

Generación Automática de Formularios desde Ontologías

Néstor Rocchetti, Gonzalo Labandera, Regina Motz

Instituto de Computación, Facultad de Ingeniería, Universidad de la República,
Julio Herrera y Reissig 565, 11300 Montevideo, Uruguay
{nrocchetti,rmotz}@fing.edu.uy
gonzalol1989@gmail.com

Resumen. Con la finalidad de proveer una interfaz sencilla para el ingreso de datos en ontologías se desarrolló un generador semiautomático de formularios. El generador provee dos tipos de formularios, un formulario de preguntas múltiple opción y uno para permitir ingreso de datos. Los formularios se generan en tiempo de ejecución, de esta forma los cambios que se realicen sobre la ontología impactarán automáticamente en los formularios. El generador desarrollado permite disponer de formularios web para el ingreso de datos para cualquier ontología. La aplicación es responsiva, o sea permite que pueda ser utilizada en diferentes dispositivos (e.g., computadora portátil, tablet, smart phone). En este trabajo se presenta el diseño y el prototipo del generador semiautomático de formularios desde ontologías junto con una evaluación inicial realizada sobre su efectividad con algunas ontologías de dominios diferentes.

1 Introducción

Las ontologías, como modelo conceptual de la realidad [1], brindan un nivel de abstracción adecuado para servir de comunicación entre los expertos del dominio y las aplicaciones informáticas a las que dan soporte. Incluso proveen un mecanismo de clasificación de instancias como resultado inherente de su capacidad de razonamiento. Sin embargo, el ingreso de instancias a la ontología cuando debe hacerse de forma manual no es una tarea simple. Se hace necesario saber manejar un editor de ontologías y comprender las restricciones expresadas ontológicamente sobre la definición de los conceptos que se están modelando y las relaciones que soportan. Trabajando junto con el investigador en arte Vladimir Muhvich del Proyecto Engrama [2], se plantea que investigadores en arte utilicen una ontología de percepciones de las piezas de obras de arte para disponer de un medio estructurado de ingreso de los datos de las piezas e inmediatamente de forma automática clasificarlas. Es en este escenario que surge la motivación para el desarrollo de formularios para el ingreso de instancias a la Ontología de Percepción de Piezas de Arte (OPPA) [3].

La principal ventaja para que los usuarios expertos en el dominio utilicen un formulario para el ingreso de datos a la ontología, en lugar de usar un editor de ontolo-

gías, es que disponen de una abstracción en los términos de su dominio sobre la ontología sin necesidad de conocer la estructura de clases ni la estructura de las propiedades que están implementadas en la ontología. La utilización del formulario les provee un entorno amigable y controlado de ingreso de datos evitando que los usuarios puedan realizar cambios en la estructura de la ontología y mejorando además la agilidad en el ingreso de datos.

Al proveer independencia de la aplicación sobre la ontología, se provee una capa de abstracción de la ontología al completar el formulario. De esta forma el usuario debe conocer el dominio (es decir que debe comprender el lenguaje utilizado en el contexto de las producciones del campo del arte en el caso de generar formularios a partir de la ontología OPPA) pero no es necesario que conozca el lenguaje en el que estas se implementan. En el caso del presente proyecto la ontología OPPA utiliza OWL 2 (Web Ontology Language versión 2) como lenguaje de especificación. OWL 2 está basado en Lógica Descriptiva (DL en inglés) [4], que se encuentra compuesta por una Tbox (i.e., donde se encuentran las definiciones de los conceptos), una RBox (i.e., donde se encuentran definidas las dependencias sobre los roles) y una ABox (i.e., es donde se definen aserciones sobre individuos). El entorno de desarrollo para la implementación de la ontología es el editor de ontologías Protégé4 pero para evitar la necesidad de que el usuario (en este caso el experto en arte) interactúe directamente con la ontología, se le provee un formulario de ingreso de datos guiado por la ontología.

Una característica importante de esta ontología es que tiene un alto grado de volatilidad debido a que varios conceptos del arte con nuevos medios o con soportes electrónicos de las piezas de arte no cuentan aún con definiciones estables estándares. En general, las ontologías evolucionan ya que modelan una realidad, la cual puede cambiar o se puede cambiar la descripción de la misma, es importante que estos cambios se vean reflejados en el formulario. Para esto es de utilidad que los formularios se generen a partir de la estructura definida en la ontología en la que están basados en forma automática, sin necesidad de volver a compilar código. Este enfoque permite que se pueda cambiar a una ontología en la que se represente una realidad diferente y, luego de realizar una instancia de configuración para la nueva ontología, el formulario automáticamente refleje dichos cambios.

Esto motiva entonces que la generación del formulario debe ser realizada por un generador de formularios que permita adaptar automáticamente el formulario a los cambios que pueda sufrir la ontología.

