



# Sistema soporte de generación de evaluaciones en entornos E-Learning basado en Tecnologías de la Web Semántica

---

Proyecto Final de Carrera  
Ingeniería en sistemas de información

Mazzali, Guillermo Sebastián  
Romitti, Marcelo Fabián

Noviembre del 2013

Directoras: - Dra. Caliusco, María Laura  
- Dra. Gutierrez, María de los Milagros

## Contenido

Capítulo 1 - Introducción .....	4
1.1. Fundamentos del Proyecto .....	5
1.2. Objetivos del Proyecto .....	8
1.3. Alcances del Proyecto .....	9
1.4. Aportes principales del Proyecto .....	9
1.5. Metodología de Desarrollo del Proyecto .....	10
1.6. Organización del informe .....	12
Capítulo 2 - Marco Teórico .....	13
2.1 Noción de ontologías .....	13
2.2 Componentes de una ontología .....	15
2.3 Tipos de Ontologías .....	17
2.4 Tecnologías de la Web Semántica .....	18
Capas de la Web Semántica .....	21
Unicode .....	22
URI .....	22
XML + NS + XMLSchema .....	22
RDF + RDFSchemas .....	22
Ontology (Ontologías) .....	22
Logic (Lógica) .....	23
Proof (Pruebas) .....	23
Trust (Confianza) .....	23
Digital Signature (Firma digital) .....	23
2.5 Las redes de ontologías en las que se basó el proyecto .....	23
Red de ontología AOnet .....	25
Ontología de Instrumento .....	26
Ontologías de Evaluaciones .....	28
Descripción del Dominio de Evaluación .....	28
2.6 Conclusiones .....	30
Capítulo 3 - Análisis y Diseño del Sistema .....	31
3.1 Metodología aplicada .....	31
3.1.1 Whiteboard .....	31
3.1.2 Sprint Backlog .....	31
3.1.3 Qualities .....	31
3.1.4 Issues .....	32
3.1.5 Impediments .....	32
3.1.6 Risk .....	32
3.1.7 Next Sprint .....	33
3.2 Sprints .....	34

3.3 Modelos de Componentes.....	35
3.4 Modelo de Casos de Uso.....	36
3.5 Modelo de Clases.....	40
3.6 Modelo de Datos.....	44
3.7 Problemas surgidos sobre el trayecto del desarrollo .....	49
Problemas a nivel de diseño .....	49
Problemas a nivel de desarrollo .....	49
Problemas a nivel de pruebas.....	50
3.8 Conclusiones .....	50
Capítulo 4 - Tecnologías.....	52
4.1 Manipulación y creación de ontologías .....	52
Razonador de propósito general de JENA .....	54
Syntaxis y estructuras de reglas .....	55
4.2 Herramientas y Frameworks.....	58
Eclipse (IDE) .....	58
Maven .....	58
Hibernate .....	58
Spring .....	59
TortoiseSVN .....	59
Base de datos: MySQL .....	59
4.3 Conclusiones .....	60
Capítulo 5 - Funcionamiento del Sistema OFGA.....	61
5.1 Tabla de Referencia de Componentes.....	61
5.2 Interfaces Graficas .....	63
Acceso al sistema.....	63
Selección de Materia .....	64
Menú Principal.....	65
Administración de evaluaciones.....	68
Administración de actividades.....	71
Administración de reactivos .....	74
Administración de instrumentos .....	77
Administración de conjuntos de reglas .....	84
Administración de reglas .....	87
Capítulo 6 - Conclusiones y Perspectivas futuras .....	90
6.1 Puntos de vistas y logros obtenidos gracias a la investigación e implementación de la tesis .....	90
6.2 Perspectivas futuras de implementaciones sobre el sistema realizado.....	90
6.3 Entorno donde va a ser utilizado el software y usuarios potenciales .....	91
Referencias .....	92
Anexos .....	96

Escenarios de Casos de usos .....	96
El módulo de login .....	96
El módulo de instrumentos .....	98
El módulo de reactivos .....	103
El módulo de actividades .....	109
El módulo de evaluaciones .....	115
El módulo de reglas .....	121
El módulo de conjuntos de reglas .....	127
El módulo de validación.....	133

## Capítulo 1 - Introducción

En la actualidad, la incorporación de tecnologías de información al proceso educativo permite un mayor alcance del conocimiento a lugares donde la sociedad no contaba con acceso al mismo de manera apropiada. Con la aparición de cursos y carreras con modalidad de estudio a distancia las cosas fueron muy favorables para las personas que se encuentran en regiones alejadas de los mayores centros urbanos y que por este motivo no pueden acceder a una educación superior. El aprendizaje a distancia se está haciendo cada vez más popular en las universidades e institutos de enseñanzas, por lo que el proceso de evaluación de los cursos de estas características presenta un conjunto de retos que deben resolverse no sólo en los aspectos tecnológicos, sino también en sus concepciones y prácticas pedagógicas, modificando el proceso de enseñanza en su totalidad para dar soporte a estos nuevos procesos de evaluaciones.

Este proyecto surgió con la necesidad de asistir al docente-usuario de entornos de E-Learning en la generación semiautomática de evaluaciones válidas y confiables. Este sistema permite al docente la carga de diferentes actividades sobre los temas dictados en el curso que pretenden ser evaluados y las respuestas asociadas a las mismas. El sistema de soporte de sugerencias que se construyó, cuenta con información sobre los temas que son desarrollados y los conceptos pedagógicos que las distintas herramientas o instrumentos que pueden usarse en las evaluaciones deben cumplir. De esta manera, a medida que la evaluación es generada, el sistema le permite al docente realizar un conjunto acotado de acciones posibles sobre la evaluación que se está generando. Por ejemplo, puede inferir que un instrumento de selección múltiple contiene elementos inválidos, o que un instrumento de correspondencia no posee relaciones, sugiere agregar o eliminar pares en un ejercicio de correspondencia, agregar nuevas actividades que estén relacionadas a temas que aún no fueron evaluados y que son parte del objetivo de la evaluación, entre otras alusiones.

El sistema desarrollado en este proyecto consiste en un software donde se facilita la validación semiautomática de evaluaciones de acuerdo a distintos dominios de conocimientos. Este sistema cuenta con un conjunto de reglas establecidas para cada uno de los tipos de evaluaciones, lo que permite verificar si las evaluaciones cumplen con los principios pedagógicos, brindándole la posibilidad al docente de cargar nuevas reglas o de quitar reglas que no correspondan a sus criterios.

Este sistema brinda información oportuna que el usuario podrá utilizar para tomar mejores decisiones en la tarea que está llevando a cabo. También, cuenta con un mecanismo que le permite filtrar la información que se encuentra disponible para sólo mostrar la que es de interés para el usuario. Para diseñar este tipo de sistemas es importante determinar el dominio sobre el cuál el sistema realizará la recomendación (por ejemplo, películas, emails, e-commerce, etc.).

## 1.1. Fundamentos del Proyecto

Con las nuevas metodologías de enseñanza soportadas por las nuevas tecnologías de información (denominadas E-Learning), se generan nuevos paradigmas en el sistema educativo, donde éstos pueden implementarse a distintos niveles en situaciones donde las personas que enseñan y los alumnos se encuentran separados físicamente por una larga distancia y sólo llegan a encontrarse personalmente de manera ocasional.

Como una parte importante de estos cambios de paradigmas, la Web actual se constituye como un universo virtual en constante cambio, exigiendo que el sistema educativo se vaya adaptando todo el tiempo a ésta, creando nuevas estructuras didácticas incorporadas a estos nuevos entornos tecnológicos. (Gama, 2005).

Desde un aspecto pedagógico, la enseñanza en ambientes dispersos, como el aprendizaje a distancia (E-Learning), requiere de una amplia personalización y eficacia en la orientación de los alumnos que acceden al mismo, ya que las personas que ingresan a las plataformas de aprendizajes provienen de grupos poblacionales que poseen una amplia variedad de aspectos, ya sean culturales, sociales, u otros, produciéndose un entorno de individuos heterogéneo, donde hay que saber resolver de una manera flexible cualquier cambio que se proyecte sobre el ambiente de aprendizaje (Guardia Ortiz, 2005). Generalmente, el proceso de evaluación se realiza mediante consignas que permiten valorar diferentes niveles de habilidad de los alumnos definiendo una taxonomía de 6 niveles de comportamiento intelectual. Estos niveles son: Evaluación, Síntesis, Análisis, Aplicación, Comprensión y Conocimiento (Bloom, 1956).

Por otro lado, en lo que respecta al ambiente tecnológico, el uso de tecnologías de la web semántica en E-Learning se fue incrementando de manera notable en los últimos años. Las tecnologías semánticas pueden utilizarse como base para la construcción de sistemas de aprendizajes, ya que las mismas cuentan con todas las herramientas que la modalidad

educativa requiere, como son la conceptualización de datos basadas en ontologías, la estandarización de los componentes para compartir información y la composición en cursos educativos con entregas proactivas del material de enseñanza a través de LMS (Learning Management System) (**Chung y colab., 2003**).

En el dominio del E-Learning, las evaluaciones o exámenes que desarrolla el docente para evaluar a los alumnos son un campo de conflictos permanente, donde el docente deberá tener el cuidado suficiente para desarrollar una evaluación que sea válida y confiable (**Ashton, 2008**). Una evaluación será válida si alcanza los objetivos que el desarrollador propuso. Generalmente, cuando se plantea la necesidad de una evaluación, ésta tiene como objetivo determinar si el alumno alcanzó o no la comprensión de ciertos temas que fueron desarrollados. De esta manera, se dirá que una evaluación es válida si evalúa los temas que el docente propone como objetivos. Por otro lado se dice que una evaluación es confiable cuando persigue ciertos objetivos pedagógicos. Algunos de estos objetivos están relacionados con el entendimiento de la consigna por parte del alumno, el tipo de instrumento usado para realizar la evaluación y condiciones que deben cumplirse entre los elementos utilizados para conformar la evaluación.

Esto favorecerá a todas las partes involucradas, como pueden ser los alumnos, los docentes, miembros de la gestión institucional y el sector administrativo. Estos beneficios se ven en la reducción de recursos y tiempo en las tareas involucradas en la evaluación del conocimiento y la mejora en la performance del alumno.

En el contexto de E-Learning, las ontologías y las tecnologías de web semántica son extensamente usadas con diferentes propósitos. **Chan y Chen (2009)** propusieron una herramienta para la evaluación por pares para satisfacer las necesidades de aprendizaje cooperativo en un entorno E-Learning. **Gladun (2009)**, propuso un dominio de ontología para evaluar las habilidades de los alumnos. En este caso, el dominio de ontologías no es el único instrumento de aprendizaje sino también un medio para la prueba y la enseñanza. También, en literatura, existen diferentes enfoques que definen ontología como una estructura que permite guiar el diseño automatizado de evaluaciones, estas pueden encontrarse en **Catellanos–Nieves, 2011; Cubric y Tosic (2010)**. Sin embargo, la mayoría de estos enfoques se basan en ontologías individuales que son modelo de una parte del dominio de evaluación teniendo en cuenta sólo los aspectos estructurales de las evaluaciones. Por ejemplo, en

**Castellanos-Nieves (2011)**, los autores han definido una ontología para el apoyo a la generación de preguntas abiertas, mientras que en **Cubric y Tosic (2010)**, los autores sólo modelan preguntas tipo Verdadero y Falso. Relacionadas con la evaluación en el entorno de e-Learning, **Frutos - Morales (2010)** proponen la utilización de las tecnologías de la web semántica para generar información semántica en el proceso de Teaching Learning.

Las propuestas descriptas anteriormente sólo se enfocan en un tipo de evaluaciones que se basa en un nivel de la *taxonomía de Bloom* (**Bloom, 1956**) sin considerar los demás puntos de vista. Nos focalizaremos en los tipos de preguntas que van a utilizar los docentes dentro del sistema, considerando así, por ejemplo, preguntas de opciones múltiples, cuya respuesta es verdadera o falsa, como un tipo especial de estas últimas. Por otro lado, también, incorporan interpelaciones semánticas del mapeo entre una ontología de dominio y la ontología que da soporte a la estructura general de las evaluaciones (**Cubric y colab., 2010**).

Con el objetivo de educar de una manera más sofisticada se implementa el concepto de ontología en el diseño de la plataforma de enseñanza, ya que actualmente la existencia de herramientas para la creación, mantenimiento e intercambio de ontologías hace posible la utilización de estas en la evaluación de la enseñanza en ambientes colaborativos.

Las ontologías pueden utilizarse con diferentes propósitos:

- Para capturar la estructura de un dominio.
- Para capturar la representación de un dominio por parte de expertos.
- Para codificar y enlazar el contenido a una estructura de dominio.
- Para marcar un mapa de conocimiento.
- Para empaquetar y entregar contenido con diferente granularidad.
- Para ser parte de un sistema recomendador.
- Para proporcionar una estructura para guiar el diseño automatizado de una evaluación (**Castellanos-Nieves, 2011**).

La utilización de las ontologías hoy en día dentro de la Web semántica es una forma nueva de diseñar y construir modelos conceptuales basados en algún dominio específico.

En el trabajo de tesis desarrollado por la Mg. Lucila Romero se definió una red de ontologías denominada AONet que facilita la generación semiautomática de evaluaciones



basado en dominios de conocimiento arbitrario (**Romero y col, 2012**). Una de las ontologías que conforman la red, es la de instrumentos llamada *Instrument* la cual tiene definidas un conjunto de reglas lógicas que restringen la manera de usar los distintos instrumentos de acuerdo a principios pedagógicos establecidos por **Bolívar (2011)**.

Una red de ontologías es un conjunto de ontologías en las cuales se explicitan las meta-relaciones (**Suárez-Figueroa, 2010**). El principal beneficio de su uso es que conceptualiza un dominio dado de una manera modular. Una ontología en red es lo suficientemente pequeña como para ser comprensible por cualquier persona y su mantenimiento es fácil. Además, el hecho de decir que la red de ontología está compuesta por varias ontologías, permite a los diseñadores trabajar en paralelo sobre la red.

En conclusión esta herramienta facilitará la creación semiautomática de evaluaciones y se podrá lograr una optimización de la misma, para evaluar a los alumnos y su corrección.

## 1.2. Objetivos del Proyecto

El objetivo general del proyecto es Diseñar y Construir un Prototipo de Sistema basado en tecnologías de la Web Semántica para dar soporte a la generación semiautomática de evaluaciones válidas y confiables en entornos de E-Learning.

Dicho sistema fue construido sobre la base de una red de ontologías como modelo conceptual del dominio y reglas lógicas para evaluar la validez y confiabilidad de la evaluación generada.

Para alcanzar dicho objetivo, se propusieron los siguientes objetivos específicos:

- Estudio e Investigación sobre entornos de E-Learning, sistemas validadores de reglas y tecnologías de la web semántica.
- Analizar los requerimientos del sistema en entornos de E-Learning para la generación semiautomática de evaluaciones.
- Identificar usuarios potenciales y definir roles para los mismos.
- Diseñar un sistema para la generación de evaluaciones por parte de los usuarios teniendo en cuenta la ontología subyacente a ser utilizada y las tecnologías de la web semántica.

- Diseñar un sistema basado en tecnología de agentes inteligentes que detecte problemas en la creación de las evaluaciones y sea capaz de proponer soluciones de acuerdo a conceptos pedagógicos previamente estipulados.
- Implementar y probar los sistemas diseñados generando casos de prueba.

### 1.3. Alcances del Proyecto

Luego de una etapa en la que predominaron las actividades de investigación, planteada en la planificación del presente proyecto, se logra obtener un mejor conocimiento de los alcances del mismo.

El sistema desarrollado brinda soporte a la creación de evaluaciones, y la validación de las mismas a través de un conjunto de reglas pedagógicas establecidas de manera de favorecer el entendimiento por parte de los alumnos.

La herramienta creada, presenta como funcionalidades principales la gestión de actividades, reactivos e instrumentos que forman una evaluación y la selección de reglas propias a las actividades para examinar la composición y estructura de la evaluación, como así también respuestas posibles de todas las actividades agregadas por el profesional y su adecuación a principios pedagógicos.

La implementación de esta herramienta brinda ventajas en el área de educación, más específicamente en escuelas, colegios, terciarios y facultades, permitiendo a los profesores automatizar la creación de sus evaluaciones de acuerdo a ciertas reglas preestablecidas por los mismos. Además, los alumnos se sentirán más cómodos a la hora de realizar dicha evaluación ya que se encontrarán con exámenes válidos con una mínima cantidad de errores, evitando confusiones que posiblemente los lleven a malas interpretaciones.

