sistema BJN. Como actor se visualiza a los funcionarios del Dpto. de Jurisprudencia del Poder Judicial, quienes son los encargados de realizar el proceso de anonimización de las sentencias judiciales, para que puedan ser publicadas al público general.

Tomando como referencia el diagrama de contexto (Figura 5) que se presenta en la arquitectura de referencia descrita anteriormente, se puede establecer la siguiente correspondencia:

1. El sistema de anonimización es concretamente la Aplicación Demo que se describe en el presente documento.

2. En este caso la interfaz se establece con una base documental, la base de datos del sistema BJN

3. El sistema de gestión documental es la Base de Jurisprudencia Nacional (BJN)

4. El experto del dominio son los técnicos jurídicos del Departamento de Jurisprudencia del Poder Judicial.

Se enumeran los requerimientos funcionales y no funcionales identificados para la Aplicación DEMO en la Tabla 5. Como notación se agrega la letra E a cada

requerimiento específico de la Aplicación, de manera que se diferencien de los requerimientos genéricos identificados en la arquitectura de referencia. Cuando existe una vinculación de cada requerimiento con un requerimiento genérico, se especifica la misma en la columna "Req.Arq.Ref." de la tabla.

Tabla 5: Requerimientos Aplicación DEMO

Referencia	**Req. Arq. Ref.**	**Descripción del Requerimiento**
RFE1	RF1	El sistema debe poder procesar sentencias judiciales almacenadas en el sistema Base de Jurisprudencia Nacional.
RFE2	RF3	El sistema debe permitir a los funcionarios de jurisprudencia validar las sentencias anonimizadas, y aprobar el documento o reprobarlo brindando feedback que retroalimente al sistema.
RFE3	RF7	El sistema debe almacenar el documento anonimizado en la Base de Jurisprudencia Nacional
RFE4	-	El sistema debe permitir ejecutar la anonimización de múltiples sentencias en un solo paso.
RFE5	RF2, RF4 y RF5	El sistema debe permitir, de forma configurable, un funcionamiento cien por ciento automático, partiendo de la selección de la/s sentencia/s a anonimizar, y almacenando las mismas en la Base de Jurisprudencia Nacional.
RFE6	RF6	El sistema debe permitir registrar exclusiones de términos jurídicos específicos que se presentan en las sentencias y que pueden generar falsos positivos en las herramientas NER.
RNFE1	-	Mantenibilidad: El sistema deberá desarrollarse sobre plataforma JAVA, herramienta de uso por parte de los técnicos del Poder Judicial, de forma que el sistema sea mantenible por los mismos.

Uno de los puntos destacados en la descripción de la Arquitectura de Referencia, es el modelado del funcionamiento del sistema de anonimización como un proceso de negocios. Siguiendo esta pauta, el proceso de la aplicación DEMO fue modelado utilizando el motor opensource Bonita Open Solution[5].

Uno de los puntos más importantes a destacar, es que el proceso de la Aplicación DEMO se define por sobre el proceso de Anonimización de Documentos tal cual fue diseñado en la arquitectura de referencia. Tal es así, que en primer lugar fue implementado el proceso BPMN principal definido en el SAD de la Arquitectura de Referencia, con sus tres subprocesos (Reconocer Entidades con Nombre, Agrupar Entidades con Nombre y Anonimizar Documento) sobre el motor Bonita. Una vez dicho procesos y sus correspondientes subprocesos fueron desarrollados y probados (validando de forma empírica además el proceso genérico definido), se implementó por sobre ellos un proceso adicional, que llamaremos Aplicación DEMO, el cual modela la instanciación del sistema concreta que se define en este SAD específico.

En la Figura 16 se puede visualizar el proceso Aplicación DEMO, que utiliza el proceso de la Arquitectura de Referencia como núcleo. Seguidamente se describen las entradas, salidas y responsabilidades de las distintas tareas específicas del proceso “Aplicación DEMO”.