En este trabajo se presenta el diseño de un generador de formularios para ingreso de instancias a ontologías construido para los investigadores en arte. El generador se implementó en el marco de una aplicación web realizada a modo de prototipo. En el prototipo se proveen las funcionalidades de: configuración, generación de formularios y almacenamiento de sus datos. Con la finalidad de mejorar la experiencia del usuario al completar los formularios, en ciertos casos, la información se solicita por medio de preguntas expresadas en lenguaje natural tomando información del contexto del dato que se solicita de la ontología.

El resto del documento se organiza de la siguiente manera. La Sección 2 presenta un resumen de trabajos relacionados. La Sección 3 presenta el diseño del generador

de formularios para el ingreso de instancias desde ontologías y la Sección 4 detalla la implementación realizada. Finalmente, en la Sección 5 se evalúa el uso del generador con ontologías de otros dominios de interés y se dan algunas conclusiones y trabajos futuros.

2 Trabajos Relacionados

Los editores de ontologías proveen funcionalidades para creación de jerarquías de conceptos, relaciones, reglas e individuos. Algunos editores proveen funcionalidades adicionales como generación de formularios y soporte para el almacenamiento de datos en ontologías, tal es el caso de la herramienta presentada en el trabajo de Weiten *et al* [5] en el que se presenta OntoStudio, un entorno para creación de ontologías propietario. Provee una interfaz que permite diseñar formularios para ingreso de datos en base a una ontología, los formularios se construyen utilizando todos los componentes de la ontología y se desarrollan en tiempo de diseño pero para uso interno de la aplicación de edición.

Otra herramienta para generación de formularios desde ontologías es la presentada en el trabajo de Goncalves *et al* [6]. Esta herramienta, a diferencia de OntoStudio, no es un editor de ontologías, sino que utiliza ontologías como insumo para realizar sus funciones. El proceso de creación de formularios se apoya en una ontología en la que se representa el dominio de los formularios. Los formularios se especifican en la ontología y luego estos son generados a partir de esta. Para construir las preguntas se utiliza una ontología en la que se representa el dominio de la temática sobre la que se generan los formularios. Para guardar los datos ingresados se utiliza una tercera ontología, en la que se almacenan todas las aserciones creadas como resultado del ingreso de datos en los formularios generados. Por lo tanto, para generar formularios sobre un dominio específico la herramienta utiliza tres ontologías. Nuestra propuesta apunta a un proceso más ágil donde directamente de la información contenida en la ontología de dominio se genere el formulario sin necesidad de ontologías extras.

En el trabajo de Horridge *et al* [7], la especificación de los formularios es realizada por expertos en el dominio de la ontología a través de una aplicación. En la aplicación se crea la estructura de los formularios, en donde los campos de dichos formularios son clases y relaciones de la ontología. Luego de realizada la etapa de creación de los formularios, estos se instancian cada vez que un usuario lo necesite. En nuestro caso apuntamos a una generación de formularios directamente de la ontología sin ser instancias de un formulario previo para poder absorber de esta forma posibles cambios que sufra la ontología.

Con respecto a la generación de preguntas desde información presente en ontologías que representan dominios de conocimiento, en el trabajo presentado por Papalouros *et al.* [8] se describe la generación de preguntas múltiple opción a partir de ontologías. La finalidad de este trabajo es la de crear pruebas de múltiple opción para estudiantes de forma automática. La construcción de las preguntas se basa en las relaciones binarias entre las clases y en la sentencia que define a que clase pertenece un

individuo. Se apoya en una de las características más importantes de las ontologías que es que las relaciones son de tipo semánticas. Basados en este trabajo Tosic y Cubric [9] presentan una extensión para el editor Protégé [10] creado con la finalidad de construir preguntas de tipo opción múltiple para cualquier dominio de conocimiento que se encuentre representado en una ontología. Por último, en el trabajo de Al-Yahya [11] se presenta OntoQue, un sistema que crea preguntas sobre un cierto dominio de conocimiento. La finalidad de estas preguntas es que sean utilizadas por docentes para las pruebas de los estudiantes. El motor OntoQue es capaz de hacer tres tipos de preguntas diferentes: verdadero o falso, múltiple opción y completar el espacio en blanco. Para todos los tipos de preguntas se presentan tres enfoques diferentes: (i) uno se basa en una clase en particular y los individuos que pertenecen a ella, (ii) otro se basa en un individuo y las clases a las que este pertenece y (iii) el enfoque basado en propiedades que cumplen los individuos. El enfoque que consideramos relevante para nuestro proyecto es el (i), en dicho enfoque se pregunta por el tipo de una instancia, estos tipos aparecen como opciones.