### 1.4. Aportes principales del Proyecto

Este Proyecto Final de Carrera surgió de una investigación llevada a cabo por integrantes del centro de investigación del CIDISI, quienes proponen evaluar y examinar metodologías sintácticas y semánticas de evaluaciones, a través de la implementación de reglas, permitiendo así crear exámenes pedagógicamente correctos, recomendando las mejores prácticas estructurales a la hora de diseñar las evaluaciones.

Los resultados de este proyecto se manifiestan de las siguientes maneras:

- Aportar una herramienta concreta de un sistema de software de uso libre para generar evaluaciones válidas, brindando recomendaciones desde el punto de vista pedagógico en la construcción de las mismas.

- Colaborar con los usuarios, esto es, docentes de diferentes materias, de cualquier nivel y ámbito educativo.

- Mejorar el intercambio de información entre docentes y alumnos.

- Promover el uso de herramientas y técnicas para el manejo de ontologías.

Desde un punto de vista de estudiantes de ingeniería en sistemas de información y futuros profesionales, consideramos a este proyecto como una herramienta de aprendizaje y experiencia en nuevos conocimientos, permitiéndonos afianzar los temas adquiridos a lo largo de la carrera, como así también profundizar nuestra formación como profesionales.

Desde el punto de vista profesional, este proyecto brinda una herramienta que da soporte a los educadores en la construcción de evaluaciones haciendo recomendaciones sobre aspectos pedagógicos a tener en cuenta relacionados a la conformación de las mismas. De esta manera se asiste al educador que no tiene una base pedagógica fuerte, en las mejores prácticas usadas por expertos.

La utilización de ontologías enriquece nuestros conocimientos como futuros ingenieros en sistemas, ya que al no poseer experiencia en esta metodología, este proyecto nos brindó los conceptos base para poder implementar dicha metodología en distintos dominios de aplicación.

## 1.5. Metodología de Desarrollo del Proyecto

La metodología que se implementó en este proyecto, contempló las siguientes actividades:

- Estudio e Investigación sobre entornos de E-Learning, sistemas validadores de reglas y tecnologías de la web semántica.
- Análisis de los requerimientos de un sistema en entornos de E-Learning para la generación semiautomática de evaluaciones.
- Identificación de usuarios potenciales y descripción de roles para los mismos.

- Diseño de un sistema para la generación de evaluaciones por parte de los usuarios teniendo en cuenta la ontología subyacente a ser utilizada y las tecnologías de la web semántica.
- Diseño de un sistema que detecte problemas en la creación de las evaluaciones de acuerdo a conceptos pedagógicos previamente estipulados alertando de los mismos.
- Implementación y seguimiento de los sistemas diseñados, generando casos de prueba.

La metodología de trabajo en el desarrollo de este proyecto consistió de tres grandes etapas:

1. *Estudio e investigación:* En esta etapa comenzamos con la investigación del uso de ontologías, instanciación de las mismas, utilización de entornos de E-Learning en distintos ámbitos académicos. Realizamos investigación de herramientas y lenguajes para implementar ontologías y reglas lógicas que permitan derivar nuevo conocimiento, y nos familiarizamos con los entornos de desarrollo Protege 3.4.8 e IDE Eclipse. También analizamos la manera más óptima de almacenar en una base de datos las instancias de las ontologías correspondientes.
2. *Diseño e implementación del sistema:* Para la implementación de este proyecto decidimos utilizar una metodología ágil, particularmente se adoptó SCRUM, por lo que primeramente se recolectó un conjunto inicial de requerimientos funcionales y no funcionales para comenzar con una primera iteración para el desarrollo de la aplicación. Luego, definimos las interfaces gráficas que forman parte de la aplicación y son amigables para el usuario. Una vez comenzada la construcción de la aplicación, se detallaron todos los diagramas y estructuras que contemplaban el desarrollo y que permiten cubrir los requerimientos previamente establecidos. Además, se realizó el almacenamiento de instancias de ontologías en base de datos.
3. *Pruebas de prototipos:* Realizamos como primera instancias las pruebas de unidad para verificar que los módulos del prototipo funcionan correctamente. Luego llevamos a cabo pruebas de integración y pruebas sobre escenarios reales.

A partir de la captura de los requerimientos se comenzó a redactar la documentación pertinente al desarrollo del proyecto, detallando el alcance del sistema, sus requerimientos,

diagramas de desarrollo e implementación, como así también los resultados obtenidos durante las pruebas realizadas.

Cabe destacar que estas etapas se enmarcan dentro de un ciclo iterativo e incremental, donde cada ciclo puede quedar definido por varias de estas etapas dependiendo de los resultados obtenidos o las modificaciones y revisiones necesarias.

## 1.6. Organización del informe

En el Capítulo 2 se describe la red de ontologías “*AONet Ontology Network*”, detallando los elementos que la componen como así también las nociones de los conceptos que son considerados necesarios para su entendimiento.

En el capítulo 3, nos centraremos en el análisis y diseño del sistema. Haciendo hincapié en la metodología utilizada, y de qué manera se la llevó a cabo.

El Capítulo 4 describe todas las herramientas utilizadas para el desarrollo del sistema. Además, se describirán las dificultades que surgieron durante el desarrollo de la herramienta, las soluciones que se consideraron para resolverlas y las decisiones tomadas en cada caso, con su correspondiente justificación.

En el Capítulo 5 se describe la funcionalidad del sistema a través de cada una de las pantallas que forman parte de la interfaz de usuario del mismo, mencionando los componentes que las conforman. Esta descripción se realiza a través de un ejemplo real.

Finalmente, el Capítulo 6 describe las conclusiones de este trabajo, lo cual incluye los resultados obtenidos de las tareas de investigación y desarrollo llevadas adelante, junto con los trabajos futuros que podrían desarrollarse tomando como base la propuesta desarrollada en el presente Proyecto Final de Carrera.

## Capítulo 2 - Marco Teórico

El propósito de este capítulo es describir la red de ontologías AONet, acrónimo de Assessment Ontology Network, desarrollada por la Magister Lucila Romero en su tesis de doctorado, el cual permite guiar al usuario en la definición de una evaluación.

Debido a que AONet se construyó utilizando un conjunto de ontologías y tecnologías para su implementación, como los lenguajes de representación y de consulta, entornos de desarrollo y edición de ontologías; primeramente se hará una breve introducción a los conceptos necesarios para la comprensión de las mismas.

Luego, se detallará cada una de las ontologías que componen la red de ontologías AONet y cómo las mismas están interrelacionadas.

### 2.1 Noción de ontologías

La palabra ontología se deriva del griego *onto* (estudio del ser) y *logos* (palabra). Desde el punto de vista filosófico, ontología es una ciencia, es una explicación sistemática de la existencia, de los tipos de estructuras, categorías de objetos, propiedades, eventos, procesos y relaciones en cada área de la realidad. Una definición de Ontología es la siguiente (**Pierce 1935**):

*“Ontología es el estudio de las características más generales de la realidad y de los objetos reales”.*

Desde el punto de vista de la ciencia de la computación, en el área de la inteligencia artificial, una ontología, puede definirse como sigue (**Gomez-Pérez y colab., 2004**):

*“Una ontología es una especificación explícita de una conceptualización, es decir, proporciona una estructura y contenidos de forma explícita, que codifica las reglas implícitas de una parte de la realidad, independientemente del fin y del dominio de la aplicación en el que se usaron o reutilizaran sus definiciones.”*

Desde una perspectiva pragmática, una ontología se puede definir como una herramienta de representación sobre la base de cuatro tipos de componentes de modelado: conceptos, roles, restricciones e individuos.

*Concepto* representa las clases de objetos. *Roles* describen relaciones binarias entre los conceptos, por lo que también permiten la descripción de las propiedades de los conceptos. *Restricciones* se utilizan para expresar las propiedades de los roles, como ser la cardinalidad. Los *individuos* que representan instancias de clases, por ejemplo objetos. Además, es posible usar axiomas y reglas para inferir nueva información. Los axiomas son oraciones lógicas siempre verdaderas, que expresan las propiedades del modelo. Las reglas son sentencias lógicas que expresan características del dominio, por ejemplo, las reglas de negocio.

En la comunidad ontológica, las ontologías pueden ser clasificadas como livianas o pesadas. Una ontología liviana es una ontología simplemente sobre la base de una jerarquía de conceptos y una jerarquía de las relaciones, mientras que una ontología pesada es una ontología liviana enriquecida con las normas utilizadas para fijar la interpretación semántica de los conceptos y las relaciones (Gómez-Pérez, et, al. 2004) .

Otro referente en el área de inteligencia artificial, define de la siguiente forma una ontología (Maedche, 2002):

“Una ontología es una 6-tupla que consiste de conceptos, relaciones, jerarquía, función que relaciona conceptos no taxonómicamente, un conjunto de axiomas, y un conjunto de reglas.”

Formalmente:  $O := \{C, R, H, Rel, A, Ru\}$  donde:

- Dos conjuntos disjuntos, C (conceptos que representan clases de objetos) y R (las relaciones que describen las relaciones entre conceptos binarios).
- Una jerarquía de conceptos, una relación dirigida  $H \subseteq C \times C$  que se llama jerarquía de conceptos o taxonomía. Por lo tanto,  $H(C1, C2)$  significa C1 es un sub-concepto de C2.
- Una función de relación:  $Rel \rightarrow C \times C$  que relaciona los conceptos no taxonómicamente.
- Un conjunto de axiomas A que son oraciones lógicas siempre verdaderas que expresan las propiedades del modelo, expresados en un lenguaje lógico adecuado.
- Un conjunto de sentencias lógicas sobre el dominio del discurso Ru (reglas), expresado en un lenguaje lógico adecuado.

El término ontología se refiere al intento de formular un esquema conceptual exhaustivo y riguroso en un dominio dado, facilitando la comunicación y la posibilidad de compartir información entre diferentes sistemas. Esto hace la diferencia con el significado filosófico de la palabra ontología.

Utilizando estas técnicas semánticas, se adquiere la posibilidad de hacer los recursos accesibles de acuerdo a su contenido semántico. Estos conceptos semánticos se utilizan para describir un entorno, con una serie de conceptos atómicos existentes asociados a este, de forma estructurada jerárquicamente y que se utiliza como esqueleto fundamental para una base de conocimiento.

El uso de ontologías permite un tratamiento ponderado del conocimiento, sirviendo como herramienta para recuperar información de una manera automatizada.

## 2.2 Componentes de una ontología

La ontología define un modelo base que tendrá la definición semántica de las clases de objetos. En la ontología se definen un conjunto de términos representativos llamados conceptos. Las ontologías dependientes de dominio proveen conceptos en un dominio específico, que se enfoca en el conocimiento en un área limitada; las ontologías genéricas proveen conceptos en un dominio no específico, enfocado al conocimiento de un área ilimitada o de conceptos que son comunes a varios dominios, como por ejemplo el concepto de tiempo, masa, conjunto, evento, etc.

En una ontología, los conceptos son las unidades fundamentales para la especificación.

Proveen una base para la descripción de información. Cada concepto consta de 3 componentes básicos: *términos*, *atributos* y *relaciones*. Los términos son los nombres utilizados para referirse a un concepto específico que puede incluir un conjunto de sinónimos que especifican los mismos conceptos. Los atributos son las características de un concepto y describen el concepto a más detalle. Las relaciones se utilizan para representar correspondencias entre diferentes conceptos y para proveer una estructura general a la ontología.

Cada concepto define una clase que es una representación para una agrupación conceptual de términos similares. Por ejemplo, las computadoras podrían ser representadas como una



clase, que a su vez, tiene subclases representando a las distintas computadoras que pueden encontrarse, como ser: computadoras personales, mainframes, workstations, etc.

Las ontologías están formadas de los siguientes componentes que servirán para representar el conocimiento de algún dominio en específico:

- **Conceptos:** son las ideas básicas que intentan formalizar estos conceptos, pueden ser clases de objetos, métodos, planes, estrategias, procesos de razonamiento, etc. Un concepto abarca o encapsula un conjunto de elementos que tienen las mismas características.

- **Relaciones:** representan la interacción y el enlace entre los conceptos del dominio. Por ejemplo: subclase-de, parte-de, antes-que, formado-por, trabaja-en, etc. Particularmente, la relación subclase-de, también llamada es-un, es usada para formar la taxonomía del dominio identificando conceptos y sub-conceptos.

- **Funciones:** son un tipo concreto de relación, donde se identifica un elemento mediante el cálculo de una función que considera varios elementos de la ontología. Por ejemplo, categorizar-clase, etc.

- **Instancias:** utilizadas para representar objetos determinados de un concepto. Por ejemplo para el concepto *Persona*, una instancia es *María*.

- **Axiomas:** son teoremas que se declaran sobre relaciones que deben cumplir los elementos relacionados en la ontología. Por ejemplo: “Si *X* y *Y* son de la clase *Z*, entonces *X* no es subclase de *Y*” o “Para todo *X* que cumpla con la condición **Cond1**, *A* es *B*”, etc.

Las ontologías se pueden ver como un conjunto de conceptos-definiciones, estos conceptos pueden ser ordenados en jerarquías de taxonomías y tener propiedades asociadas.

En un sentido de manejo de conocimiento, las ontologías aportan una herramienta para la representación de la semántica de la información y automatización del proceso de recuperación; son el vehículo para la representación e intercambio del conocimiento en diferentes niveles de granularidad de diferentes dominios.

La ontología codifica la estructura relacional de conceptos, utilizada para describir aspectos del mundo. En el proceso de creación de una ontología, se hace uso de conocimiento previo almacenado en una ontología para la indexación y solución a problemas de búsquedas.

La anotación basada en ontologías es relativamente nueva, pero que ha mostrado ser de gran utilidad.

## 2.3 Tipos de Ontologías

En la literatura podemos encontrar diferentes tipos de ontologías. Se usan dos criterios principalmente para tales clasificaciones:

- El tipo de conocimiento contenido
- La motivación de la ontología

La primera de ellas, la clasificación por el contenido, fue propuesta en (Van Heijst, 1997), donde se distinguen tres tipos de ontologías:

**Ontologías terminológicas, lingüísticas:** Especifican los términos usados para representar conocimiento en el dominio. Un ejemplo de este tipo de ontologías es la red semántica UMLS (*Unified Medical Language System*) (Lindberg, 1993).

**Ontologías de información:** Especifican la estructura de los registros de la base de datos. Los esquemas de bases de datos serían un ejemplo.

**Ontologías para modelar conocimiento:** Especifican conceptualizaciones de conocimiento. Estas ontologías tienen una estructura interna mucho más rica que los anteriores tipos de ontologías, y éstas son las ontologías que interesan a los desarrolladores de sistemas basados en conocimiento.

Si clasificamos las ontologías de acuerdo a la motivación de las mismas podemos clasificarlas en 4 grupos:

**Ontologías para la representación de conocimiento:** Permiten explicar las conceptualizaciones que subyacen de los formalismos de representación de conocimiento (Davis, 1993).

**Ontologías genéricas:** Definen conceptos considerados genéricos en diferentes áreas. Ejemplos de tales conceptos serían componente, subclase, proceso, estado, etc. Estas ontologías son reutilizables en diferentes dominios. Se llaman también ontologías abstractas o superteorías porque permiten definir conceptos abstractos, y dichas ontologías pueden ser usadas para definir conceptos de forma más específica en diferentes dominios. Como

ejemplos podemos ver la taxonomía, la mereología, la topología y la teoría general de sistemas.

**Ontologías del dominio:** Definen conceptualizaciones específicas del dominio. Las metodologías actuales de adquisición de conocimiento distinguen entre ontologías y conocimiento del dominio, porque el último describe situaciones factuales del dominio, mientras que las ontologías imponen descripciones sobre la estructura y contenido del conocimiento del dominio.

**Ontologías de aplicación:** Están ligadas al desarrollo de una aplicación concreta. Tales ontologías cubren los aspectos relacionados con aplicaciones particulares. Típicamente, estas ontologías toman conceptos de ontologías del dominio y genéricas, así como métodos específicos para realizar la tarea, por lo que no son muy adecuadas para ser reutilizadas.

## 2.4 Tecnologías de la Web Semántica

Actualmente, la Web es un espacio preparado para el intercambio de información diseñado para el consumo humano. Las páginas web son creadas por personas para ser entendidas por personas. No existe un formato común para mostrar la información, por lo cual, los desarrolladores de páginas web crean sus páginas dependiendo de los potenciales usuarios que van a visitarlas.

Los actuales navegadores web realizan la búsqueda de información, con más o menos fortuna, mediante palabras claves que aparecerán en el código HTML de las páginas web dispersas en Internet. En los últimos años, algunas empresas están realizando anotaciones de datos introducidas dentro de este código HTML, siguiendo algún esquema de anotación común, normalmente basado en XML, para facilitar a los buscadores encontrar dicha información más fácil y rápido.