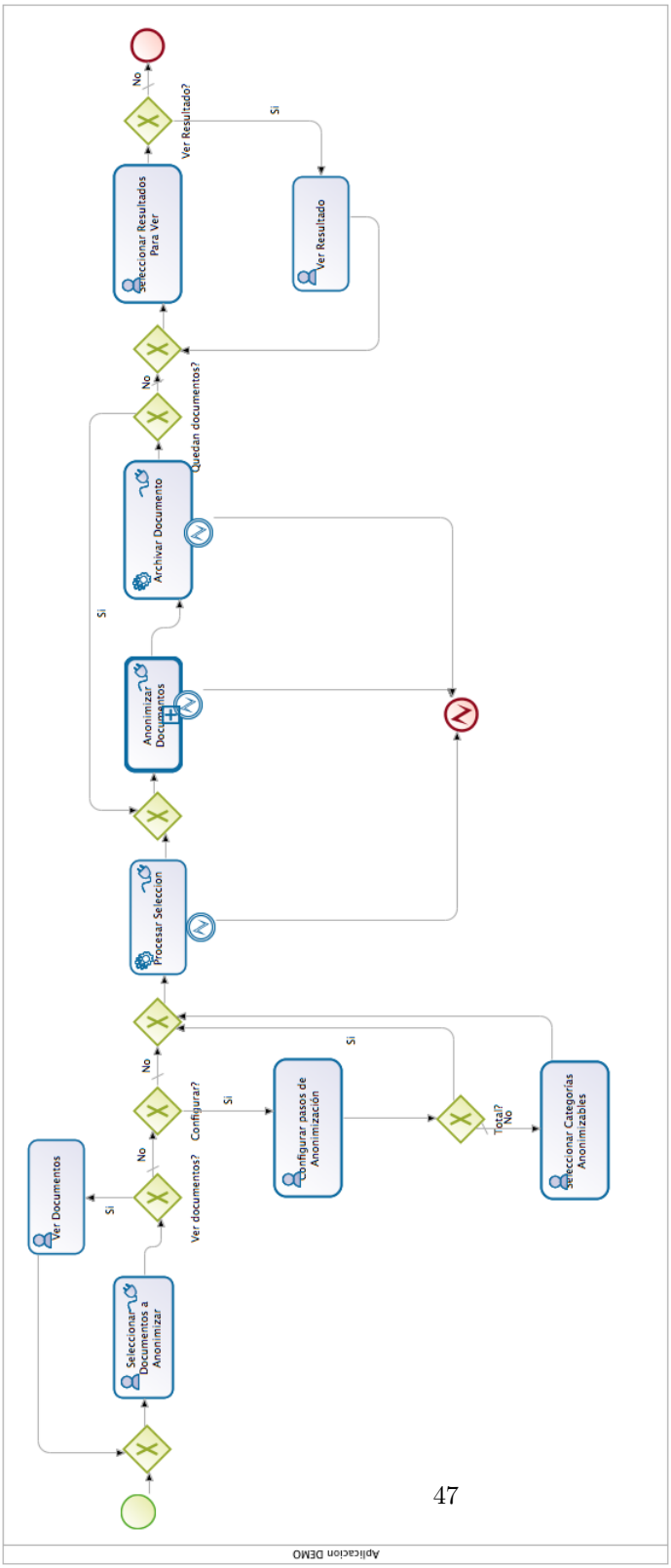

Figura 16: Proceso aplicación DEMO

Las estructuras utilizadas en el sistema para modelar el documento a anonimizar es la especificada para la Arquitectura de Referencia. En la Figura 17 se presenta un diagrama para reflejar este modelo instanciado, donde se pueden visualizar algunas adiciones al modelo abstracto presentado en la arquitectura de referencia, a saber:

1. Se definió un modelo de NEClass específico para el motor NER OpenCalais, a través de la clase OpenCalaisNEClass que se visualiza en el diagrama. Esta clase agrega algunas categorías de Entidades con Nombre que aporta el motor OpenCalais, el cual es el más vasto en este sentido de las herramientas utilizadas con este fin.

2. La clase NamedEntity adiciona algunos servicios, con el fin de simplificar entre otras cosas su agrupamiento. Destaca un método isAcronym para determinar si una Named Entitiy es acrónimo de otra.

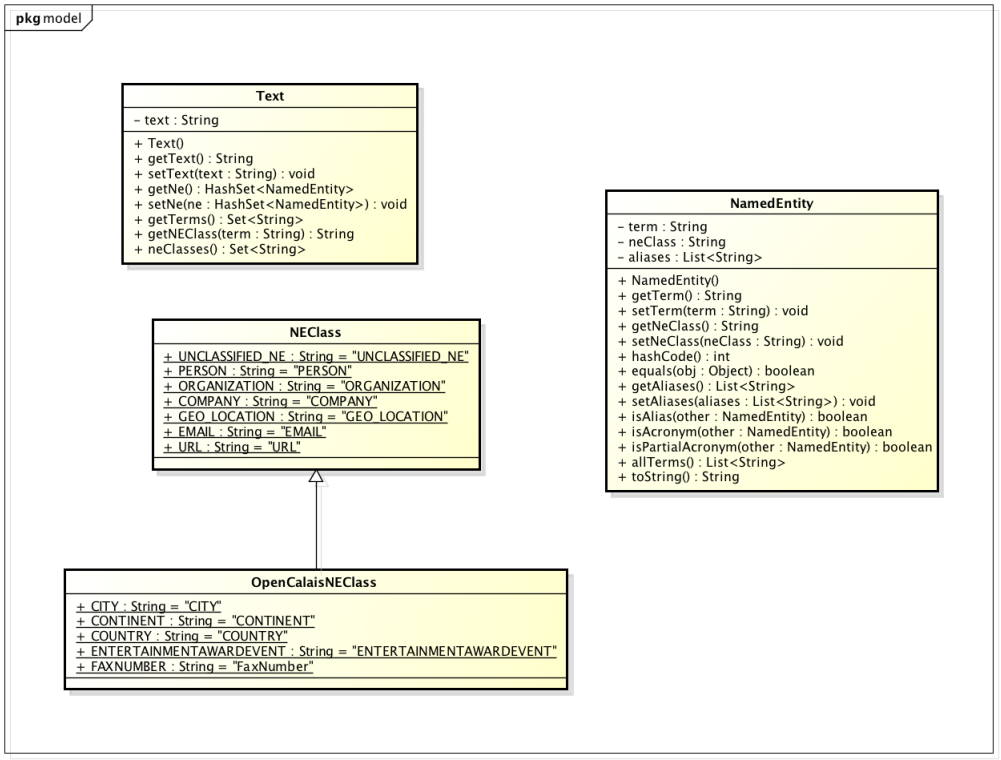

Figura 17: Modelo de Datos

Como interfaz con el sistema Base de Jurisprudencia Nacional, se definió una tabla de acceso común entre ambos sistemas sobre una base de datos MySQL[6]