3 Generador Automático de Formularios de Ontología

El formulario se crea con la finalidad de ingresar un individuo a un concepto de la ontología, a este concepto lo llamamos *concepto objetivo*. Este concepto objetivo puede tener relaciones con otros conceptos de la ontología, los cuales a su vez pueden estar relacionados con otros conceptos. Se deben entonces ingresar instancias de relaciones entre el concepto objetivo y otros conceptos ya existentes en la ontología.

Puede suceder que no todos los conceptos relacionados sean relevantes para el ingreso de datos mediante formularios. A los conceptos relevantes los denominamos *anclas*. Estas anclas definen los formularios a generar, cada uno de estos se construye con base en la jerarquía de subclases del concepto ancla.

Para definir el concepto objetivo y las anclas es necesario una instancia de configuración previa a la creación de los formularios. Para poder realizar esta instancia es necesario conocer la ontología y la realidad conceptualizada en esta. Se distinguen dos tipos de usuarios, aquellos que solo completan el formulario, y usuarios administradores que realizan la configuración. La configuración tiene como objetivo principal definir los tipos de formularios a utilizar.

3.1 Tipos de Formularios

Existen dos formas de ingresar datos a formularios: (i) ingresar un valor y (ii) seleccionar un valor de un conjunto de opciones. Ingresar un valor significa crear un individuo nuevo o asociar un dato por medio de una propiedad al concepto objetivo. Mientras que seleccionar un valor se traduce en relacionar individuos ya existentes de una clase al concepto objetivo. Estas dos formas de ingresar datos motivan la creación de dos tipos de formularios, uno para ingresar datos (lo llamamos Formulario de

Ingreso de Datos) y otro para asociar datos previamente cargados (lo llamamos Formulario Múltiple Opción). Un formulario se puede componer de estos dos tipos de subformularios.

Ejemplo Formulario Múltiple Opción. En la Fig. 1 se muestra un ejemplo de un formulario múltiple opción creado para la inserción de un individuo de PiezaExposición (i.e., concepto objetivo) en la ontología OPPA [5]. El objetivo de la pregunta es relacionar (si corresponde) un individuo de la clase Timbre ya existente, con uno del nuevo individuo de la clase PiezaExposición. En el formulario se realiza una pregunta construida mediante el uso de un algoritmo que utiliza el contexto establecido en la ontología para obtener los componentes de la frase, el contexto se compone de una clase de la ontología y una propiedad de objeto. Cada pregunta que se genera para el formulario se realiza acerca de un concepto que se encuentra relacionado mediante una propiedad de objetos con el concepto objetivo. Las opciones presentadas son los individuos que pertenecen al concepto sobre el que se está preguntando.

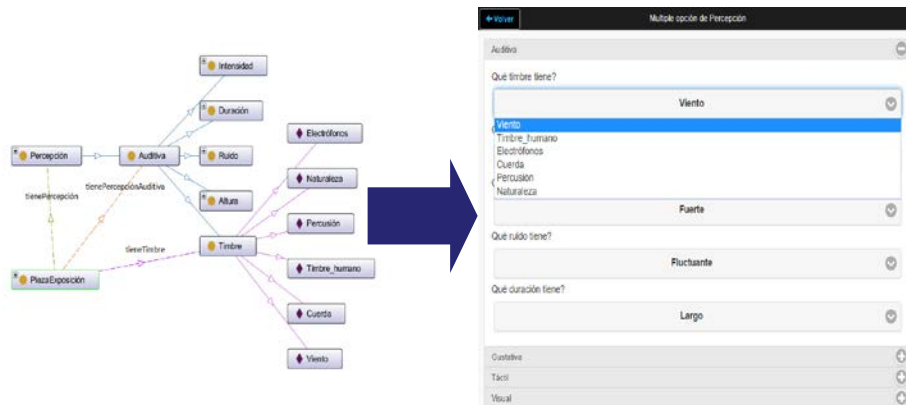


Fig. 1: Ejemplo de Formulario Múltiple Opción desde la ontología OPPA.

Ejemplo Formulario Ingreso de Datos. En el formulario de ingreso de datos, el tipo de dato que se quiera ingresar define el tipo de campo del formulario que se utilizará. En el caso de ingreso de un nuevo individuo el campo es el de ingreso de texto. En este caso en lugar de realizar una pregunta se coloca el nombre del concepto al que se le agregará el nuevo valor. En la Fig. 2 se muestra un ejemplo, el primer campo es el nombre de una nueva Pieza asociada al individuo nuevo que se va a ingresar y que pertenecerá a la clase PiezaExposición. El segundo campo también es de ingreso de valores, en este caso es un botón para tomar una fotografía de la pieza a ingresar, la fotografía es un valor asociado a la pieza.