Otra carencia de la situación actual es que, con los estándares web del momento, no se puede diferenciar entre información personal, académica, comercial, etc. Es decir, cuando un buscador web realiza una consulta con algunas palabras clave, normalmente aparece información que no es útil porque no corresponde a lo que estamos buscando. Además no todas las páginas proporcionan igual cantidad de información, debido precisamente a que no existe un formato o convenio que establezca qué contenido debe añadirse a las páginas web. Por otro lado, los agentes de búsqueda actuales no se diseñan para comprender la información

que reside en la web, precisamente porque es prácticamente imposible conocer la representación de todos los datos ubicados en las diferentes páginas. La información que puede encontrar el buscador puede estar relacionada con la prensa, demandas judiciales, información que reúna compañías de seguros, entre otras.

Normalmente se pierde mucho tiempo en seleccionar la información que nos puede ser útil, y navegando por las referencias URL hasta encontrar, con suerte, lo que estamos buscando. Por ejemplo, si queremos encontrar un dentista particular para poder acordar una visita, no es seguro que los dentistas tengan páginas personales con información de su agenda de consulta de pacientes, o que la obra social tenga un listado de sus dentistas, y que la información de los dentistas esté relacionada con la ciudad que estoy buscando. Si por casualidad encontrásemos varias páginas personales de dentistas que tuvieran en su anotación HTML esas palabras clave, es bastante improbable que la información que nos proporcionen sea similar; algunos pueden tener datos informativos, otros publicaciones, especialidades, horarios. Es indudable que las ventajas que ofrece Internet son enormes a la hora de buscar información, pero adolece de una manera de encontrar información de forma precisa y de poder realizar deducciones con la información existente. Principalmente es necesario destacar en este escenario que para encontrar la información necesaria y poder concertar un turno, es necesario que lo haga una persona que entienda toda la información y que pueda inferir cual es el dentista que más le conviene ya sea por el lugar en donde vive, la obra social que posee y la disponibilidad horaria de ambos, paciente y médico. Si ahora pensamos que este mismo proceso lo realice un agente de software, el mismo tendría que tener habilidades de interpretar lenguaje natural y procesarlo tal como lo hacen los humanos. Hasta el momento, este tipo de tecnología está en desarrollo y es un área de investigación inmadura. El uso de ontologías en la creación de las páginas web, facilita el trabajo de agentes y de la interpretación que éstos pueden hacer de los recursos web. Así, un agente no necesita interpretar el lenguaje natural sino más bien conocer los términos lógicos usados para representar ciertos conceptos. Luego su tarea es encontrar dichos conceptos y relacionarlos con otros a través de las relaciones definidas en la ontología usada.

Entre los principales componentes de la Web Semántica podemos encontrar XML, XML Schema, RDF, RDF Schema y OWL. La descripción de OWL en inglés “Ontology Web Language” describe las funciones y relaciones de cada componente de la Web Semántica:

- **XML:** Provee una sintaxis elemental para las estructuras de contenidos dentro de documentos.
- **XML Schema:** Es un lenguaje para proporcionar y restringir la estructura y el contenido de los elementos contenidos dentro de documentos XML.
- **RDF:** Es un lenguaje simple para expresar modelos de los datos, que refieren a los objetos “recursos” y a sus relaciones. Un modelo de RDF-based se puede representar en sintaxis de XML.
- **RDF Schema:** Es un vocabulario para describir propiedades y clases de recursos RDF-based, con semántica para generalizar-jerarquías de las propiedades y clases.
- **OWL (Ontology Web Language):** Es un mecanismo para desarrollar temas o vocabularios específicos en los que podamos asociar esos recursos.

RDF y OWL son estándares para la Web Semántica que proporcionan un marco para la gestión, integración, compartir y reutilización de datos en el Web.

OWL proporciona más vocabulario que RDF(S) para describir propiedades y clases tal como: relaciones entre clases (por ejemplo “clases disjuntas”), cardinalidad (por ejemplo “exactamente uno”), igualdad, más tipos para las propiedades, características de las propiedades (por ejemplo “simetría”), y clases enumeradas.

OWL consta de 3 sub lenguajes:

- **OWL Lite:** Suficiente para los usuarios que tan sólo piden posibilidades de clasificación en la jerarquía de conceptos (clases) de la ontología y restricciones simples. Por ejemplo, aunque OWL proporciona restricciones de cardinalidad, sólo permite valores 0 ó 1. Por lo tanto tiene una complejidad formal inferior a OWL DL.
- **OWL DL (Description Logic):** Es el lenguaje indicado para los usuarios que requieren el máximo de expresividad pero exigiendo completitud computacional (se garantiza que todas las conclusiones son computables) y decibilidad (todos los cálculos acaban en un tiempo finito). Incluye todos los constructores de OWL, pero sólo se pueden usar con restricciones; por ejemplo: mientras una clase puede ser a la vez subclase de muchas clases, no puede ser una instancia de otra clase.
- **OWL Full:** Se dirige a aquellos usuarios que necesitan la máxima expresividad y la libertad sintáctica de RDF pero sin garantía computacionales. Permite, por ejemplo,

aumentar el significado de vocabulario predefinido (en RDF o en OWL), por lo que es muy improbable que ningún software de razonamiento sea capaz de soportar razonamiento completo para cualquier característica de OWL Full.

OWL Full se puede ver como una extensión de RDFS, mientras que OWL Lite y OWL DL se pueden ver como extensiones de una vista restringida de RDF. Cualquier documento OWL (Lite, DL, Full) es un RDF Document, y cualquier documento RDF es un documento OWL Full pero sólo algunos documentos RDF serán documentos OWL Lite o OWL DL legales. Por esta razón, hay que tener cuidado cuando se desea migrar un documento RDF a OWL.

### Capas de la Web Semántica

La figura 2.1 muestra las capas de la web semántica. Como se puede apreciar, la misma tiene diferentes niveles que representan grados de descripción y restricciones. Las capas más bajas soportan una definición sintáctica de un dominio mientras que las capas superiores aportan definiciones semánticas y pragmáticas del dominio, donde la sintaxis está asociada a la descripción de una sentencia válida en cuanto a la estructura que adopta, la semántica aporta significado a los conceptos y la pragmática la intencionalidad del uso de los mismos.

A continuación se describen los diferentes estándares asociados a cada una de las capas

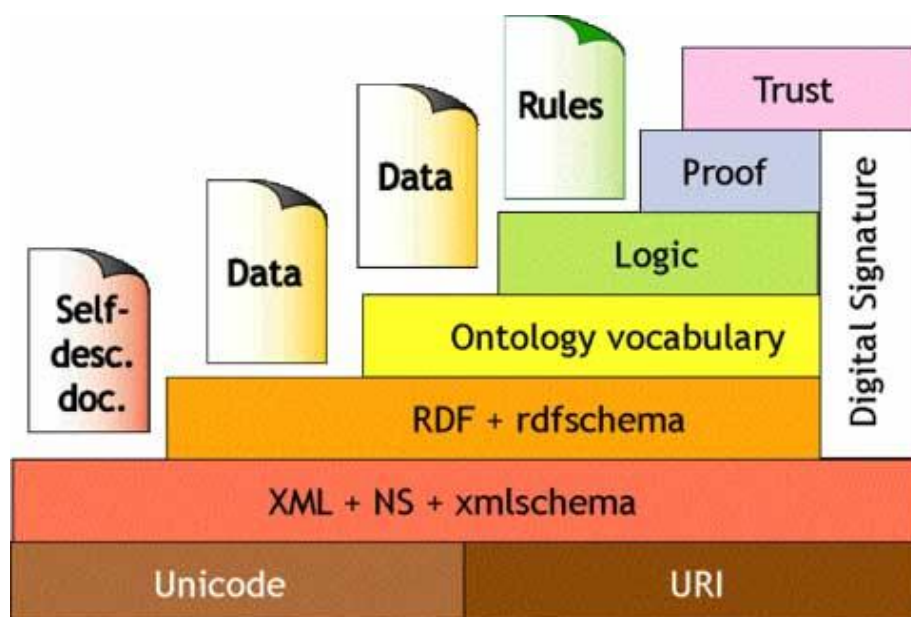


Figura 2.1 - Capas de la web semántica

de la web semántica:

## Unicode

Es un estándar cuyo objetivo es proporcionar el medio por el cual un texto en cualquier forma e idioma pueda ser codificado para el uso informático. El mismo nos permite mostrar información en cualquier idioma y con la certeza de que no aparezcan símbolos extraños.

## URI

Son cadenas que permiten acceder a cualquier recurso de la Web. En la Web Semántica las URIs son las encargadas de identificar objetos. Todos los objetos pueden ser identificados mediante una URI. Si dos objetos cuentan con la misma URI pueden existir colisiones. El grupo de trabajo del W3C está intentando resolver este problema.

## XML + NS + XMLSchema

Esta es la capa más técnica de la Web Semántica. En ella se encuentran agrupadas las diferentes tecnologías que posibilitan la comunicación entre agentes.

El XML (eXtensible Markup Language) nos ofrece un formato común para el intercambio de documentos, Namespaces (NS) proporciona un método para calificar elementos y atributos de nombres usados en documentos XML asociándolos con espacios de nombre identificados por referencias URIs. XML Schema es un lenguaje que permite describir la estructura y restringir el contenido de documentos XML.

## RDF + RDFSchema

Está basada en la capa anterior, define el lenguaje universal con el que podemos expresar diferentes ideas en la Web Semántica. RDF es un lenguaje que define un modelo de datos para describir recursos mediante tripletas sujeto-predicado-objeto.

Los dos primeros serán URIs y el tercero puede ser URI o un valor literal. RDF Schema es un vocabulario RDF que nos permite describir recursos mediante una orientación a objetos. Esta capa no sólo ofrece una descripción de los datos, sino también cierta información semántica.

## Ontology (Ontologías)

Nos permite clasificar la información. Esta capa permite extender la funcionalidad de la Web Semántica agregando nuevas clases y propiedades para describir los recursos.

### Logic (Lógica)

Además de ontologías se precisan reglas de inferencia, por lo que en esta capa se definen las reglas lógicas para inferir nuevos conocimientos.

### Proof (Pruebas)

Se intercambiarán “pruebas” escritas en el lenguaje unificador de la Web Semántica. Este lenguaje posibilita las inferencias lógicas realizadas a través del uso de reglas de inferencia.

### Trust (Confianza)

Hasta que no se haya comprobado de forma exhaustiva las fuentes de información, los agentes deberían ser muy escépticos acerca de lo que leen en la Web Semántica.

### Digital Signature (Firma digital)

Utilizada por los ordenadores y agentes para verificar que la información ha sido ofrecida por una fuente de confianza (**Web semántica, 2007**).

## 2.5 Las redes de ontologías en las que se basó el proyecto

Una red de ontologías es un conjunto de ontologías relacionadas entre sí a través de una variedad de diferentes relaciones, tales como mapeo, modularización, versión y dependencia. Los elementos de este conjunto se llaman ontologías en red (**Allocca, y col., 2009**).

Una red de ontologías difiere de un conjunto de ontologías individuales interconectadas en las relaciones definidas entre ontologías ya que en una red de ontologías las meta-relaciones entre las ontologías de la red se expresan explícitamente (**Diaz y col., 2012**). Hay algunos modelos que cubren tanto los aspectos sintácticos y semánticos para el manejo de las relaciones entre las ontologías de la red. En la ontología DOOR (*Descriptive Ontology of Ontology Relations*) se definen las relaciones generales entre ontologías que conforman una red mediante primitivas y reglas (**Allocca, y col., 2009**) ontológicas, como ser: *includedIn*, *equivalentTo*, *similarTo* e *isAVersionOf* entre otras. En este trabajo, se utilizan las siguientes relaciones:

- **isTheSchemaFor**: Conceptualmente, esta relación mantiene el vínculo entre un modelo y su metamodelo.
- **usesSymbolsOf**: Esta relación ocurre cuando las propiedades en una ontología O implica individuos de otra ontología O', de tal manera que la ontología O' define



algunas propiedades que tienen valor en los individuos que se clasifican por clases de la ontología O. Son relaciones de enlaces entre las ontologías O y O' de tal manera que se abstraen de la ontología particular, O para ser importados y se centra en cambio en los símbolos de O' que se vayan a reutilizar.

- ***cross-ontologySubsumption***: Esta relación se mantiene entre una clase perteneciente a una ontología y su superclase que pertenece a otra ontología.
- ***relatedTo***: Esta relación se mantiene entre dos clases que pertenecen a diferentes ontologías.

## Red de ontología AOnet

La red de ontologías *AONet* propuesta por **Romero y col. (2012)** provee una herramienta para guiar el diseño automatizado de evaluaciones dentro de ambientes de E-Learning. Esta Red está compuesta de 5 ontologías que conceptualizan tres dominios: dominio del tema (*Topic Domain*), recursos educativos (*Educational Resources*) y evaluación (*Assessment*). La figura 2.2 muestra la red de ontologías *AONet* donde es posible distinguir los tres dominios involucrados, sus relaciones y las diferentes ontologías que participan en cada dominio.

La evaluación es un recurso educativo (*Educational Resources*) y como tal, tiene

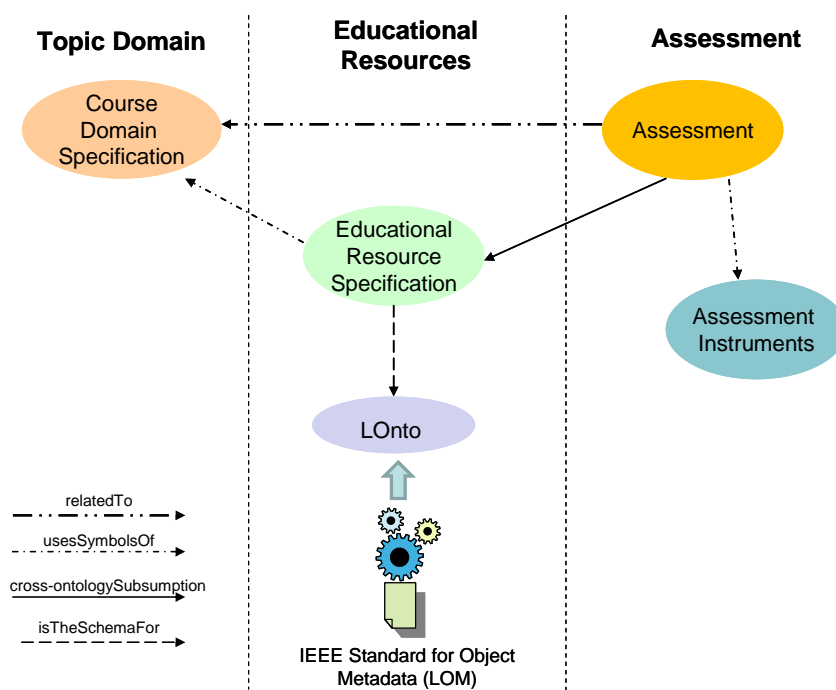


Figura 2.2 - red de ontologías AONet

metadatos que la describen representados por la ontología *LOnto*. Para el desarrollo de una evaluación, hay una gran cantidad de instrumentos que se puede utilizar conceptualizados en la ontología *Assessment Instruments*. Además, una evaluación evalúa temas o materias que deben ser aprendidas por los estudiantes, descriptos en la ontología *Topic domain*. En este contexto, AONet conceptualiza de una manera modular, todos los aspectos de evaluación del dominio. Para la realización de este trabajo nos basaremos en el dominio *Assessment* teniendo en cuenta la ontología de evaluaciones (*Assesment ontology*) y la ontología de Instrumentos (*Assessment Instruments ontology*).

## Ontología de Instrumento

Esta ontología formaliza la conceptualización de los diferentes instrumentos que se utiliza para generar una evaluación como ser por ejemplo: selección simple, selección múltiple, preguntas, completar los espacios en blanco, relacionar, entre otros. También, se definen reglas lógicas para restringir el uso de diferentes instrumentos, teniendo en cuenta las recomendaciones pedagógicas. De esta manera, es posible guiar a los educadores en el uso correcto de los instrumentos, dando como resultado evaluaciones válidas y confiables. Esta ontología contempla diferentes tipos de evaluación. La evaluación se puede clasificar en evaluación formal, informal y semi-formal, dependiendo de la formalidad y la estructura de los instrumentos utilizados. De esta manera las evaluaciones formales se estructuran de la siguiente manera: hay un lugar y una hora en que se llevan a cabo. Las evaluaciones semi-formales son deberes y tareas que el estudiante pueda tener durante días de lección y continuar fuera de él. Las evaluaciones informales no están estructuradas en absoluto.