5.5. La estructura de la tabla se describe en la Figura 18. De la lista de campos, los relevantes a los efectos del sistema de anonimización son los siguientes:

1. id: Clave primaria
2. texto: Contiene el texto de la sentencia (documento a anonimizar).
3. textoSensible: En este campo se almacena el texto una vez la sentencia es anonimizada.

Figura 18: Tabla Sentencia de la base BJN

El acceso a la base de datos se establece mediante la JDBC, utilizando el driver para MySQL.

Adicionalmente, para persistir las reglas heurísticas que se pueden definir dinámicamente en Aplicación Demo, se maneja una pequeña base de datos MySQL adicional llamada "Anonimizacion", donde simplemente se tiene una tabla para representar cada una de estas reglas. La estructura de la tabla se puede visualizar en la Figura 19.

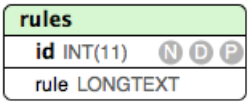

Figura 19: Tabla Rules de la base Anonimizacion

En el Anexo C se puede encontrar documentada la Vista de Despliegue del sistema Aplicación DEMO detallada. La Figura 20 resume los distintos componentes que ejecutan en Aplicación DEMO, dentro de sus respectivos contenedores de software y hardware.

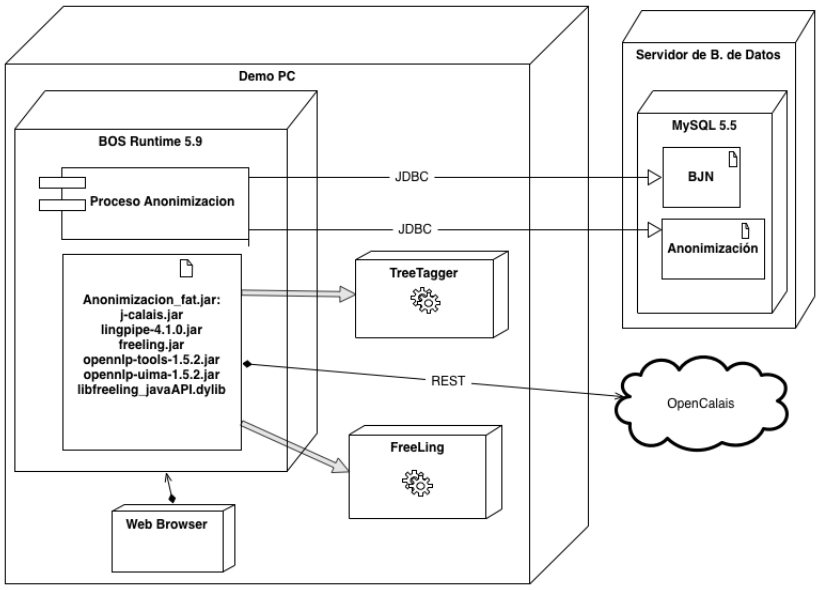

Figura 20: Modelo de Despliegue

En el sistema Aplicación DEMO fueron integradas cinco de las herramientas de procesamiento de lenguaje natural que fueron presentadas con anterioridad, sumando un adaptador original que se diseñó denominado MultiNER que se presentará más adelante. A continuación de resume brevemente las estrategias utilizadas para integrarlos, siguiendo los lineamientos de la arquitectura de referencia que define el uso del patrón de diseño "Adaptador" para facilitar dicha integración.

FreeLing

FreeLing[30] es una librería pensada para ser integrada e invocada desde otras aplicaciones, y tal es así que se distribuye con distintas APIs para integrarla en desarrollos sobre diversas plataformas, entre ellas JAVA. Es decir, no se trata una aplicación "standalone", si bien provee de una herramienta wrapper llamada analyzer que permite invocar a las distintas funciones de FreeLing desde la línea de comandos.

Al momento de utilizar FreeLing, se evaluaron entonces dos alternativas para interoperar con esta librería:

1. Una opción es invocar el wrapper de línea de comandos mencionado (analyzer), y encapsular su complejidad dentro de un wrapper propio dentro del sistema.

2. La segunda opción es utilizar la API para JAVA que provee esta librería. Esta API interactúa con FreeLing utilizando una biblioteca nativa la cual es invocada desde JAVA mediante JNI (Java Native Interface).