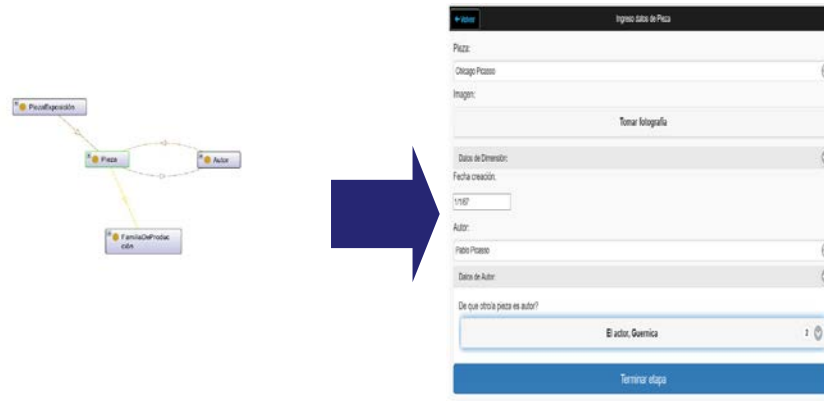


Fig. 2: Ejemplo de Formulario Ingreso de Dato desde la ontología OPPA.

El caso de las relaciones de datos es diferente debido a que el recorrido de una propiedad de datos nos puede dar mas información acerca del tipo de campo a insertar en el formulario. Los tipos de datos soportados actualmente por el prototipo son: cadena de texto, entero, booleano, fecha y URL. Es posible extender el soporte a todos los tipos de datos implementados en OWL 2.

3.2 Estructuras y algoritmos para la construcción de formularios

Los formularios se generan en función de la estructura, los conceptos y las relaciones presentes en la ontología. Para poder generar estos formularios en forma dinámica se construye una estructura auxiliar a partir de conceptos extraídos de la ontología. Esta estructura permite independizarse de la interfaz de administración de la ontología (i.e. OWL API [12]), debido a que acceder a la ontología en forma intensiva tiene como consecuencia que la aplicación disminuya su rendimiento al momento de generar los formularios. La estructura no cambia mientras que no se altere la configuración o cambie la ontología en la que se encuentra basada, esto permite que esté disponible cada vez que se crea un formulario y no se necesite acceder a la ontología.

Para la representación de clases, se realiza una correspondencia de la jerarquía de clases a una representación de árbol, cuya raíz es un concepto ancla. Cada nodo del árbol representa una clase de la ontología. A su vez, cada uno de los nodos tiene asociado propiedades de datos, propiedades de objeto, subclases e individuos. Desde cada nodo, se hace referencia a un conjunto de propiedades de objeto y propiedades de dato, de las que la clase (representada por el nodo) es dominio. El conjunto de clases asociadas al nodo son las del primer nivel de la jerarquía de subclases. En caso de que la clase representada tenga individuos, se referencia al conjunto compuesto por estos.

El algoritmo para la generación de formularios de tipo múltiple opción realiza un recorrido en profundidad de la jerarquía de clases en el grafo a partir del nodo ancla.

Si bien toda la estructura representa un grafo, la jerarquía de clases corresponde a una estructura de árbol. Se recorre cada rama del árbol de la jerarquía de clases hasta las hojas, al llegar a las hojas se genera la pregunta. Las opciones de la pregunta que se muestran son los individuos que pertenecen a la hoja.

Para representar la relación *es subclase de* se asocian todas las subclases directas a un panel, en el ejemplo de formulario web (Fig. 1) se puede observar que Timbre se encuentra dentro de un panel de nombre Auditiva, esto es debido a que, según se puede observar en la parte izquierda de la Fig. 1, la clase Timbre es subclase de Auditiva. Las clases de mayor jerarquía son representadas como el título del formulario. En el ejemplo, la clase de mayor jerarquía se llama Percepción, se puede observar que es la misma que el título del formulario. En el ejemplo entonces se ayuda a establecer que la pregunta acerca del tipo de timbre se realiza en el contexto de la percepción auditiva.

Dependiendo de si la propiedad que relaciona la clase objetivo con la clase que tiene los individuos a seleccionar es funcional o no, se permite la selección de un individuo o de varios.

La representación de las propiedades de objeto tiene asociadas clases que representan el dominio y el recorrido de la propiedad de objeto en la ontología, también tiene un atributo que indica si la propiedad de objeto representa una relación funcional. A la representación de las propiedades de datos se le asocia el nombre de la clase dominio y el tipo de dato recorrido de la propiedad.