La figura 2.3 muestra un extracto de la ontología de instrumentos llamada *Assessment Instrument*. Como se puede ver en el gráfico, el concepto padre del modelo es la clase *Instrument*, la cual tiene como subclases inmediatas *FormalInstrument* (para definir los instrumentos formales) y *SemiformalInstrument* (para definir los instrumentos semiformales).

A su vez, cada una de estas clases se especializan en instrumentos de cada una de sus ramas, la clase *FormalInstrument* se especializa en *EssayActivity* (para los instrumentos de ensayo) y en *ObjectiveActivity* (para los instrumentos objetivos del modelo).

Los instrumentos *EssayActivity* indican instrumentos del tipo pregunta – respuesta, las cuales se pueden especializar más en *RestrictedEssay* (para las preguntas que tienen un tema acotado) o en *UnrestrictedEssay* (para las preguntas de temas más amplios o de interpretación más abierta).

Por el lado de los instrumentos *ObjectiveActivity*, se llega a especializar al mismo en 3 clases de instrumento, estas son *Choice* (para los instrumentos de selección, ya sean de selección simple o “*SimpleChoice*” o de selección múltiple o “*MultipleChoice*”), *Correspondence* (para los instrumentos donde se deben unir mediante líneas 2 frases correlacionadas) y *Completion* (para los instrumentos donde se debe completar una frase con una palabra o una pequeña frase para que la oración en si tenga coherencia).

Los instrumentos de *Choice* tienen asignadas opciones (instancia de la clase *Option*), las cuales pueden ser del tipo verdaderas (la clase *TrueOption*) o falsa (la clase *Distractor*). Estas opciones tienen asignadas respuestas del tipo *TrueFalse*.

Los instrumentos de *Correspondence* y *Completion* tienen asignadas respuestas del tipo *Answer*.

Esta clase *Answer* se especializa en 4 clases de respuestas: *TrueFalse* (para las respuestas booleanas), *Numeric* (para las respuestas numéricas), *Text* (para las respuestas del tipo textual) y *Relation* (para las respuestas del instrumento de *Correspondence*, donde se relacionan 2 frases).

Por el lado de los instrumentos semi-formales (*SemiformalInstrument*) representa las tareas que los estudiantes hacen durante el curso y continúa fuera de él. Este instrumento se define en 2 grandes clases, la clase de *SimpleInstrument* (para los instrumentos simples) y la clase *CompositeInstrument* (para los instrumentos compuestos de instrumentos simples). Los instrumentos simples se pueden definir en *ConceptualMap*, *Essay* y *Exercise* y los instrumentos compuestos solo se pueden especializar en *Portfolios*.



Esta ontología modela la estructura de una evaluación en un proceso de enseñanza – aprendizaje en vista de un dominio E-learning.

Teniendo en cuenta la didáctica, las evaluaciones son necesarias para valorar el proceso de aprendizaje como también para realizar un seguimiento del mismo. Dentro de un entorno de educación a distancia (E-Learning), las evaluaciones son consideradas como difíciles de realizar dado principalmente la relación entre profesor-alumno que se genera en este tipo de entornos. Sin embargo, las evaluaciones son un elemento necesario ya que proporcionan información para orientar y mejorar el proceso de enseñanza-aprendizaje.

Página | 28

Hay un momento en el proceso de enseñanza-aprendizaje en el cual la evaluación tiene efecto. Si la evaluación es temprana en el proceso, se dice que es una evaluación de diagnóstico. Su objetivo es determinar el nivel de conocimiento de un estudiante antes de iniciar el ciclo de aprendizaje para que el contenido de este ciclo se adapte a los resultados.

Si no se realiza la evaluación durante el proceso de enseñanza-aprendizaje, con el fin de hacer un seguimiento de los conceptos adquiridos hasta entonces, es formativa. Su objetivo es dar información a los estudiantes y educadores acerca de qué tan bien los estudiantes entendieron el contenido específico.

Si la evaluación se produce al final del proceso, con el objetivo de entender el grado general de aprendizaje, es una evaluación sumativa. Su meta es un juicio para obtener una calificación, titulación o acreditación.

La Figura 2.4 muestra el modelo de la ontología *Assessment*. El concepto principal es la clase *Assessment*. Este es un requisito previo para la consecución de la calidad educativa. En cuanto a composición de evaluación, la relación *isComposedBy* establece que una evaluación está compuesta por las actividades que a su vez tienen uno o más reactivos.

La clase *Moment* tiene tres instancias: Diagnóstico, sumativa y formativa, que representan la clasificación de la evaluación según el momento en que entre en vigor (Romero y col., 2012).

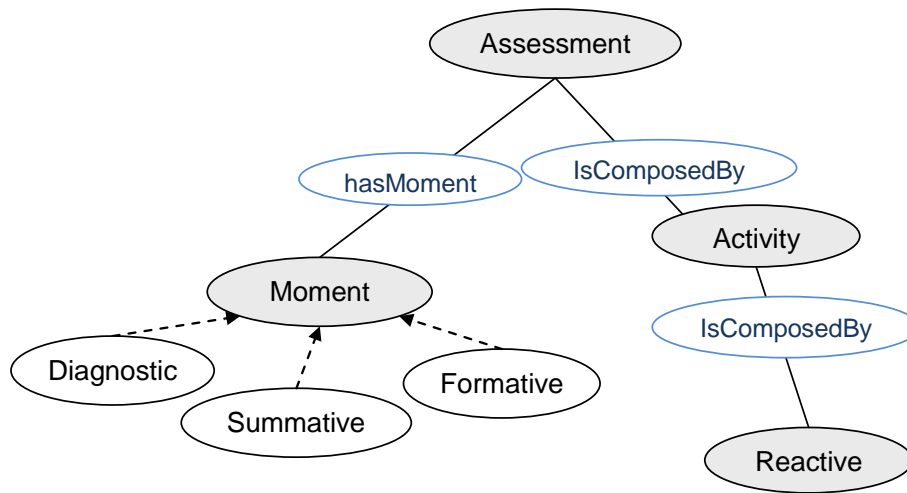


Figura 2.4 - Ontología de Assessment

## 2.6 Conclusiones

En este capítulo presentamos los conceptos que sirvieron como base de la investigación e implementación del sistema, como así también el ámbito que cubren las ontologías, el cual se está expandiendo cada vez más en distintas áreas del desarrollo. Con el progreso de estas nuevas tecnologías se desprenden ideas a la hora de buscar soluciones más factibles.

## Capítulo 3 - Análisis y Diseño del Sistema

### 3.1 Metodología aplicada

El desarrollo del sistema se llevó a cabo utilizando metodologías ágiles, más específicamente la técnica de Scrum.

Esta metodología la pusimos en práctica de la siguiente forma:

- Nos propusimos hacer 2 Sprint que abarcaron alrededor de 6 meses.
- Muestras semanales a las directoras del proyecto para validación relevamiento de requerimientos nuevos.
- Desarrollo del sistema en paralelo a la escritura del informe final.
- Pruebas semanales para verificar el funcionamiento de los módulos terminados por los integrantes del grupo.
- Versionado diario de código en un servidor SVN.

Para lograr todo esto usamos la herramienta de gestión de desarrollo de metodologías ágiles denominada Kunagi (**Documentación Kunagi**). La misma presenta un conjunto de funcionalidades que fueron aplicadas y ajustadas a nuestro proyecto de la siguiente forma:

#### 3.1.1 Whiteboard

El whiteboard del aplicativo nos sirvió como representación visual del progreso de la corriente de Sprint. En esta pantalla se pueden observar las tareas libres, las tareas tomadas por algún integrante del grupo y las tareas terminadas, todo esto medido por porcentaje basado en puntos. Definimos que 1 punto es equivalente a 4 horas de trabajo.

#### 3.1.2 Sprint Backlog

Esta sección nos ayudó a organizar el trabajo diario que se planteaba en el sprint, especificándose características propias del sprint, como el periodo que cubría el mismo y las tareas involucradas. La figura 3.1 muestra la ventana de backlog donde se puede ver que se completó 1 punto (4 hs. de trabajo) y quedan todavía 7 puntos que no fueron completados.

#### 3.1.3 Qualities



A medida que íbamos desarrollando se fueron estableciendo requisitos no funcionales que afectaron a múltiples historias. Por ejemplo la implementación de la ontología en memoria o el control de acceso al sistema por parte de los agentes

**Stories in this Sprint**

Story ID	Story Name	Progress
sto8	Functional Quality Backlog	100% completed, 10 hrs burned
sto9	Groundbraking Scrum Backlog Vision	100% completed, 20 hrs burned
sto7	Amazing Product Backlog Vision	50% completed, 12 hrs left
sto6	Unsurpassed Concept Vision	34% completed, 46 hrs left

**Sprint Properties**

Label: Our first Sprint!      My emoticon: 😊 😐 😞 😡 😢

Goal: .

Releases: re12 1.1      Create a comment...

Begin: 2012-09-22      Length: 33      End: 2012-10-24

Stories: Completed 1 p      Estimated 7 pts

Tasks: Burned 66 hours      Remaining 58 hours

Planning Note: .

Review Note: .

Retrospective Note: .

[Download as PDF](#)

[User Guide](#)

Figura 3.1 - ventana de Backlog de la herramienta Kunagi

### 3.1.4 Issues

Esta vista es la de edición, donde se realizaron seguimientos de retroalimentación, problemas y nuevas ideas del proyecto.

### 3.1.5 Impediments

En lo que respecta a impedimentos, solo podemos decir que surgieron por desconocimiento del dominio, principalmente en el ámbito de las ontologías.

### 3.1.6 Risk

En esta parte se detallan los riesgos que se pueden encontrar durante la implementación. Por lo general, estos riesgos fueron descubiertos sobre la marcha.

### **3.1.7 Next Sprint**

En esta fase preparamos el próximo sprint, estimando el tiempo que nos iba a consumir.

## 3.2 Sprints

Al proyecto lo partimos en 2 grandes sprints, dividiendo los mismos de acuerdo a las funcionalidades de los componentes.

En el primer sprint nos encargamos de desarrollar el subsistema de gestión de evaluaciones, junto al de actividades, reactivos y los instrumentos, definiendo el modelo de los elementos, sus DAOs, Services y Views. También generamos un conjunto de casos de pruebas para las diferentes partes del sistema de gestión y corroborar su correcto funcionamiento.

Una vez desarrollado completamente el primer sprint y todas las pruebas que definimos pasaron con éxito, comenzamos a desarrollar el segundo sprint que corresponde a la creación de la ontología de una evaluación en memoria, la gestión de las reglas y los conjuntos de estas, y la aplicación de las mismas dentro de la ontología para validar el modelo creado.

### 3.3 Modelos de Componentes

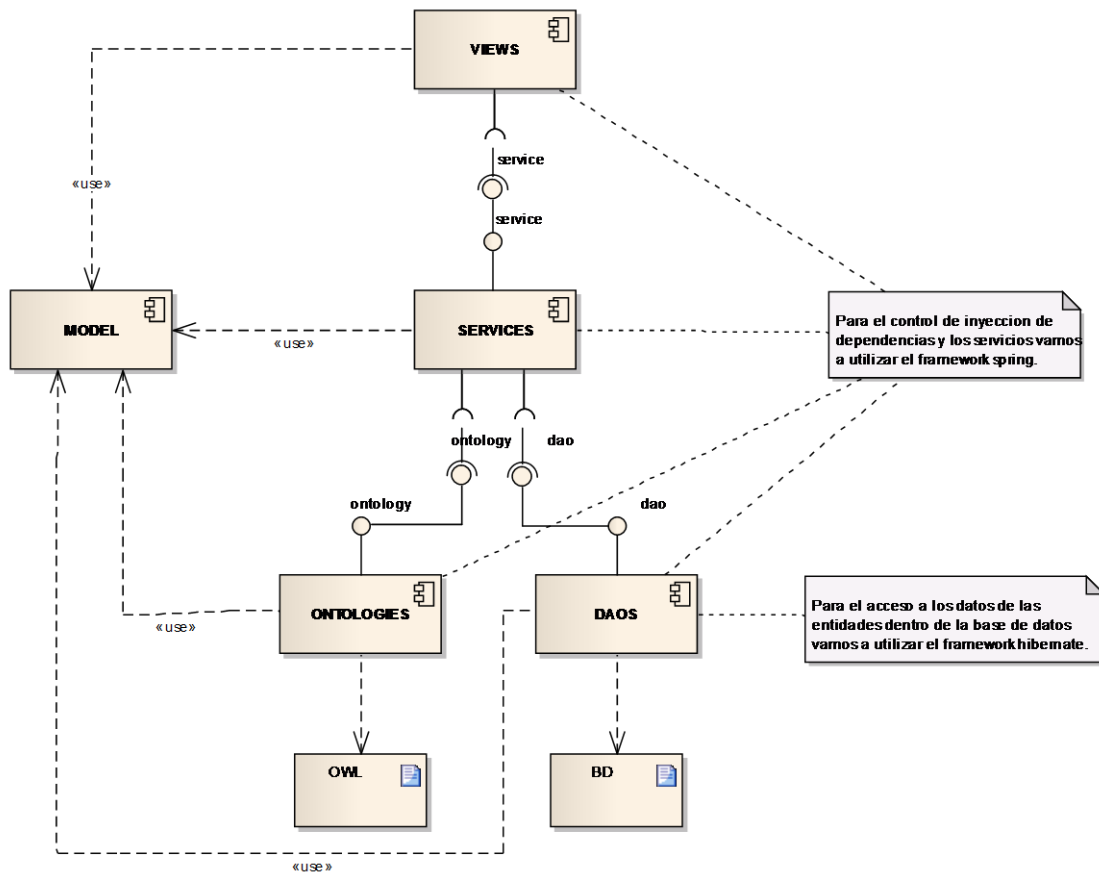


Figura 3.2 – Arquitectura de componentes

Como se observa en la figura 3.2, la estructuración del sistema se encuentra definida por el enfoque de desarrollo MVC, por lo que vemos claramente definidas las capas de acceso a los datos persistentes (el componente DAOs). Sobre esta capa, utilizando los servicios proporcionados por esta, tenemos la capa de prestación de servicios (el componente SERVICE), el cual también hace uso de los servicios prestado por la capa de acceso a la ontología (el componentes ONTOLOGIES) definidos al mismo nivel que la capa de acceso a los datos. Sobre la capa de servicios definimos la capa de interfaces gráficas (el componente VIEW), la cual utiliza los servicios prestados por la capa inferior (en este caso la capa de servicios). Este sistema utiliza un modelo de clases (el componente MODEL) definido como un lenguaje de uso común para la comunicación transparente entre las capas.

### 3.4 Modelo de Casos de Uso

A continuación se presentan los casos de usos más significativos que fueron definidos para el desarrollo del sistema, para mayor detalle de los mismos, referirse al anexo del informe.

Primeramente se definieron los casos de usos del sistema de gestión de evaluaciones y de reglas.

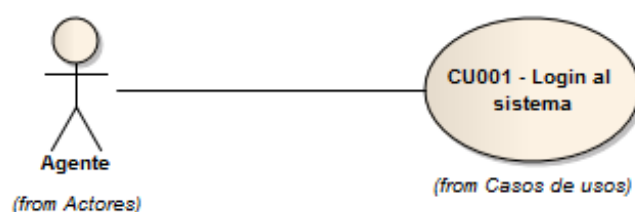


Figura 3.3 - CU ingreso

En este caso de uso podemos apreciar que el actor principal que va a interactuar con el sistema es el agente (investigador). Es el usuario más familiarizado con la herramienta y lo hace a través de la autenticación de usuario contra el sistema. En caso de que los datos no coincidan, el sistema mismo se encarga de informarle por medio de un mensaje el fallo de ingreso.