   Se consideró más "elegante" la segunda estrategia, es decir utilizar la API para integrar FreeLing al sistema de anonimización. Sin embargo, cabe decir que la integración de esta herramienta en particular revistió especial dificultad. Para poder utilizar la API fue necesario compilar la librería JNI nativa, para cada una de las plataformas en las que se probó el sistema. Para ejemplificar esto, para compilar esta librería en Mac OS X, fue necesario previamente compilar e instalar librerías de C que son dependencia de la misma: ICU4C, BOOST. Una vez se logra compilar la librería nativa, es necesario agregarla en CLASSPATH de la aplicación, junto con la API JAVA de FreeLing que se distribuye en formato JAR.

LingPipe

LingPipe[4] es un framework desarrollado en JAVA puro, por tal motivo su integración al sistema fue muy sencilla, consistiendo en añadir el JAR correspondiente en el CLASSPATH, así como los recursos necesarios para operar con el módulo NER (un modelo binario pre-entrenado).

En el marco de la Aplicación DEMO, LingPipe tuvo una aplicación más amplia que el resto de las herramientas, ya que para esta librería se construyeron dos adaptadores. Uno para encapsular el manejo de su módulo NER, y otro para interoperar con sus herramientas de clustering.

OpenCalais

OpenCalais[42] es una herramienta disponible en línea, la cual provee interfaces de web services tanto para SOAP como para acceso REST.

Para desarrollar el wrapper se optó en primera instancia por desarrollar un cliente REST utilizando la API JAX-WS. Sin embargo posteriormente se encontró que existía un desarrollo de un tercero llamado J-Calais, el cual provee una API JAVA que encapsula la invocación a los web services. El adaptador para OpenCalais utiliza entonces J-Calais para interoperar con los web services.

OpenNLP

OpenNLP[20], al igual que LingPipe, es un framework desarrollado cien por ciento en JAVA. Por tal motivo su integración también consistió en simplemente agregar el JAR correspondiente junto con sus recursos en el CLASSPATH.

TreeTagger

La integración de TreeTagger[22] a una aplicación JAVA es similar a la de FreeLing, con la salvedad de que no se requiere interoperar mediante JNI ni compilar módulos y dependencias adicionales. TreeTagger provee una API Java, que no hace otra cosa que encapsular las invocaciones al ejecutable de TreeTagger que tengamos instalado en el sistema. La API es básicamente un adaptador tal como el que se pretendía construir. De todas maneras, se encapsularon las particularidades de dicho adaptador dentro de un wrapper análogo a los construidos para las demás herramientas, el cual sigue la firma y estructura definida en la arquitectura de referencia.

MultiNER

El adaptador MultiNER no se trata de un adaptador para una herramienta en particular, sino que es un componente "original" del prototipo desarrollado, el cual permite integrar y utilizar en paralelo múltiples herramientas NER.

A la vista de la variable efectividad de las herramientas, surgió la idea de implementar un wrapper adicional y genérico que permitiera utilizar al mismo tiempo todas (o un subconjunto de) las herramientas NER sobre cada documento.

Este wrapper fue denominado "MultiNER", y funciona de la siguiente manera:

- Se procesa en paralelo (utilizando "threads") el documento con cada una de las herramientas disponibles o seleccionadas. Para ello se utiliza una especialización de la clase Thread de JAVA implementada en la clase "NotifyingThread".
- Las herramientas comparten una estructura de tipo "HashTable" llamada rankTable. En dicha tabla se guarda como clave la entidad con nombre identificada, y como valor la cantidad de herramientas que identificaron dicha entidad con nombre.
- Una vez finaliza la ejecución de todas las herramientas, se realiza un análisis de la tabla rankTable resultante, y se toman como válidas solamente aquellas entidades con nombre que hayan sido identificadas por un cierto número de herramientas (el umbral mínimo de herramientas para que una entidad sea considerada es configurable, aportando una vez más al atributo Configuración identificado en la arquitectura).

  La secuencia de llamadas, asincrónicas y sincrónicas se ilustra en la Figura 21.

En lo que refiere a la clasificación de las entidades en el módulo MultiNER, dado que como se explicó las herramientas presentan diversas capacidades de clasificación, cuando se determinan distintos tipos de clase por parte de distintas herramientas prevalece la primera simplemente como convención. Pero cuando una herramienta no clasifica a una entidad o la clasifica con alguna categoría

poco específica, si otra herramienta realiza una mejor clasificación se toma la de ésta.