El algoritmo propuesto para la generación del Formulario de Ingreso de dato consiste en realizar una recorrida en profundidad de la jerarquía de clases generada a partir del ancla. En el ejemplo de la Fig. 2, en la etapa de configuración (previa a la generación del formulario) se definió como concepto objetivo a PiezaExposición y como una de las anclas de ingreso de datos al concepto Pieza que se encuentra relacionado directamente con el concepto objetivo. El algoritmo de recorrido de la jerarquía de clases al llegar a un nodo hoja, crea un campo de tipo ingreso de texto que luego se mostrará en pantalla. El valor ingresado en este campo corresponde al nombre de un nuevo individuo que pertenecerá a la clase representada por el nodo hoja. Para el caso de ingreso de datos (Fig. 2) para las propiedades de datos que tienen como dominio la clase de la ontología, se utiliza el recorrido definido para saber de qué tipo es el campo a generar. Para los tipos de datos texto y entero se utiliza un tipo de campo de entrada de texto, luego si la propiedad de dato indica que este campo es entero, se controla que el valor ingresado sea efectivamente un entero. Para el caso de ingreso de una fecha, se utiliza un componente de tipo calendario, para facilitarle al usuario la selección. Luego, para el tipo booleano se permite elegir entre los valores SI y NO. El último valor diferenciado es una URL (que en la ontología toma el nombre de AnyURI), para este caso en pantalla se despliega un botón que redirige a una página especial que conecta con la cámara del dispositivo utilizado para poder sacar una foto, la foto se almacena en formato PNG en una URL accesible desde fuera de la aplicación. Luego de tomar la fotografía se vuelve al formulario principal guardando la URL de la imagen como valor de la propiedad de datos.

3.3 Almacenamiento de datos desde los formularios a la ontología

En el proceso de almacenamiento de datos, las operaciones son las de creación de nuevos individuos, inserción de nuevos individuos recientemente creados en las clases que corresponden, y el relacionamiento entre individuos según relaciones ya existentes en la ontología. No existen instancias en las que se cambia la estructura de la ontología, es decir que implican creación, modificación o eliminación de clases o de propiedades de objetos o de datos.

El primer paso del proceso de almacenamiento de datos en la ontología es la creación de un nuevo individuo y asociar dicho individuo a la clase objetivo. Se guardan además otros individuos e instancias de relaciones de objetos y de datos.

El algoritmo de almacenamiento es diferente dependiendo del tipo de formulario a guardar. Para los formularios de múltiple opción, no se insertan nuevos individuos, únicamente se crean relaciones entre individuos ya existentes dentro de la ontología. Las instancias de relaciones que se ingresan son únicamente de propiedades de objeto que tienen como dominio al concepto objetivo. En el caso de los formularios de ingreso de datos, se diferencian tres tipos de situaciones: (i) ingreso de un nuevo individuo, (ii) ingreso de un valor de una propiedad de datos, (iii) relaciones que forman ciclos. En (i), se comienza creando una nueva instancia del individuo a ingresar, pero en caso de que el individuo ya exista se obtiene dicha instancia. El individuo se crea con la finalidad de relacionarse con un individuo del concepto objetivo, por lo tanto se crea una instancia de la propiedad de objeto que los relaciona. En (ii), se ingresa un valor de una propiedad de datos, se crea directamente la instancia de la propiedad de datos para el concepto objetivo. En las ontologías se diferencia entre datos e individuos, en el caso de los datos, no es necesario crear una instancia de estos antes de crear una instancia de una relación. En (iii), los ciclos se crean con individuos ya existentes en la ontología. Se crea una instancia de la relación entre el individuo perteneciente al concepto objetivo y el individuo seleccionado de la clase en la que se genera el ciclo. No se crean nuevos individuos.

3.4 Restricciones para la Generación de Formularios

Llamamos restricciones para la generación de formularios a las condiciones que tiene que cumplir una ontología para que, a partir de esta, se puedan generar formularios con el algoritmo planteado. La principal restricción es que las propiedades de objeto que relacionan a una clase principal de una jerarquía directamente o indirectamente con la clase objetivo, deben tener el dominio y el recorrido definido para ser elegibles como anclas.

Como restricción a las propiedades de datos, estas son tenidas en cuenta si tienen un dominio definido y además alguna clase de ese dominio es parte del grafo (i.e.: estructura para la construcción de formularios). Si bien no es obligatorio que el recorrido esté definido, la definición del mismo en una propiedad de datos restringe los posibles tipos de datos a ingresar, ya que el recorrido es utilizado para saber de que tipo debe ser un campo para una propiedad de datos, esto también permite implementar controles al momento de ingresar datos en estos campos. Por ejemplo si el recorri-

do de una propiedad de dato es entero y se ingresa un valor en el campo que no es numérico se le indicará al usuario dicho error. Si no se define recorrido no se hará ningún control sobre el valor ingresado en el campo.

Los formularios pueden ser muy extensos, por tanto es importante que se formulen la menor cantidad de preguntas posible. Por esta razón se decidió formular las preguntas en las hojas de las jerarquías recorridas, por este motivo para el formulario de múltiple opción solo se mostrarán como opciones los individuos que se encuentren en dichas clases.