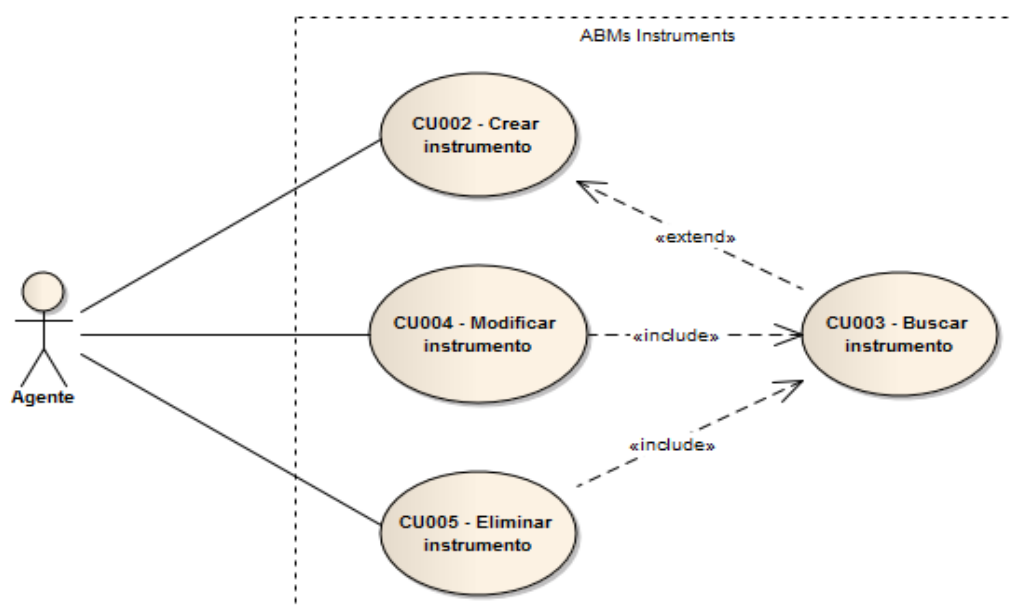


Figura 3.4 - CU Administración de instrumentos

La figura 3.4 muestra los casos de usos para la gestión de instrumentos, representan la funcionalidad propia que tienen los profesores a la hora de crear, modificar y borrar, clases de ejercicios que se tomarán en una evaluación. El sistema permite la búsqueda y selección de los mismos.

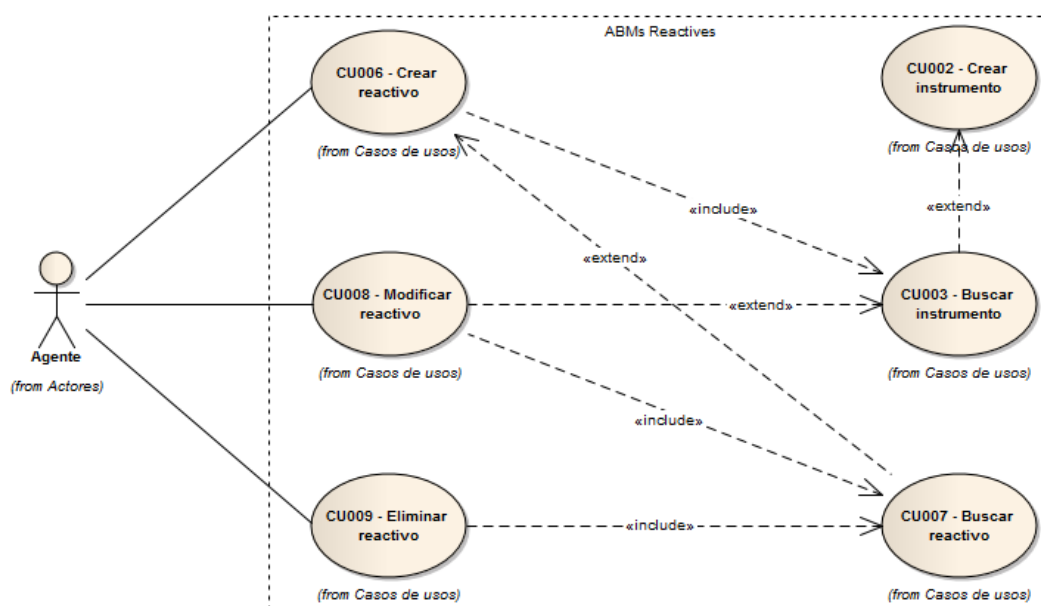


Figura 3.5 - CU Administración de reactivos

La administración de reactivo (figura 3.5) contempla un conjunto de casos de usos para la creación, modificación y eliminación de reactivos. Incluyen a un tipo de instrumento, donde su asignación representa la clase de reactivo que se está tratando.

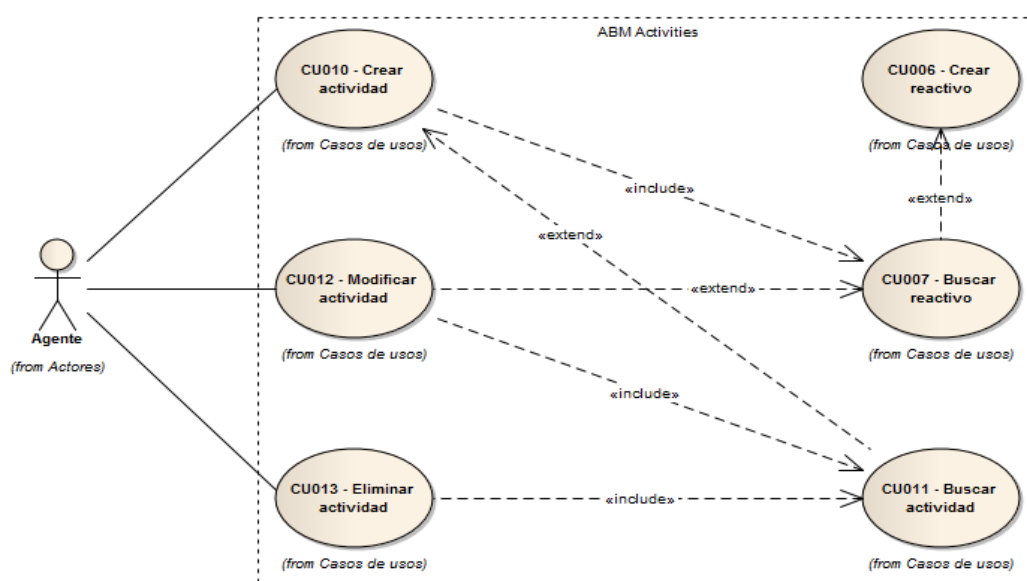


Figura 3.6 - CU Administración de actividades

Los casos de usos que identifican toda la administración de las actividades (figura 3.6), dan la funcionalidad a los usuarios de crear, editar y borrar actividades que forman parte de las evaluaciones ya creadas. El usuario tiene la posibilidad de realizar búsquedas de actividades y de sus reactivos asociados.

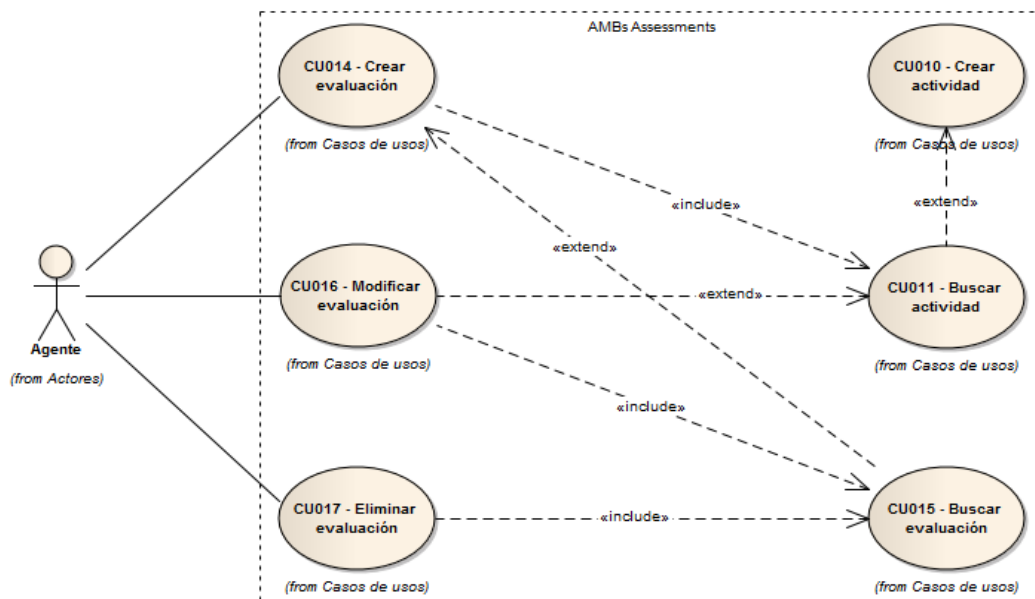


Figura 3.7 - Administración de evaluaciones

La gestión de evaluaciones se ve vinculada por un conjunto de casos de usos (figura 3.7), que permiten a los usuarios accionar sobre ese elemento, creando, editando y eliminando, donde la misma tiene que ser generada, para que el sistema tenga sentido.

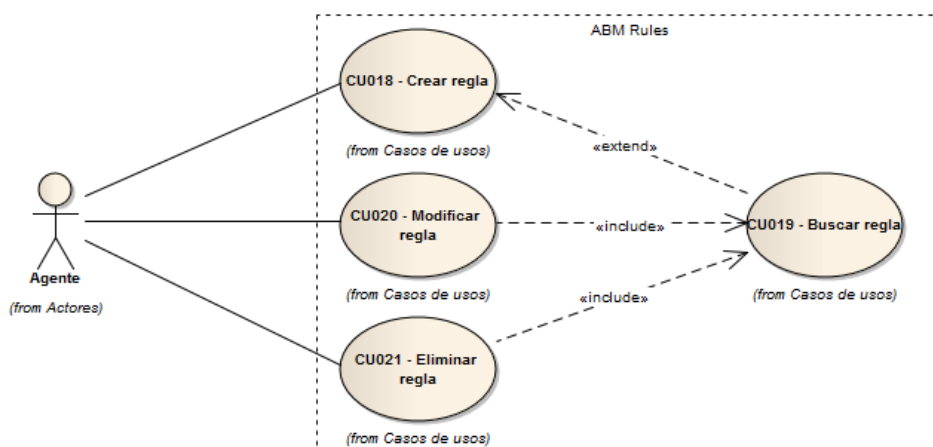


Figura 3.8 - Administración de reglas

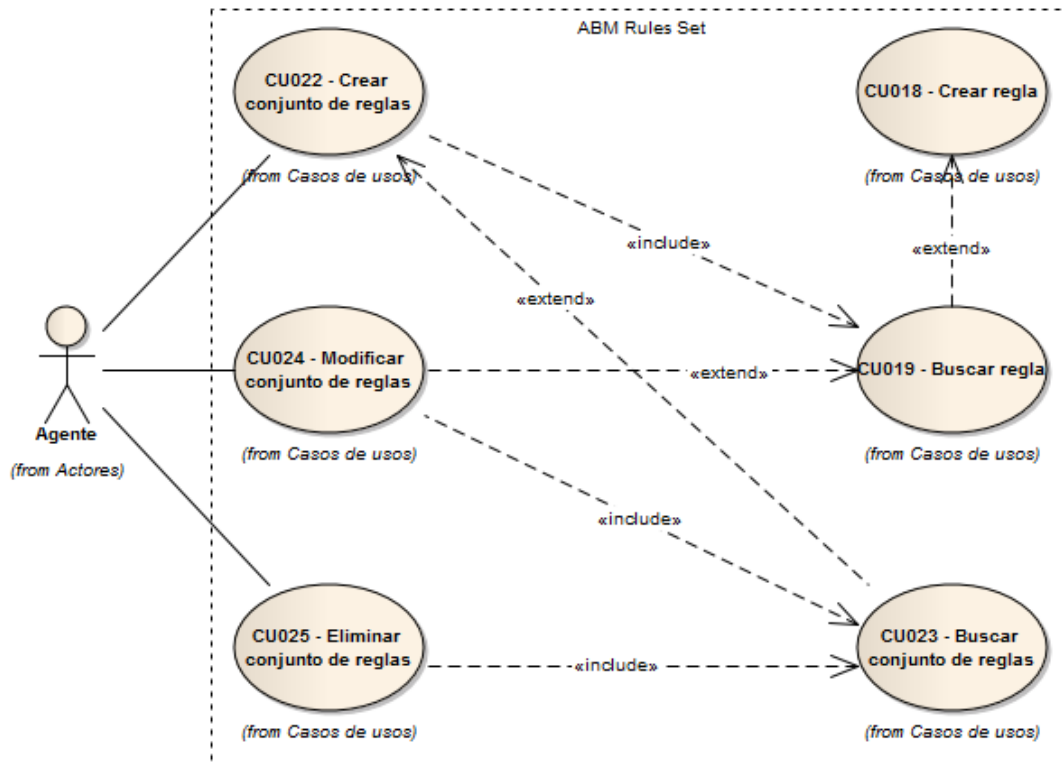


Figura 3.9 - Administración de conjuntos de reglas

El usuario tiene la opción de crear, modificar y eliminar reglas y a la vez conjuntos de las mismas, que se forman por la agrupación de una o varias reglas seleccionadas por el actor. La administración de reglas y conjuntos permite también realizar búsquedas como se muestra en la figura 3.9.

Finalmente la figura 3.10 muestra los casos de uso para la validación de evaluaciones.

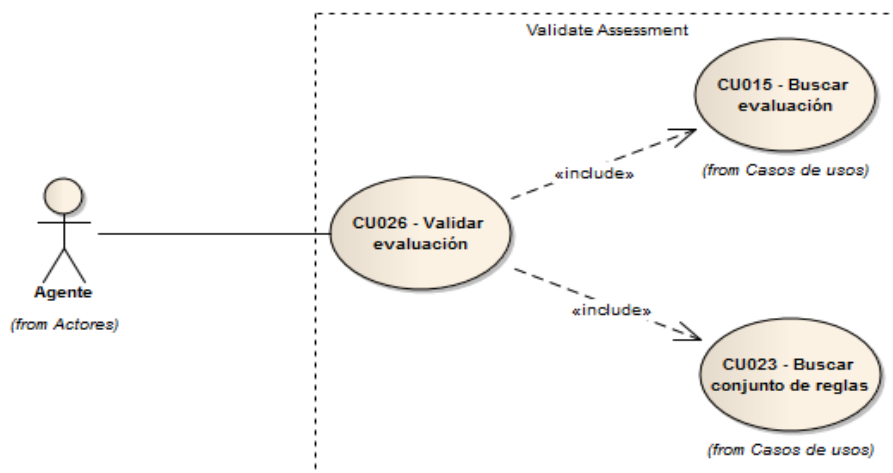


Figura 3.10 - Validación de evaluaciones



### 3.5 Modelo de Clases

El modelo de clases que definimos para la comunicación del sistema de manera íntegra se presenta en partes por el tamaño del mismo.

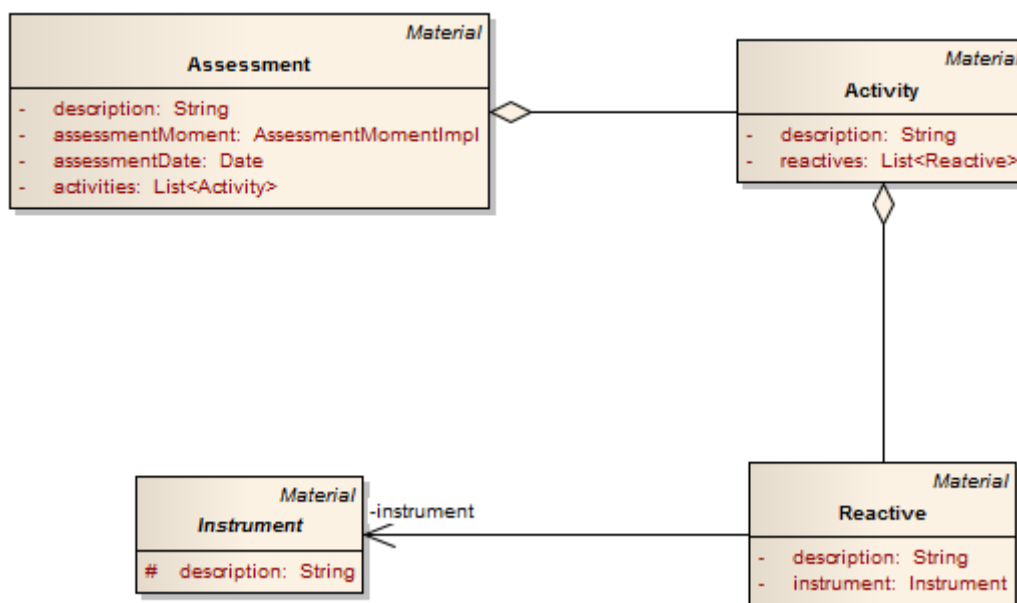


Figura 3.11 – Diagrama de evaluaciones

En el diagrama 3.11 vemos definidas la clase de *Assessment* (Evaluación), la cual está compuesta por un conjunto de instancias de *Activity* (Actividades). A su vez, cada una de estas actividades contiene un conjunto de *Reactive* (Reactivos) asociados. Estos reactivos solo tienen asociados un *Instrument* (Instrumento) cada uno.

En el diagrama 3.12 vemos el diagrama de las clases que se definen bajo la clase *Instrument*. En este diagrama solo se ven los instrumentos formales porque fueron los únicos especificados y delimitados.