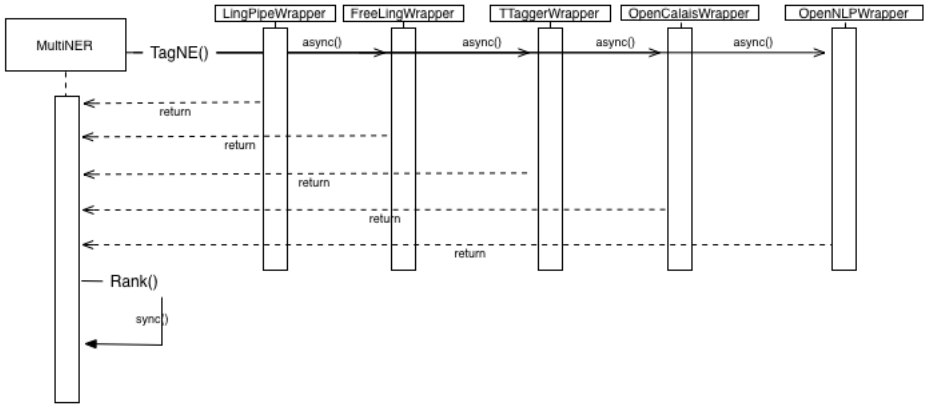

Figura 21: Diagrama de Secuencia - MultiNER

Otro aspecto a destacar en cuanto a esta implementación concreta de un sistema de anonimización, es que en Aplicación DEMO se hace un uso extensivo de introspección de tipos (reflection), una herramienta poderosa que proveen lenguajes orientados a objetos maduros como JAVA. En particular en el adaptador MultiNER que se presentará más adelante en este documento, se utiliza introspección para gestionar todo el manejo de los múltiples adaptadores que se pueden llegar a invocar desde éste módulo. Pero en general todos los adaptadores NER se manejan de forma "abstracta" mediante la superclase NERWrapper, y son instanciados en su tipo específico en tiempo de ejecución. Esto permite que los adaptadores puedan ser incorporados dinámicamente, sin necesidad de recompilar código alguno. El resultado es que se simplifica enormemente la configuración y posterior invocación desde el motor de procesos de los distintos adaptadores, con cero acoplamiento entre el motor y las herramientas externas que se terminan invocando para cada tarea particular. Análogamente los adaptadores para las herramientas de clustering, también son instanciados mediante introspección desde la superclase ClusteringWrapper.

6.1. Resultados Obtenidos

En la Figura 22 se puede apreciar la vista de la interfaz de usuario del sistema Aplicación DEMO en la primera tarea del proceso, donde permite seleccionar un conjunto de sentencias a ser anonimizadas. Si bien el foco de esta etapa del trabajo estaba acotado a implementar un prototipo de un sistema de anonimización basado en la arquitectura de referencia definida, fue posible desarrollar un sistema cien por ciento funcional que permite demostrar la viabilidad de llevar dicha arquitectura a la práctica con éxito.

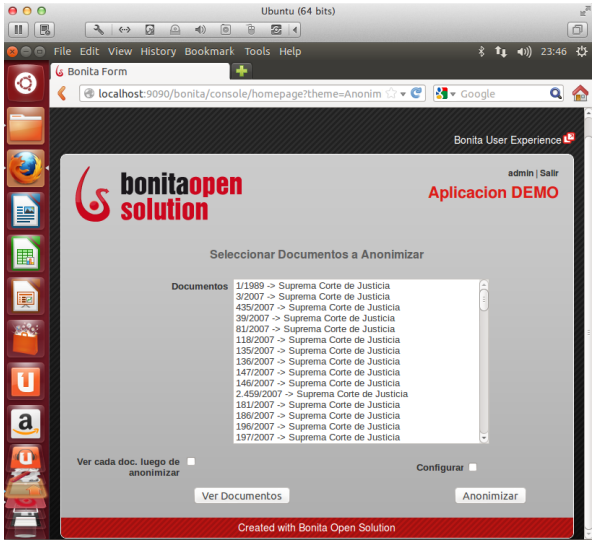

Figura 22: Sistema Aplicación Demo

Una de las decisiones de diseño de la arquitectura que agregaron particular valor a la hora de llevarla a una implementación concreta, fue el modelado de la lógica del sistema como un proceso de negocios. Esto permitió pasar del diseño del proceso abstracto directamente a un sistema ejecutable, con lo cual se redujo considerablemente el tiempo de implementación del prototipo. Además se obtuvo de forma transparente la lógica de seguridad y auditoría propia del motor de procesos, nuevamente aportando en la reducción de tiempos de desarrollo.

El prototipo desarrollado demuestra por otra parte el efecto del atributo "configurabilidad" que fue identificado como conductor de la arquitectura de referencia. Siguiendo la estrategia definida del uso del patrón de diseño adaptador y sumada la flexibilidad que aporta el uso del motor de procesos, fue posible integrar cinco herramientas diferentes que pueden ser utilizadas indistintamente para realizar el subproceso de identificación de entidades con nombre.

En cuanto a la efectividad observada empíricamente en lo que refiere a la identificación de los datos sensibles por parte de las herramientas utilizadas, si bien este no era el foco del presente trabajo, se pudo observar que para el dominio específico de las sentencias judiciales (jurisprudencia), la identificación resulta en apariencia menos efectiva que en el caso de textos más genéricos que fueron probados en etapas de estudio del estado del arte. Particularmente las sentencias contienen muchas referencias a bibliografía o autores vinculados al derecho, que pueden ser identificados erróneamente como datos sensibles, afectando la semántica del documento al ser anonimizados. También es habitual en las sentencias encontrar terminología en latín, que "confunde" a las herramientas cuando éstas fueron entrenadas sobre un corpus en español. Se pudieron observar mejores resultados utilizando el adaptador MultiNER que fue construido específicamente como idea original en este sistema.

7. Conclusiones y trabajos futuros

En esta sección se presentarán las conclusiones que derivan del presente trabajo de tesis, así como algunas ideas de trabajos futuros que se podrían desarrollar vinculados a la temática estudiada.