En el caso de formularios de ingreso de datos, el valor de las etiquetas de los ingresos de nuevos individuos es el nombre de la clase del individuo a ingresar. El caso de ingreso de datos para propiedades de datos (i.e.: atributos de instancia), el valor de la etiqueta es el del nombre de la propiedad de datos. En el caso de los campos de múltiple opción, la etiqueta asociada a dicho campo tiene como valor una pregunta. La forma de generación de preguntas se basa en la propuesta de Papasalouros [8]. La pregunta se genera en base al nombre de la propiedad de objeto que lo relaciona con la entidad objetivo y el nombre de la clase de las instancias que se presentan como opciones. Si el nombre de la clase se encuentra incluido dentro del nombre de la propiedad de objeto, este se ignora (e.g.: si la propiedad de objeto es *tieneAltura* y la clase es *Altura*, entonces la pregunta es *Qué altura tiene?*). La generación de preguntas para los ciclos con relaciones inversas es diferente, se utiliza el hecho de que se conoce para agregar la palabra *otra* (e.g.: con la propiedad *esAutorDe* y con *Pieza* como recorrido se genera la pregunta *De cuál otro/a pieza es autor?*).

Para el caso de la ontología OPPA, la forma del nombre de las relaciones se adapta al caso general de pregunta propuesta. Sin embargo no hay certeza de que el método para generar preguntas funcione correctamente en otras ontologías.

4 Prototipo del Generador Automático de Formularios de Ontologías (GAFO)

El prototipo del Generador Automático de Formularios desde Ontologías (GAFO) tiene dos funcionalidades básicas, la creación de la configuración y la generación del formulario. En la configuración se provee una interfaz para que en forma guiada se logre la definición de los parámetros necesarios para la generación de los formularios. La generación del formulario se realiza de forma automática basado en la configuración definida e incluye el posterior almacenamiento de los datos ingresados en la ontología. Además de estas funcionalidades básicas, se implementaron otras funcionalidades que involucran el manejo de usuarios y distintas formas de visualización de datos de la ontología. Entre las formas de visualización de datos, se encuentra la representación de un resumen de los datos ingresados en una tabla, y la generación de un grafo para ser visualizado en la plataforma Gephi¹.

¹ Gephi - the open graph viz platform. <https://gephi.org/>

Para el diseño arquitectónico del prototipo se utilizó un modelo de capas jerárquicas. Cada capa provee servicios a las capas superiores y requiere de servicios de las inferiores, a su vez utiliza un modelo estricto, por lo que sólo se consumen servicios de capas inmediatamente inferiores. La Fig. 3 muestra un diagrama UML de las capas del prototipo.

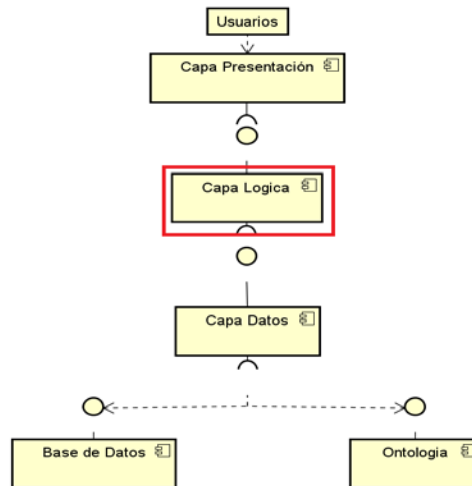


Fig. 3: Diagrama UML de diseño arquitectónico del prototipo GAFO.

La capa de presentación es la encargada de proveer una vista adecuada dependiendo si el usuario que se conecta a la aplicación tiene un perfil con rol administrador o no administrador. Por este motivo esta capa se constituye de dos componentes, en el componente Front-End se encapsula la presentación para acceder a funcionalidades de no administrador, y Back-Office donde se encapsula la presentación para acceder a funcionalidades de administrador.

La capa lógica implementa la lógica de negocio de la aplicación, proveyendo interfaces a la capa de presentación. En esta capa se implementa la gestión de usuarios y de las configuraciones. También se encuentra el componente encargado realizar operaciones sobre datos provenientes de la ontología, el cual es llamado Manejador de Ontología. Por último se encuentra el componente encargado de la generación de grafos para la aplicación Gephi, denominado Generador Gephi. Este componente consume servicios que provee el Manejador de Ontología para obtener datos de la ontología y así poder generar el grafo para Gephi.

La última capa es la de datos, en donde se gestiona la comunicación con la base de datos relacional y con la ontología. En esta capa se encuentran los componentes llamados Usuario Datos y Configuración Datos, estos se comunican con la base de datos relacional para realizar las operaciones solicitadas por la capa lógica sobre las tablas correspondientes a usuarios y configuraciones. Otro componente de la capa de datos es el Comunicador Ontología, en donde se implementan las operaciones consumidas desde la capa lógica en lo referente a datos de la ontología.