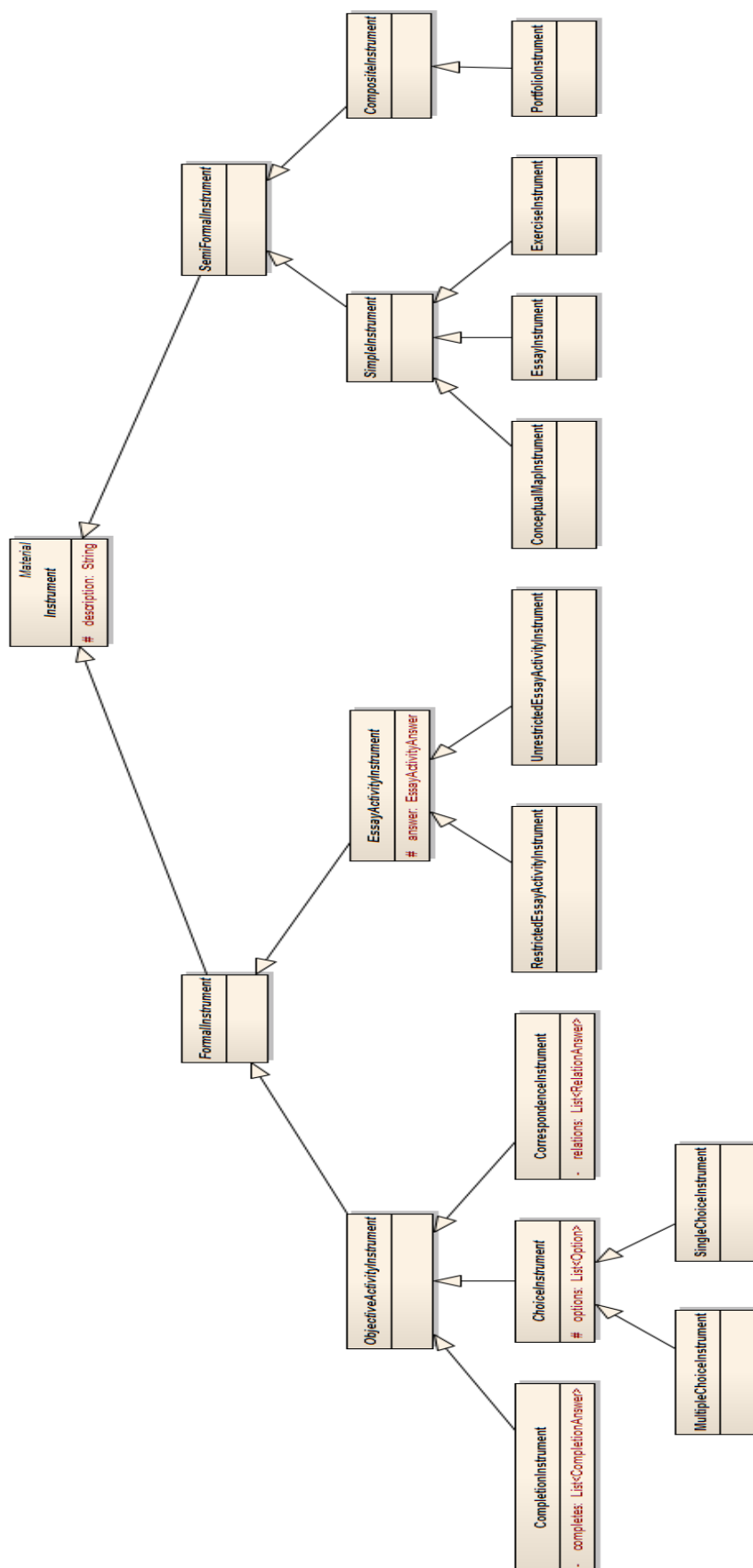


Figura 3.12 – Diagramas de instrumentos

Los Instrumentos de Selección (*ChoiceInstrument*) tienen asignados un conjunto de Opciones (*Option*), mientras que los demás instrumentos (*EssayActivityInstrument*, *CorrespondenceInstrument* y *CompletionInstrument*) contienen una única (en el caso de los *EssayActivityInstrument*) o un conjunto de respuestas (en el caso de los *CorrespondenceInstrument* y *CompletionInstrument*).

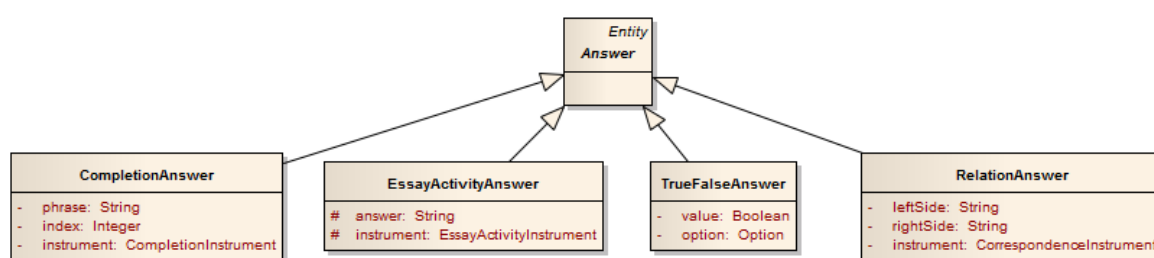


Figura 3.13 – Diagramas de respuestas

En el diagrama de la figura 3.13 tenemos el modelo de las respuestas para los instrumentos dentro del sistema. Son utilizadas para dar a cada tipo de instrumento una respuesta dada, aunque en este diagrama solo se describen las respuestas para los instrumentos formales.

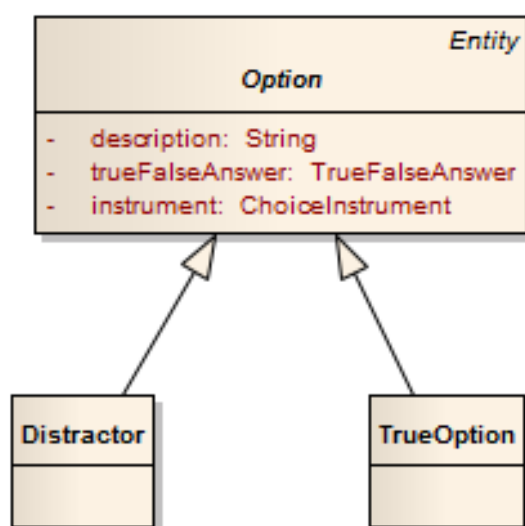


Figura 3.14 – Diagramas de opciones

En la figura 3.14 vemos el modelo de clases de las opciones que pueden presentar los instrumentos de selección (*ChoiceInstrument*).

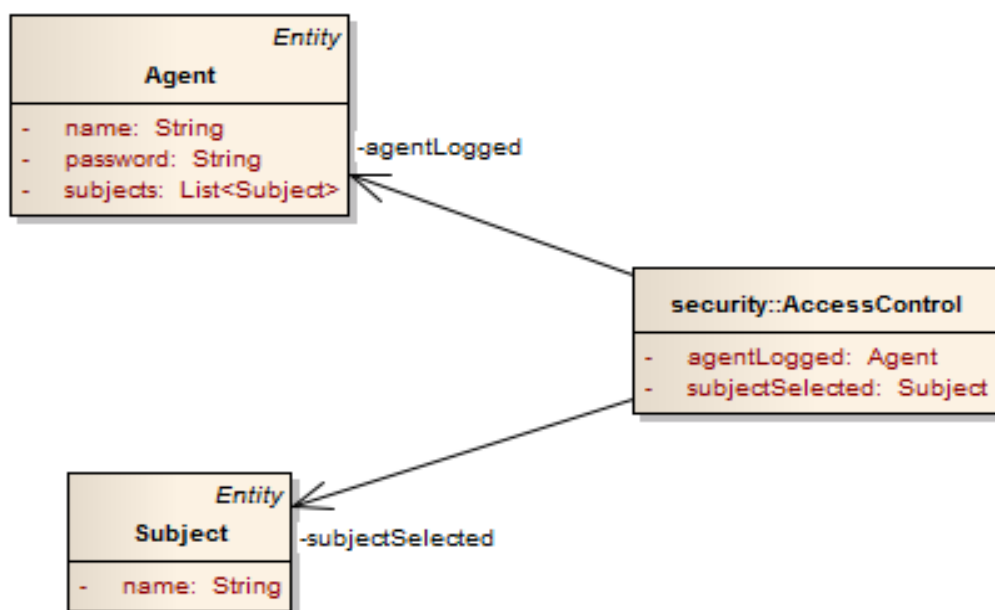


Figura 3.15 – Diagramas de acceso

En la figura 3.15 se ven las clases que son utilizadas para el control de acceso al sistema por parte de los agentes (*Agent*) y las materias (*Subject*) a las que está suscripto el mismo. La materia en la que se ingresa al sistema se utiliza para enlazar los materiales (evaluaciones, actividades, reactivos, instrumentos y conjuntos de reglas) y así poder administrarlos de manera global por asignatura.

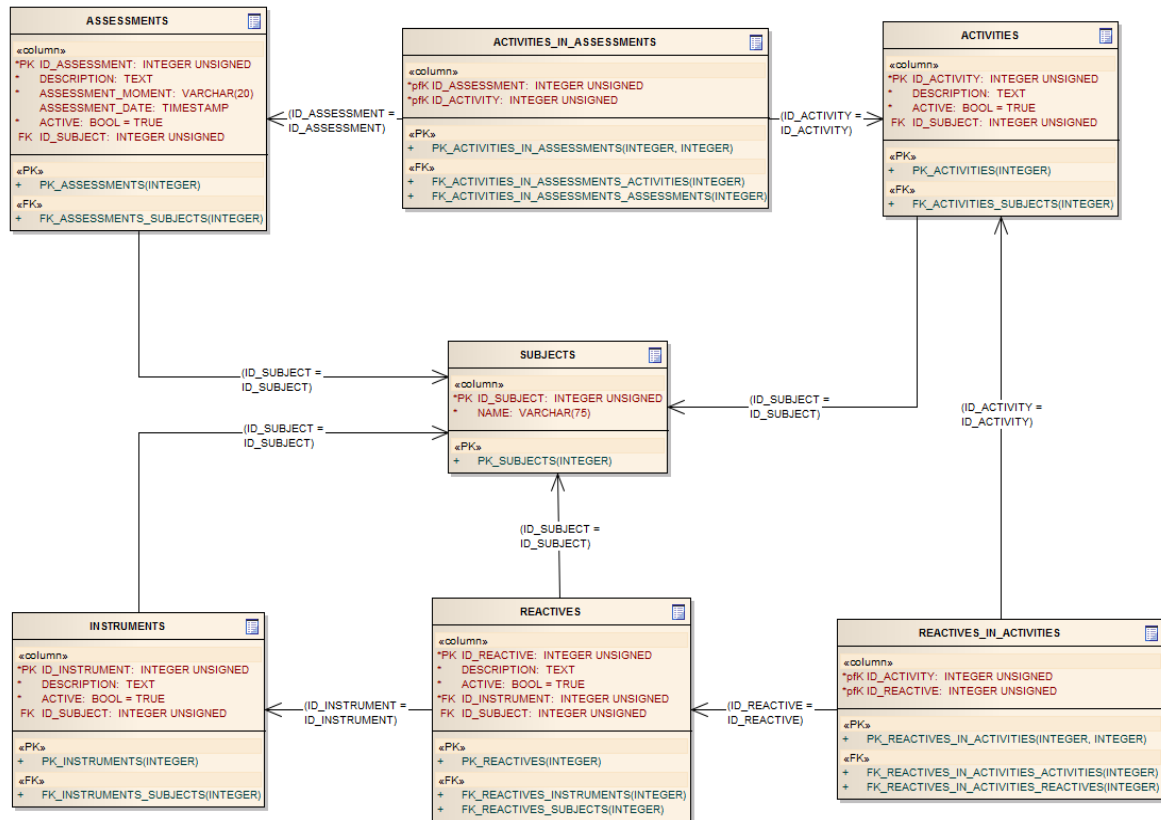


Figura 3.16 – Diagramas de reglas

En el diagrama de la figura 3.16 se presentan las clases que utilizamos para definir las reglas y sus conjuntos que vamos a utilizar para validar los modelos de ontologías que vamos a crear en memoria.

### 3.6 Modelo de Datos

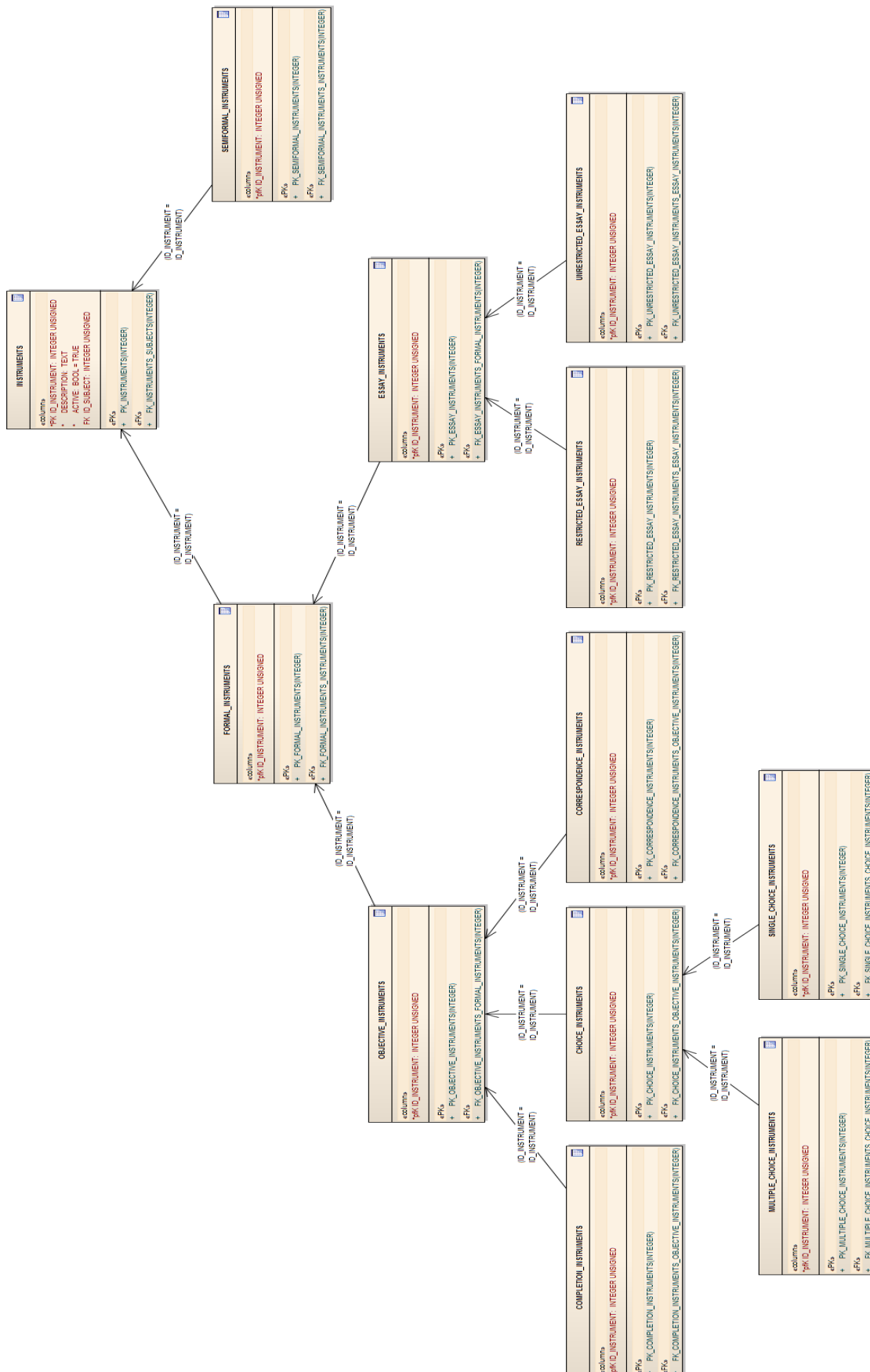
En esta sección vamos a mostrar el diagrama de base de datos que utilizamos para el sistema. Este esquema se desplegó dentro de un motor MySQL.