7.1. Conclusiones

La protección de datos personales plantea un desafío importante en las organizaciones. Este reto se ve potenciado por la aparición de legislación específica que sanciona la divulgación no autorizada de información de personas físicas y/o jurídicas en poder de las organizaciones. En nuestro país la legislación en la materia es incipiente, generando la reciente necesidad de abordar algunos problemas computacionales, tales como el que ataca la temática vinculada a este trabajo de tesis: la anonimización. La anonimización permite a las organizaciones extender el uso de información científica, técnica, o simplemente documentación de interés general en su poder, que de otra manera se haría confidencial no por su valor intrínseco, sino por las referencias a personas que pudiera contener.

Hemos visto que la anonimización abarca un amplio espectro en lo que refiere a sus marcos de aplicación, en dominios tales como las ciencias biomédicas, y resultando particularmente útil en el ámbito gubernamental.

La anonimización de documentos no estructurados, vista como un proceso computacional cien por ciento automático o aún con cierto grado de asistencia humana o de interactividad, presenta un desafío importante desde el punto de vista de la ingeniería de software.

De acuerdo al primer objetivo que se había fijado para esta tesis y como se resume en la Sección 3, se han estudiado diversas propuestas académicas que presentan arquitecturas de sistemas de anonimización, y se determinaron algunos elementos en común y otros específicos surgidos de dichos trabajos. Se investigó además la disponibilidad de herramientas libres y comerciales de procesamiento de lenguaje natural que permiten hacer frente a distintos subprocesos de un sistema de anonimización, tales como la identificación de entidades con nombre y el agrupamiento de las mismas.

En cuanto al segundo objetivo trazado oportunamente, en el presente trabajo se describe y documenta una arquitectura de referencia, que pretende recoger los elementos centrales vistos en todas las propuestas, integrando además los aspectos específicos que se consideraron más relevantes. La arquitectura presenta adicionalmente algunas características innovadoras que la diferencian de todas las propuestas estudiadas, tales como el modelado de la anonimización como un proceso de negocios descrito mediante lenguaje BPMNv2. También son clave en la propuesta arquitectural, un fuerte foco en los atributos de calidad "configurabilidad" y "adaptabilidad", pretendiendo fomentar la integración de las diversas herramientas existentes estudiadas, y otras que pudieran surgir a futuro. Esta arquitectura es resumida en la Sección 4 del presente documento y desarrollada en mayor profundidad en documentos anexos.

Definida y descrita la arquitectura de referencia, esta fue a puesta a prueba

mediante el desarrollo de un prototipo de sistema de anonimización sobre la base de dicha arquitectura, y orientado para un dominio específico: la anonimización de sentencias judiciales o jurisprudencia, lográndose de esta manera el tercer gran objetivo de las tesis que se había propuesto inicialmente. La descripción de este sistema concreto se puede visualizar en la Sección 6 de éste documento y anexos.

Es posible concluir que la arquitectura presentada es efectiva, en virtud de que fue posible trasladarla desde la propuesta genérica hacia un sistema concreto, sin desviarse del diseño de la misma. Por otra parte el modelar la anonimización como proceso de negocios, facilitó la traducción de la idea abstracta descrita en la arquitectura de referencia a la implementación específica. Pero fundamentalmente esta decisión de diseño aportó notablemente en cuanto a la flexibilidad del sistema desarrollado, y la productividad a la hora de su implementación.

La propuesta arquitectural además propone fuertemente la utilización del patrón de diseño "adaptador", con el objetivo de dar una respuesta a la necesidad de desarrollar sistemas de anonimización configurables y adaptables, que permitan integrar distintas herramientas para hacer frente a las diferentes fases o etapas que componen el proceso de anonimización.

7.2. Trabajos futuros

A continuación se describen algunas líneas de trabajo que surgen de esta tesis, que podrían abordarse a futuro como complemento a la propuesta aquí desarrollada.

- **Etiquetado morfológico como complemento a la identificación de entidades con nombre:** Si bien la identificación de la información sensible para el proceso de anonimización propuesto se centra en identificar únicamente las entidades con nombre en el texto analizado, podría pensarse como alternativa aplicar un etiquetador morfológico que categorice (etiquete) todas las palabras del texto. De esta manera podrían evitarse falsos positivos, ya que si una palabra es clasificada en cierta categoría (por ejemplo, si es etiquetada como un verbo), puede ser descartada como posible entidad con nombre con un mayor grado de certeza.

- **Integración de un corrector ortográfico como mecanismo de retroalimentación:** De forma complementaria al punto anterior, de aplicarse un etiquetado morfológico completo al texto, podría procesarse el mismo con un corrector ortográfico que analice aquellas palabras que no pudieron ser clasificadas en ninguna categoría, aumentando aún más el grado de efectividad de la clasificación, que permite identificar en última instancia las entidades con nombre. La integración de un corrector ortográfico se puede observar en la arquitectura MOSTAS [15] estudiada.

- **Retroalimentación del sistema:** Si bien el proceso de anonimización propuesto contempla la retroalimentación al sistema por parte del usuario, para mejorar la identificación de la información sensible mediante la

introducción de reglas y patrones o exclusiones, esta idea podría profundizarse permitiendo por ejemplo retroalimentar de alguna manera los propios modelos estadísticos que utilizan las herramientas de procesamiento de lenguaje natural utilizadas.