La distribución física del prototipo es la presentada en la Fig.4, esta distribución se encuentra compuesta por nodos. Un nodo es el servidor web, en donde se aloja la

capa de presentación, dicho nodo es el encargado de recibir las peticiones de usuarios, proveyendo funcionalidades para que usuarios interactúen con la aplicación. En otro nodo se encuentra el servidor de aplicaciones, el cual contiene la capa lógica, la de datos y la ontología que utiliza el prototipo. En la capa de datos se utiliza la OWL API [12] para acceder a la ontología. En el prototipo se utiliza una base de datos relacional para gestionar los usuarios y la configuración, el servidor de base de datos en donde se encuentra la base de datos utilizada se encuentra en otro nodo. El manejador de base de datos utilizado es MySQL².

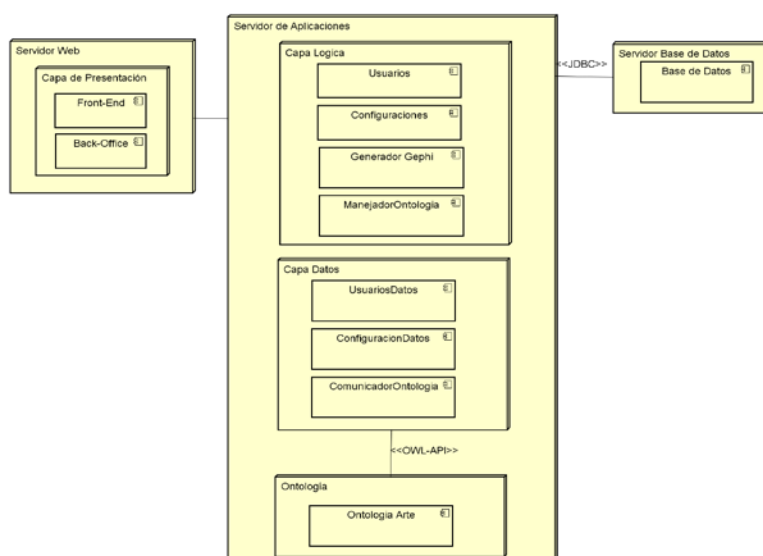


Fig 4.: Distribución del prototipo GAFO.

El Generador Automático de Formularios desde Ontologías (GAFO) es capaz de adaptarse a ontologías de diferentes dominios. Para confirmar esta afirmación se probó con una selección de la lista de ontologías disponibles en el repositorio VIVO³ y en el repositorio BioPortal⁴. Las pruebas realizadas consistieron en intercambiar la ontología y probar con un concepto que tuviera relaciones con otros de manera de crear subformularios. Algunos ejemplos representativos de ontologías que resultaron no compatibles:

MFOEM⁵ es una ontología en la que se representa el dominio de las emociones y estados de ánimo, esta ontología no resultó compatible debido a que las relaciones no tienen el dominio y el recorrido definido.

² MySQL - <https://www.mysql.com>

³ Directorio de ontologías VIVO - <http://swl.slis.indiana.edu/repository/>

⁴ Sitio online de BioPortal - <http://bioportal.bioontology.org/>

⁵ <http://www.ontobee.org/ontology/MFOEM>

IDODEN⁶ es una ontología del virus del dengue. Similar a lo ocurrido con MFOEM, no resultó compatible debido a que el dominio y el recorrido de las relaciones no se encontraba definido.

APAONTO⁷ - Es un tesoro con una lista de términos de psicología. Esta ontología no tiene relaciones, por lo tanto ningún concepto tendría anclas que agregar.

Especificación de conceptos clínicos⁸ - En esta ontología se especifican los términos clínicos y el contexto en el que se utilizan. En este caso se realizó una configuración, se generó el formulario y los resultados del mismo fueron guardados en la ontología. Sin embargo los nombres de los conceptos no eran los mismos que su identificador como nombre del campo a completar, y en el generador se muestra el identificador, entonces el contexto no se logró comprender. Para lograr que el contexto se comprenda se debería haber usado como etiqueta del campo de los formularios, en este caso, el nombre del concepto en lugar de su identificador.

5 Conclusiones y Trabajos Futuros

Concluimos que el generador GAFO es útil para casos en los que se quiera ingresar datos sobre conceptos que tienen relación con jerarquías de conceptos (e.g. PíezExposición se encuentra relacionado con Percepción, que tiene una jerarquía de conceptos). Si se tiene un concepto que tiene relaciones con otros conceptos de una ontología, pero estos no conforman una jerarquía, sucederá que la botonera principal será muy grande y se tendrá que navegar por muchas pantallas, transformando el proceso de completar el formulario en algo engorroso. De todas formas el funcionamiento del prototipo es correcto, es decir que es posible mediante el formulario generador ingresar datos a la ontología.