**Figura 3.17 - Modelo de datos de materiales**

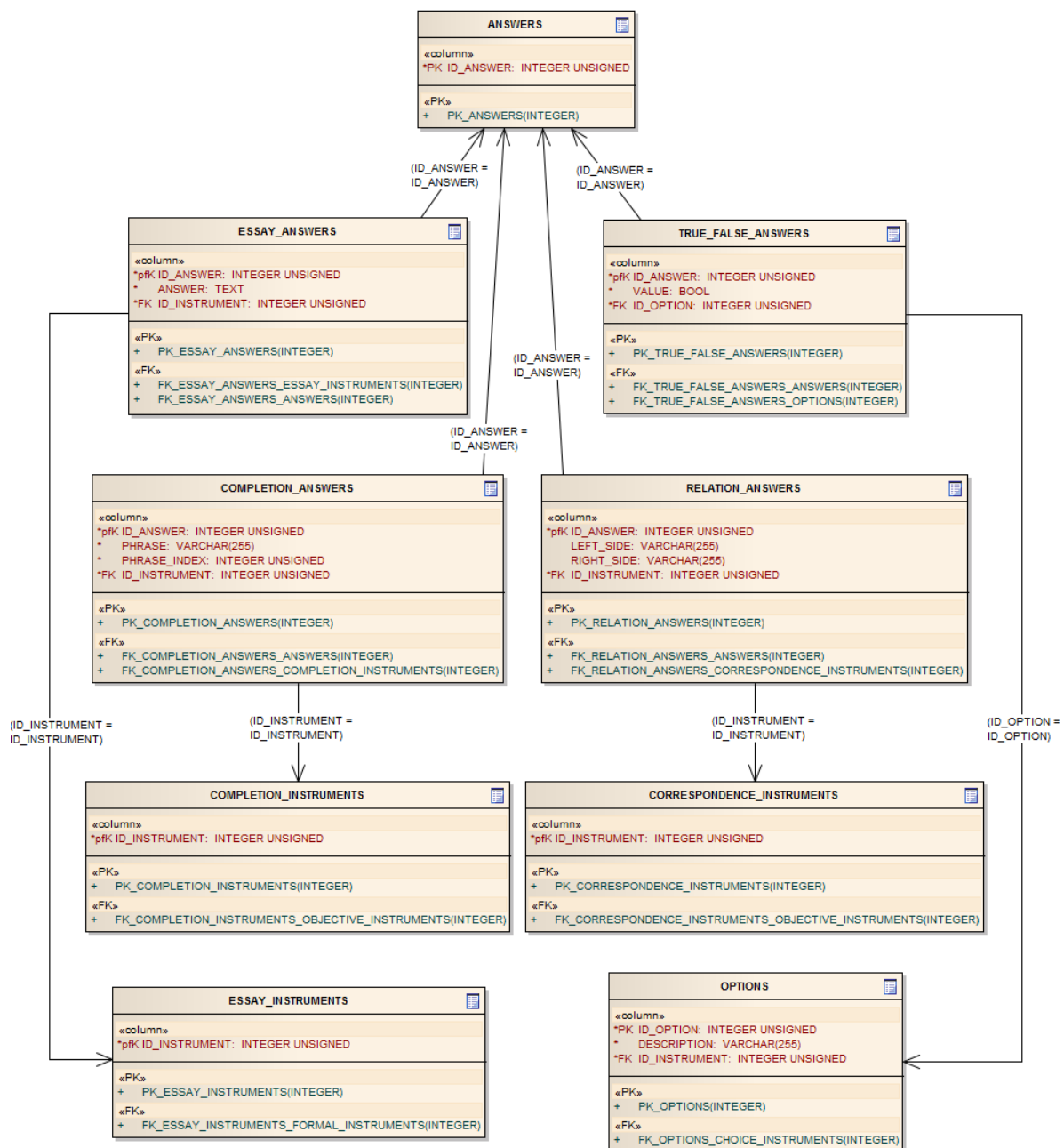
El esquema de datos se observa en la figura 3.17. El mismo se realizó a partir del modelo de ontología que utilizamos de base, por lo que posee una similitud a la misma.

El modelo de datos de materiales se compone de todas entidades, que tienen como fin representar los elementos que forman parte de las evaluaciones. Cada una de estas se caracteriza respecto a sus atributos y relaciones con las demás.



**Figura 3.18 - Modelo de datos de instrumentos**

El modelo de datos que se muestra en la figura 3.18, posee toda la información pertinente a los tipos de instrumentos que son almacenados en el sistema, para su futura utilización. Se dividen en dos grandes clases. “Formal\_instruments” y “Semiformal\_instruments”. Los primeros son la base fundamental donde se orientó el sistema y en los cuales se creó las herramientas para su construcción. Los semiformales, solo los mencionamos como tipos futuros a ser contruídos.



**Figura 3.19 - Modelo de datos de respuesta**

El modelo de datos de respuesta (figura 3.19), identifica las mismas para los instrumentos que componen las evaluaciones. Se dividen en distintos tipos como son: *ESSAY\_ANSWER*, *TRUE\_FALS\_ANSWER*, *COMPLETION\_ANSWER* y *RELATION\_ANSWER*. Las respuestas de verdadero y falsos se agrupan dentro del tipo de *OPTION*. Las respuestas de ensayo son respuestas dadas a una pregunta en concreto. Las respuesta de verdadero o falso son para las opciones de los instrumentos de selección. Las respuestas de completar identifican los “completion\_instruments” y las de relacionar representan las “correspondence\_instruments”

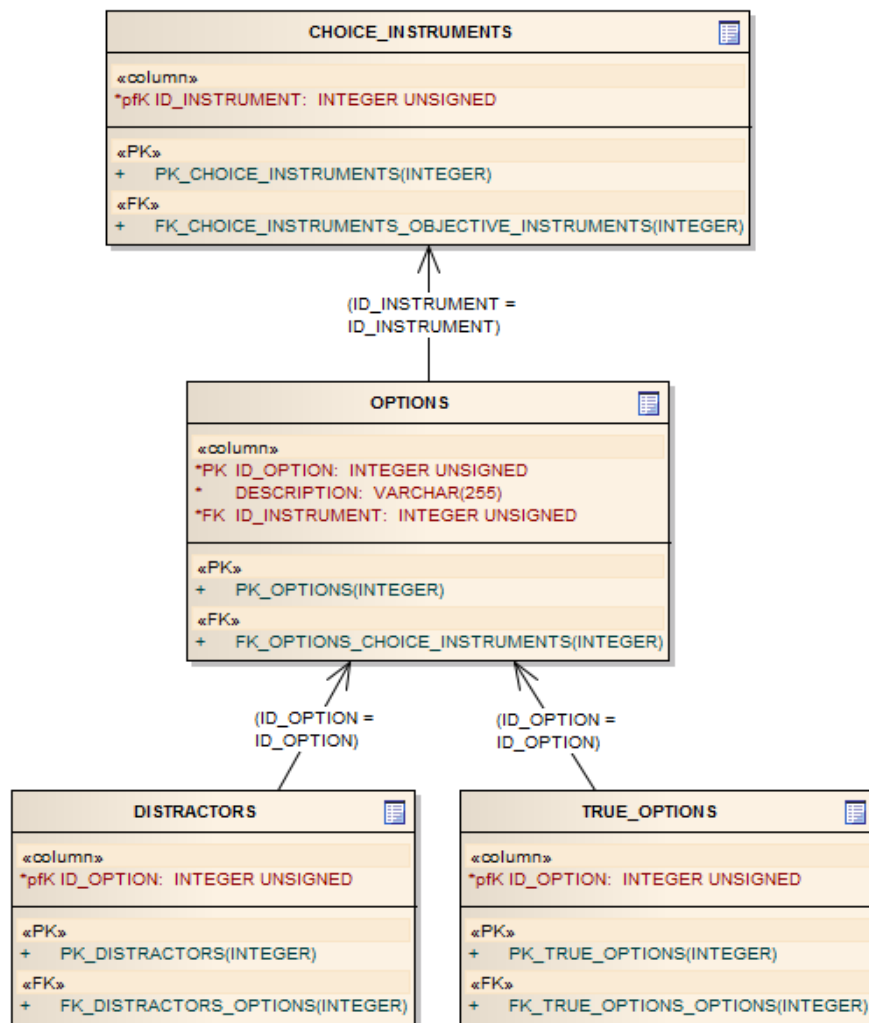
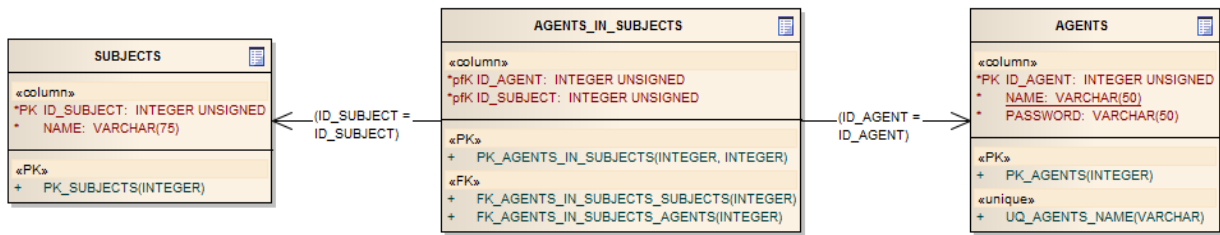


Figura 3.20 - Modelo de datos de opciones

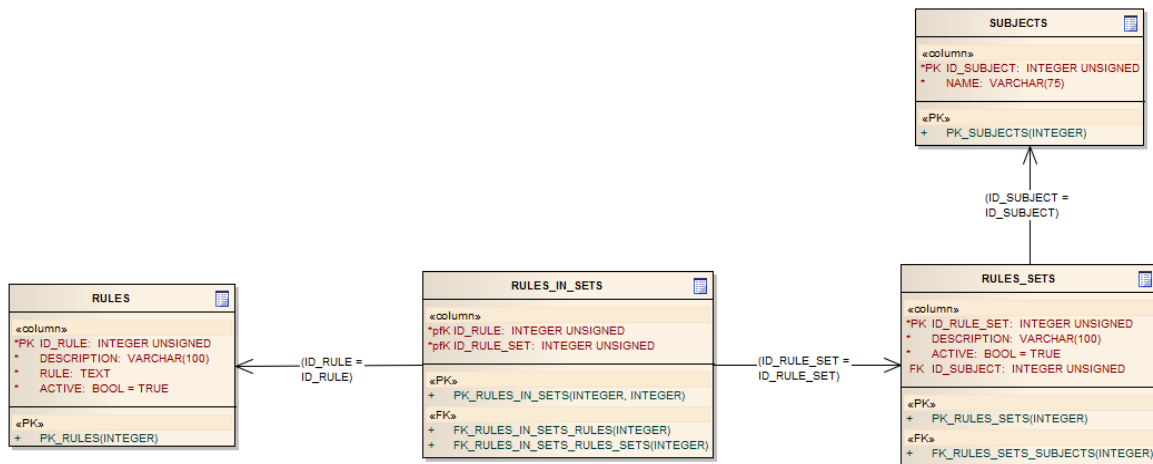
Las entidades que son parte del modelo de datos de opciones (figura 3.20), poseen la información pertinente a los instrumentos de selección, clasifican como opciones válidas los *distractors* y las *true\_options*.





**Figura 3.21 - Modelo de datos de acceso**

El modelo de datos de acceso (figura 3.21) contiene la información de las materias y sus respectivos agentes que la componen. Estos datos son necesarios para la autenticación del usuario al ingresar al sistema.



**Figura 3.22 - Modelo de datos de reglas y conjuntos**

Todo el conjunto de entidades que representa las reglas y los conjuntos (figura 3.22), son necesarios para la administración e interacción de dichas reglas para ser usadas en la validación de la evaluación.

### 3.7 Problemas surgidos sobre el trayecto del desarrollo

#### Problemas a nivel de diseño

En la etapa de diseñar el software, se fueron presentando varias dudas de como diseñar el sistema en base a las investigaciones realizadas. A continuación se enumeran los conflictos encontrados:

1. En un principio se desconocía el dominio que abarcaba el sistema, su entorno, su aplicación, su base para la implementación, sus componentes.
2. Desconocimientos en el ámbito de las ontologías, web semánticas, reglas lógicas que se tuvieron que aplicar.
3. Cambios de Requerimientos sobre el desarrollo del sistema. Los requerimientos se manifestaban en forma incremental.

#### Problemas a nivel de desarrollo

En esta fase se presentaron problemas muy complejos y que repercutieron principalmente en el tiempo.

Los problemas que se presentaron en el proyecto son:

1. A la hora de elegir un razonador para las reglas lógicas en el ámbito de las ontologías, en un principio adquirimos una librería de JESS, una versión universitaria, que nos fue enviada por la empresa, gracias a nuestro pedido.
2. Se leyó documentación, ejemplos, trabajos realizados, pero todo era muy escaso y no dejaban claro su aplicación. Se decidió a nivel de grupo en conjunto con las directoras, seguir investigando por otra herramienta. Luego de intensas búsquedas se optó por utilizar la librería JENA.
3. A lo que refiere a la programación, los framework e IDEs como java, maven, hibernate, spring, fueron optados por su gran potencial y manejo de los datos en la actualidad, permitiendo una implementación más óptima. El problema surgido en todo esto fue que en su momento no eran herramientas tan familiares, donde tuvimos que destinar tiempo a la investigación y prueba de los entornos de programación anteriormente mencionados.
4. La forma en cómo se iban a guardar las ontologías en base de datos, fue otro punto de decisión, a nivel de diseño e implementación. El problema

principalmente fue el desconocimiento en el dominio de las ontologías. En un estado avanzado sobre la investigación nos llevó a la conclusión de que la manipulación de la ontología en un archivo de texto RDF, ocupa mucho mayor espacio en memoria y tiempo de procesamiento, por lo que optamos por crear ontologías directamente en memoria mediante un Framework propio.

### Problemas a nivel de pruebas

Las pruebas correspondieron a la interacción del sistema, semanalmente por las directoras del proyecto y diariamente por los integrantes del grupo de desarrollo del proyecto, donde se efectuaban diferentes pruebas, planteadas a distintos niveles, con el fin de optimizar y crear un sistema seguro y sin errores.

A continuación detallamos problemas surgidos en las respectivas pruebas:

1. Desde los roles de desarrolladores, las pruebas se hacían a medida que se termina una funcionalidad. Las mismas constituían testeo del código y casos de pruebas al azar. Esto implicó la falta de pruebas más estrictas y casos de pruebas más variados. Varios de los errores fueron encontrados con el uso por partes de las directoras y comunicados a nosotros.
2. La gran cantidad de cambios y nuevos requerimientos surgidos por pedido de las directoras o por ideas que se fueron dando en el desarrollo del sistema, provocaron cambios en interfaces, funciones, servicios, donde la demanda de pruebas crecieron y se acomplexaron.
3. Otros tipos de problemas son los surgidos de Framework, IDEs y base de datos seleccionados para el proyecto, que sumando que en su momento no había un dominio completo de esas herramientas, fueron condicionantes en las pruebas realizadas.

## 3.8 Conclusiones

Este capítulo presentó el análisis y diseño de la arquitectura sobre la que se desarrolló el software. Se hizo una descripción de la metodología utilizada y se especificaron los pasos y actividades dentro de cada etapa de la misma. Se mostró la herramienta utilizada para la correcta gestión de la aplicación de la metodología. Se describieron los casos de uso que fueron tenidos en cuenta para la definición de los requerimientos de usuario y se mostraron

los diagramas de clases que describen al sistema. También se presentó la organización de las bases de datos utilizadas en el desarrollo del proyecto. Toda esta documentación pertinente queda plasmada para los futuros usuarios o desarrolladores que quieran expandir o mejorar el sistema.

## Capítulo 4 - Tecnologías

Este capítulo presenta una serie de herramientas y tecnologías, que sirvieron de motores para la manipulación de la ontología creada, por un lado, y por otro lado se describe brevemente los Framework y tecnologías seleccionadas apropiadamente para el desarrollo del sistema. Todo esto fue acompañado de problemas generales que fueron surgiendo, estos también serán descriptos en el capítulo.

### 4.1 Manipulación y creación de ontologías

La creación y manipulación de ontologías dentro del sistema se va a realizar por completo en memoria y tomando solo los elementos necesarios para no sobrecargar la misma y caer en una baja de performances por evaluar elementos que no son requeridos.

Para poder realizar esto vamos a hacer uso de un Framework conocido en el ámbito del desarrollo, denominado Apache Jena.

Jena es un Framework Open Source para la Web Semántica desarrollado completamente en lenguaje JAVA. Provee una API para extraer datos o escribirlos dentro de grafos RDF. Estos grafos son representados como un modelo abstracto. Este modelo puede cargarse a partir de archivos, base de datos, URLs o una combinación de estas fuentes. Estos modelos pueden ser consultados mediante reglas SPARQL y actualizados mediante SPARUL.

Jena también posee su propio conjunto de reglas de inferencia denominadas *Jena Rules*. Estas reglas poseen una construcción basada en las reglas SWRL para consulta y validación de modelos OWL.