- **Entrenamiento de herramientas sobre un corpus específico**: En lo que refiere al prototipo implementado sobre el dominio específico de la jurisprudencia, surge la posibilidad de realizar un experimento adicional que resultaría de interés. La propuesta sería entrenar las herramientas utilizadas sobre un corpus compuesto por documentos jurídicos. Es de esperar que de esa manera se mejoraría la efectividad en la identificación de entidades con nombre en este contexto.

- **Integración de fuente de conocimiento específica:** También sería interesante para el dominio del sistema de anonimización de jurisprudencia, integrar alguna fuente de conocimiento específica como pudiera ser un índice de referencias o bibliográfico, que pudiera hacer frente a las limitantes vistas al utilizar el prototipo. De esa forma se podría además utilizar una herramienta externa concreta, cuya integración está prevista y soportada en el proceso definido para la arquitectura de referencia.

Referencias

[1] Cavoukian A. and El Emam K. Dispelling the myths surrounding de-identification: Anonymization remains a strong tool for protecting privacy. Information & Privacy Commissioner of Ontario, June 2011. URL http://www.ipc.on.ca/images/Resources/anonymization.pdf.

[2] Grosskopf A., Decker G., and Weske M. *The Process: Business Process Modeling using BPMN*. Meghan Kiffer Press, 2009. URL http://www.bpmn-book.com.

[3] Sánchez A. Memetracker-web: Desarrollo de una interfaz web para un sistema de monotorización de la blogosfera política. URL http://e-archivo.uc3m.es/bitstream/10016/13797/1/MemoriaPFC_Memetracker_Web.pdf. Pág. 44.

[4] Alias-I. Lingpipe, 2003. URL http://alias-i.com/lingpipe/.

[5] BonitaSoft. Bonita open solution, 2001. URL http://es.bonitasoft.com/.

[6] Oracle Corporation. Mysql, 1995. URL http://dev.mysql.com.

[7] Cáceres D. Administración de bases de datos - tesis para optar el título de ingeniero civil, 2011. URL http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/943/NUNURA_CACERES_DIANA_ADMINISTRACION_BASE_DATOS.pdf?sequence=1.

[8] Reino de España. *Ley 14/2007, de 3 de julio, de investigación biomédica*. Colección Textos legales. Ministerio de Sanidad y Consumo, 2007. ISBN 9788476706886. URL http://books.google.com.uy/books?id=eP4xQwAACAAJ.

[9] Ministerio de Justicia y Derechos Humanos. Sistema argentino de informática jurídica, 2012. URL http://www.saij.jus.gov.ar/servicios/online/tesauro.htm.

[10] República Oriental del Uruguay. Ley nº 17.930 - presupuesto nacional 2005-2009, 2007. URL http://www0.parlamento.gub.uy/leyes/AccesoTextoLey.asp?Ley=17930&Anchor=.

[11] República Oriental del Uruguay. Ley 18.335 - pacientes y usuarios de los servicios de salud, Agosto 2008. URL http://200.40.229.134/leyes/AccesoTextoLey.asp?Ley=18335&Anchor=.

[12] República Oriental del Uruguay. Ley nº 18.331: Protección de datos personales y acción de "habeas data", 2008. URL http://www0.parlamento.gub.uy/leyes/AccesoTextoLey.asp?Ley=18331.

[13] Castro E., Iglesias A., Martínez P., and Castaño L. Automatic identification of biomedical concepts in spanish-language unstructured clinical texts. In *Proceedings of the 1st ACM International Health Informatics Symposium*. ACM, 2010. ISBN 978-1-4503-0030-8. doi: 10.1145/1882992.1883106. URL http://doi.acm.org/10.1145/1882992.1883106.

[14] Real Academia Española. Diccionario en línea de la real academia española., 2012. URL http://lema.rae.es/drae/?val=anonimizar.

[15] Iglesias A. et al. Mostas: Un etiquetador morfo-semántico, anonimizador y corrector de historiales clínicos. *Procesamiento de Lenguaje Natural*, 41(0), 2008. ISSN 1989-7553. URL http://sinai.ujaen.es/sepln/ojs/ojs/index.php/pln/article/view/2581.

[16] Troyano J.A. et al. Identificación de entidades con nombre basada en modelos de markov y árboles de decisión. *Procesamiento del lenguaje natural. Nº 31*, 2003. URL http://rua.ua.es/dspace/bitstream/10045/1552/1/PLN_31_28.pdf.

[17] Baladán F. Marco legal de la historia clínica electrónica en uruguay, 2011. URL http://www.agesic.gub.uy/innovaportal/file/1652/1/marco_legal_baladan.pdf.

[18] Pla F., Molina A., and Prieto N. Evaluación de un etiquetador morfosintáctico basado en bigramas especializados para el castellano. *Procesamiento de Lenguaje Natural*, (0), 2001. ISSN 1989-7553. URL http://sinai.ujaen.es/sepln/ojs/ojs/index.php/pln/article/view/3362.