Un caso exitoso por ejemplo fue probar con la ontología *pizza.owl* de la Guía Práctica de Protégé-OWL [13]. El concepto *Pizza* tiene ingredientes de dos tipos, una base y una cubierta. Según la combinación de la base y la cubierta, un individuo de *pizza* es clasificado en diferentes categorías de *pizza*. La prueba con esta ontología consistió en realizar una configuración para ingresar una *pizza* y seleccionar la base y la cubierta de las existentes. El concepto objetivo en la configuración es *Pizza*, el ancla es *Ingredient* y es de tipo múltiple opción. Se elige *Ingredient* como ancla debido a que es superclase de *PizzaBase* (i.e., base de la *pizza*) y también de *PizzaTopping* (i.e., la cubierta de la *pizza*). En la Fig. 5 se muestra el formulario de múltiple opción generado con la selección de *Mozarella* como cubierta y de *Base fina* como base. Dado que el generador de preguntas se pensó para ser utilizado con ontologías en español, las preguntas tienen componentes en español.

⁶ Descargada de - <http://purl.bioontology.org/ontology/>

⁷ Descargada de <http://bioportal.bioontology.org/ontologies/APAONTO>

⁸ Clinical Building Blocks - WebProtege - <http://webprotege.stanford.edu/>

Fig.5: Formulario para la Ontología Pizzas.owl.

En el caso del formulario de múltiple opción se presentó la utilización de una estrategia basada en el contexto del concepto que provee las opciones como apoyo para la generación de una pregunta. La estrategia de generación de preguntas se desarrolló para ontologías implementadas en español, como trabajo futuro se prevee soporte a otros lenguajes.

La aplicación fue pensada en principio para ser utilizada por un grupo de reducido usuarios no administradores, expertos en el dominio. Eventualmente podría extenderse a ser utilizado por muchos usuarios, a su vez se podría utilizar el módulo de manejo de la ontología como proveedor de servicios para otras aplicaciones. A tal fin se propone mejorar la distribución física de la aplicación agregando un nuevo servidor de aplicaciones el cual contendrá una capa de servicios (para exponer los servicios que va a brindar el módulo), una capa lógica que contiene al componente Manejador de Ontologías, y una capa de datos con el Comunicador de Ontologías que mediante la OWL-API se conecta a la ontología. Esta nueva distribución independiza el modulo de gestión de datos y creación del grafo a partir de la ontología. Este módulo deja así de compartir recursos con la aplicación web, teniendo a su disposición los mismos para brindar el servicio a otras aplicaciones.

Referencias

1. Gruber, T. R. Knowledge acquisition, 5(2):199–220, 1993.
2. Muhvich., V. Engrama, investigación sobre modelos de visualización morfológica. Museo Nacional Centro de Arte Reina Sofía. Madrid, pages 249–272, 2014.
3. Rocchetti, N., Labandera, G. Generación automática de formularios para el ingreso de datos en ontologías - Aplicación en la implementación de una ontología de percepciones de piezas de arte. Proyecto final de carrera de Ingeniería en Computación. Universidad de la República, 2016.
4. Rudolph, S. Foundations of description logics. In Reasoning Web. Semantic Technologies for the Web of Data, pages 76–136. Springer, 2011.
5. Weiten, M. Ontostudio R an ontology engineering environment. Springer, 2009.
6. Goncalves, R.S., Tu, S. W., Nyulas, C. I., Tierney, M. J., Musen, M.A. Structured data acquisition with ontology-based web forms. En ICBO. 2015.
7. Horridge, M., Brandt, S., Parsia, B., Rector A.L. A domain specific ontology authoring environment for a clinical documentation system. En 27th International Symposium on Computer-Based Medical Systems (CBMS), IEEE, pages 329–334, 2014.
8. Papasalouros, A., Kanaris, K., and Kotis, K.. Automatic generation of multiple choice questions from domain ontologies. En e-Learning, pages 427–434. 2008.
9. Tasic M., Cubric, M. Sencq–protégé plugin for automatic ontology driven multiple choice question tests generation. En Procs of the 11th International Protege Conference. Stanford Center for Biomedical Informatics Research, 2009.
10. Knublauch, H., Fergerson, R.W., Noy, N. F., Musen, M. A. The protégé owl plugin: An open development environment for semantic web applications. En The Semantic Web–ISWC 2004, pages 229–243. Springer, 2004.
11. Al-Yahya, M. OntoQue: a question generation engine for educational assesment based on domain ontologies. In Advanced Learning Technologies (ICALT), pages 393–395. IEEE, 2011.
12. Matthew Horridge, Sean Bechhofer The OWL API: A Java API for OWL Ontologies Semantic Web Journal 2(1), Special Issue on Semantic Web Tools and Systems, pp. 11-21, 2011.
13. Horridge, M. , Jupp, S. , Moulton, G., Rector, A., Stevens, R., Wroe, C., and Brand, S. . A practical guide to building owl ontologies using protégé 4 and co-ode tools edition1. 3. The University of Manchester, 2011.