[19] Sarasola F. Ley de protección de datos personales. *Universidad ORT Uruguay*, 2009. URL http://www.ort.edu.uy/fi/pdf/florenciasarasolalicsistemasort.pdf.

[20] Apache Software Foundation. Apache opennlp, 2000. URL http://opennlp.apache.org.

[21] Schmid H. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, Manchester, UK, 1994. URL http://www.stttelkom.ac.id/staf/imd/Riset/POS%20Tagging/Using%20Decision%20Tree.pdf.

[22] Schmid H. Treetagger, 1994. URL http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/.

[23] IHTSDO. Snomed ct, 1999. URL http://www.ihtsdo.org/snomed-ct/.

[24] Div. Tecnología Informática. Base de jurisprudencia nacional pública, 12 2011. URL http://bjn.poderjudicial.gub.uy/.

[25] Gardner J. and Xiong Li. An integrated framework for de-identifying unstructured medical data. *Data Knowl. Eng.*, 68(12):1441–1451, December 2009. ISSN 0169-023X. doi: 10.1016/j.datak.2009.07.006. URL http://dx.doi.org/10.1016/j.datak.2009.07.006.

[26] Martínez Rodríguez J. Sistema de clustering de named entities. Master's thesis, Universidad Politécnica de Cataluña, 2008. URL http://www.recercat.cat/handle/2072/15414.

[27] González J.C. Anonimización: un enfoque útil para protección de la privacidad y de la confidencialidad, 2011. URL http://blog.daedalus.es/2011/06/21/anonimizacion-enfoque-util-proteccion-privacidad-confidencialidad/. DAEDALUS S.A. es la empresa comercial fundada por los creadores de la herramienta STILUS, utilizada por la arquitectura ANONIMYTEXT.

[28] Bass L., Clements P., and Kazman R. *Software Architecture in Practice*. Addison-Wesley, 2003. ISBN 9780321154958.

[29] García Moya L. Un etiquetador morfológico para el español de cuba. Master's thesis, Universidad de Oriente - Santiago de Cuba - Facultad de Matemática y Computación, 2008.

[30] Padró L. Freeling, 2003. URL http://nlp.lsi.upc.edu/freeling/.

[31] Rodríguez Yunta L. Bases de datos documentales: estructura y uso. *La información especializada en Internet - CINDOC/CSIC - Ministerio de Educación y Ciencia del Reino de España*, 2001.

[32] Poder Legislativo. Decreto 274/010, 2010. URL http://www.elderechodigital.com.uy/notas/ppla04.html.

[33] Xion Li and Gardner J. Hide (health information de-identification), 2008. URL http://code.google.com/p/hide-emory/wiki/Overview.

[34] Marrero M., Sánchez-Cuadrado S., Lara J., Morato, and Andreadakis G. Evaluation of named entity extraction systems. *Advances in Computational Linguistics. Research in Computing Science*, 41:47–58, 2009. URL http://site.cicling.org/2009/RCS-41/047-058.pdf.

[35] Rozanski N. and Woods E. *Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives*. Addison-Wesley Professional, 2005. ISBN 0321112296.

[36] United States of America. Health insurance portability and accountability act, 1996. URL http://www.gpo.gov/fdsys/pkg/PLAW-104publ191/pdf/PLAW-104publ191.pdf.

[37] OMG. Business process model and notation (bpmn) versión 2.0, 2011. URL http://www.omg.org/spec/BPMN/2.0/PDF.

[38] Clements P., Kazman R., and Klein M. *Evaluating software architectures: methods and case studies*. SEI series in software engineering. Addison-Wesley, 2001. ISBN 9780201704822. URL http://books.google.com.uy/books?id=DV917BZ9RAgC.

[39] Kruchten P. Architectural blueprints - the "4+1"view model of software architecture. Paper published in IEEE Software, 11 1995. URL http://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf.

[40] Ruch P., Baud R., Rassinoux A., Bouillon P., and Robert G. Medical document anonymization with a semantic lexicon. *Proceedings / AMIA ... Annual Symposium. AMIA Symposium*, January 2000. ISSN 1531-605X. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2244050\&tool=pmcentrez\&rendertype=abstract.

[41] Pérez-Laínez R, De Pablo-Sánchez C., and Iglesias A. *ANONIMYTEXT : Anonymization of unstructured documents*. KDIR 2009 - 1st International Conference on Knowledge Discovery and Information Retrieval, Proceedin, 2008.

[42] Thomson Reuters. Opencalais, 2008. URL http://www.opencalais.com.

[43] Meystre S., Friedlin F., , South B., Shen S., and Samore M. Automatic de-identification of textual documents in the electronic health record: a review of recent research. *BMC Medical Research Methodology*, 2010. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2923159&tool=pmcentrez&rendertype=abstract.

[44] UNE/ISO. Une/iso 15489-1 información y documentación: Gestión de documentos, 2005. URL http://gestioninfo.wikispaces.com/file/view/UNE-ISO+15489-1.pdf.

[45] Muñoz Casals V. Sistemas de gestión documental, 2007. URL http://www.monografias.com/trabajos-pdf/sistema-gestion-documental/sistema-gestion-documental.pdf.