El subsistema de inferencia Jena está diseñado para permitir una amplia gama de motores de inferencia o razonadores que se conecta a Jena. Estos motores se utilizan para derivar afirmaciones RDF adicionales que se implicaron de alguna base de RDF junto con cualquier información de ontología opcional y los axiomas y las reglas asociadas con el razonador. El uso principal de este mecanismo es apoyar el uso de los lenguajes como RDFS y OWL que permiten que nuevos hechos se deduzcan de los datos de las instancias y las descripciones de clases. Sin embargo, esta herramienta está diseñada para ser bastante general y, en particular, incluye un motor de reglas genérico que se puede utilizar para muchos procesamiento RDF o

tareas de transformación. Este motor de inferencia genérico es el que vamos a utilizar para nuestro sistema ya que nos permite utilizar reglas muy parecidas a las reglas SWRL.

Vamos a usar el término inferencia para hacer referencia al proceso de obtener nueva información derivada de la información conocida y el término razonador para referirse a un objeto de código específico que lleva a cabo esta tarea.

La estructura general de la maquinaria de inferencia de Jena se ilustra en la figura 4.1.

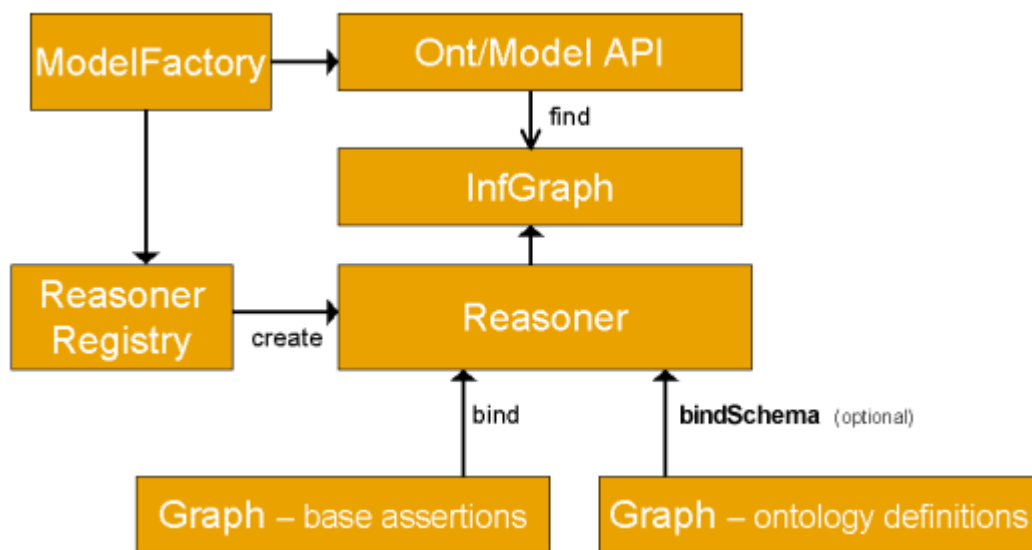


Figura 4.1 - Modelo de componentes de inferencia de JENA

Las aplicaciones normalmente acceden a la máquina de inferencia utilizando la clase *ModelFactory* para crear y asociar un conjunto de datos con algún modelo de ontología que va a utilizarse como base de conocimiento por parte del razonador. Este modelo se puede crear a partir de archivos, conexiones a una base de datos, URLs o una mezcla de estas fuentes. Nosotros en este caso, creamos la ontología propiamente dicha en memoria para no perder tiempo y espacio en una ontología que vamos a utilizar para validar una evaluación y luego ésta se descarta. Si a ese modelo creado se lo enlaza a un motor de inferencia, se crearán los datos derivados de este motor, por lo que las consultas al modelo creado regresarán, no sólo las declaraciones que estaban presentes en los datos originales, sino también las declaraciones adicionales que se pueden derivar de los datos a través de las normas u otros mecanismos de inferencia aplicados por el razonador.

Como se ilustra en el gráfico, la máquina de inferencia es implementada al nivel de un grafo SPI, de modo que cualquiera de las diferentes interfaces de modelo puede ser construido

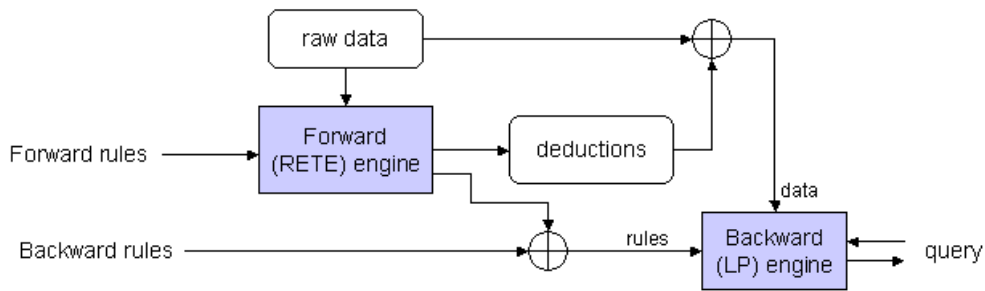
en torno a un grafo de inferencia. En particular, esta API ofrece métodos convenientes para vincular razonadores apropiados en los modelos de ontologías (*OntModels*) que son construidos. Como parte general de la API RDF, también se proporciona un modelo de inferencia (*InfModel*), el cual es una extensión de la interfaz normal del modelo que proporciona un control adicional y el acceso a un gráfico de inferencia subyacente.

La API del razonador apoya la idea de la especialización, ya que este enlaza a un conjunto de datos de esquema u ontología utilizando una llamada *bindSchema*. El razonador especializado entonces se puede conectarse a diferentes conjuntos de datos de la instancia, utilizando llamadas de esta clase. Hay situaciones en que la misma información de esquema es utilizada múltiples veces con diferentes conjuntos de datos de la instancia, esta técnica permite un poco de reutilización de inferencias a través de los diferentes usos del esquema. En RDF no hay una fuerte separación entre esquema de datos y datos de instancia y cualquier datos, ya sea de clase o instancia relacionados, pueden ser incluidos en cualquiera de los dos el enlace o llamadas *bindSchema*.

Para mantener el diseño lo más flexible posible Jena también incluye un *ReasonerRegistry*. Esta es una clase estática a través del cual el conjunto de razonadores actualmente disponibles puede ser examinado. Es posible registrar nuevos tipos de razonadores, y buscar dinámicamente razonadores de un tipo determinado. La clase *ReasonerRegistry* también proporciona un cómodo acceso a las instancias de pre-compilados de los principales razonadores suministrados. (**Documentación JENA**).

### Razonador de propósito general de JENA

Jena incluye un razonador basado en reglas de propósito general que es utilizado para implementar los razonadores de RDFS y OWL. Este razonador soporta inferencia basada en reglas sobre grafos RDF y provee inferencia por encadenamiento hacia delante (dirigido por los datos), hacia atrás (dirigido por las metas) y un modelo híbrido de ejecución. Para ser más exacto, este razonador posee 2 motores de inferencia (ver figura 4.2), uno para el encadenamiento hacia adelante (RETE) y otro motor de encadenamiento hacia atrás (LP). Estos pueden ser corridos separadamente o bien el motor de encadenado hacia adelante puede usarse para mejorar el motor de encadenado hacia atrás que es arrancado para responder consultas.



**Figura 4.2 - Razonador de propósito general**

Las configuraciones variadas de este motor son todas accesibles a través de un conjunto simple de parámetros. Como mínimo este razonador requiere un conjunto de reglas para definir su comportamiento. La instancia de este razonador con el conjunto de reglas puede ser usado como cualquier otro razonador. Esto puede ser enlazado al modelo de datos y usado para contestar consultas lo que resulta en un modelo de inferencia.

El razonador también puede extenderse para registrar nuevos procedimientos primitivos. El razonador actual incluye un conjunto de primitivas que son suficientes para implementaciones RDFS y OWL y son fácilmente extensibles.

### Sintaxis y estructuras de reglas

Una regla para este razonador está definida por un objeto *Rule* en JAVA que tiene un conjunto de premisas (cuerpo), una lista de conclusiones (cabeza) y un nombre y dirección opcional. Cada término de ésta, es definido por la clase *ClauseEntry*, la cual es un patrón triple de elementos, una llamada o una función primitiva. Un conjunto de reglas es simplemente una lista de reglas.

El parseador de reglas permite ser configurado para poder leer reglas desde archivos de textos, produciendo como salida un conjunto de reglas.

Una descripción informal de la sintaxis de una regla se muestra en el siguiente cuadro:

**Rule** := *bare-rule*  
           o [ *bare-rule* ]  
           o [ *ruleName* : *bare-rule* ]

**bare-rule** := *term*, ... *term* -> *hterm*, ... *hterm*  
               o *bhterm* <- *term*, ... *term*

**hterm** := *term*  
           o [ *bare-rule* ]



**term** := (node, node, node)  
           o (node, node, functor)  
           o builtin(node, ... node)

**bhterm** := (node, node, node)

**functor** := functorName(node, ... node)

**node** := uri-ref  
           o prefix:localname  
           o <uri-ref>  
           o ?varname  
           o 'a literal'  
           o 'lex'^^typeURI  
           o number

El separador “,” es opcional en las reglas.

La diferencia entre la sintaxis de las reglas para encadenamiento hacia adelante y hacia atrás es solo relevante para la estrategia de ejecución híbrida.

Las funciones (o builtIn) propios del razonador de propósito general son las siguientes:

Función	Descripción
isLiteral(?x), notLiteral(?x) isFunctor(?x), notFunctor(?x) isBNode(?x), notBNode(?x)	Evalúa si el argumento es un elemento en particular.
bound(?x...) unbound(?x...)	Evalúa si todos los argumentos están enlazados.
equal(?x,?y) notEqual(?x,?y)	Evalúa si x==y o x != y.
lessThan(?x, ?y) greaterThan(?x, ?y) le(?x, ?y) ge(?x, ?y)	Evalúa si x es <, >, <= o >= y.
sum(?a, ?b, ?c) addOne(?a, ?c) difference(?a, ?b, ?c) min(?a, ?b, ?c) max(?a, ?b, ?c) product(?a, ?b, ?c) quotient(?a, ?b, ?c)	Carga en c el valor de (a+b), (a+1) (a-b), min(a,b), max(a,b), (a*b), (a/b).

<i>strConcat(?a1, .. ?an, ?t)</i> <i>uriConcat(?a1, .. ?an, ?t)</i>	Concatena el léxico de todos los argumentos, excepto el último, al que va a cargar esta concatenación.
<i>regex(?t, ?p)</i> <i>regex(?t, ?p, ?m1, .. ?mn)</i>	Permite validar si un léxico encaja en una expresión regular.
<i>now(?x)</i>	Enlaza ?x el valor correspondiente del tiempo actual.
<i>makeTemp(?x)</i>	Enlaza ?x a un nodo en blanco.
<i>makeInstance(?x, ?p, ?v)</i> <i>makeInstance(?x, ?p, ?t, ?v)</i>	Enlaza ?v para ser un nodo en blanco que tiene para ser incluido como el valor de la propiedad ?p en el recursos ?x y tipo ?t.
<i>makeSkolem(?x, ?v1, .. ?vn)</i>	Enlaza ?x para ser un nodo en blanco. Este es creado con los valores restantes de los argumentos ?vi
<i>noValue(?x, ?p)</i> <i>noValue(?x ?p ?v)</i>	Verdadero si no hay una tripleta (x, p, *) o (x, p, v).
<i>remove(n, ...)</i> <i>drop(n, ...)</i>	Elimina el elemento (tripleta).
<i>isDType(?l, ?t)</i> <i>notDType(?l, ?t)</i>	Evalúa si el literal ?l es o no es una instancia del tipo ?t.
<i>print(?x, ...)</i>	Imprime por salida estándar los argumentos.
<i>listContains(?l, ?x)</i> <i>listNotContains(?l, ?x)</i>	Evalúa si el elemento ?x esta contenido (o no) dentro de la lista ?l.
<i>listEntry(?list, ?index, ?val)</i>	Enlaza el valor ?val al índice ?index de la lista ?list.
<i>listLength(?l, ?len)</i>	Enlaza ?len a la longitud de la lista ?l.
<i>listEqual(?la, ?lb)</i> <i>listNotEqual(?la, ?lb)</i>	Evalúa si 2 listas son iguales o no.
<i>listMapAsObject(?s, ?p ?l)</i> <i>listMapAsSubject(?l, ?p, ?o)</i>	Estas funciones solo pueden usarse en la cabecera de la regla. Permiten crear un conjunto de tripletas derivadas de la lista de argumentos.
<i>table(?p)</i> <i>tableAll()</i>	Declara que todos los objetivos que involucren la propiedad ?p deberían ser tabulados en un motor de encadenamiento hacia atrás
<i>hide(?p)</i>	Declara que las tripletas que involucren al predicado ?p deberían ocultarse.

## 4.2 Herramientas y Frameworks

En el transcurso de desarrollo del proyecto hicimos uso de distintas herramientas, algunas con conocimientos previos y otras sin conocimiento. La combinación de ambas nos llevó a pruebas y distintos experimentos, que formaron parte del tiempo destinados a la investigación.

A continuación se hace referencia a las herramientas, Frameworks y base de datos, utilizadas en el proyecto, detallando sus funcionalidades.

### Eclipse (IDE)

Eclipse es un programa informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Es un entorno de desarrollo integrado (del inglés IDE) que incluye llamado *Java Development Toolkit* (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). **(Documentación Eclipse).**

A este IDE se le adaptaron un conjunto de plugin necesarios para agilizar el desarrollo. Estos plugins abarcan:

### Maven

Maven utiliza un archivo de configuración denominado "*Project Object Model*" (POM) para describir el proyecto de software a construir, sus dependencias de otros módulos y componentes externos, y el orden de construcción de los elementos. Viene con objetivos predefinidos para realizar ciertas tareas claramente definidas, como la compilación del código y su empaquetado.

Una característica clave de Maven es que está listo para usar en red. El motor incluido en su núcleo puede dinámicamente descargar plugins de un repositorio, el mismo repositorio que provee acceso a muchas versiones de diferentes proyectos Open Source en Java, de Apache y otras organizaciones y desarrolladores **(Documentación Maven).**

### Hibernate

Hibernate es una herramienta de Mapeo objeto-relacional (ORM) para la plataforma Java, que facilita el mapeo de atributos entre una base de datos relacional tradicional y el

modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los POJOS (*Plain Old Java Objects*) de las entidades que permiten establecer esas relaciones.

Hibernate genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución **(Documentación Hibernate)**.

### Spring

Spring es un framework para el desarrollo de aplicaciones y contenedor de inversión de control, de código abierto para la plataforma Java.

Si bien las características fundamentales de Spring Framework pueden ser usadas en cualquier aplicación desarrollada en Java, existen variadas extensiones para la construcción de aplicaciones web sobre la plataforma Java EE. A pesar que no impone ningún modelo de programación en particular, este framework se ha vuelto popular en la comunidad al ser considerado una alternativa, sustituto, e incluso un complemento al modelo EJB (Enterprise JavaBean) **(Documentación Spring)**.

### TortoiseSVN

TortoiseSVN es un cliente gratuito de código abierto para el sistema de control de versiones *Subversión*.

TortoiseSVN maneja ficheros y directorios a lo largo del tiempo. Los ficheros se almacenan en un *repositorio* central. El repositorio es prácticamente lo mismo que un servidor de ficheros ordinario, salvo que recuerda todos los cambios que se hayan hecho a sus ficheros y directorios.

Esto permite que pueda recuperar versiones antiguas de sus ficheros y examinar la historia de cuándo y cómo cambiaron sus datos, y quién hizo el cambio **(Documentación Tortoise SVN)**.

### Base de datos: MySQL

MySQL es un sistema de administración de bases de datos (*Database Management System, DBMS*) para bases de datos relacionales. Así, MySQL no es más que una aplicación que permite gestionar archivos de bases de datos **(Documentación MySQL)**.

### **4.3 Conclusiones**



En este capítulo, se presentaron las distintas tecnologías utilizadas para la implementación del software. Se describieron las mismas especialmente aquellas utilizadas para la implementación de las ontologías, las reglas lógicas y las posibles inferencias que se logra con las mismas.





## Capítulo 5 - Funcionamiento del Sistema OFGA

En este capítulo se procederá a explicar el funcionamiento integral del sistema a través de la presentación de interfaces visuales para familiarizar a los usuarios con el diseño e interoperabilidad brindada por el sistema.

La tabla 5.1 muestra los componentes generales y sus funcionamientos que se pueden apreciar en las ventanas del sistema.

### 5.1 Tabla de Referencia de Componentes

Icono	Componente	Descripción
	Botón Aceptar Cambios	Permite aceptar y guardar todos los cambios hechos por el usuario. Dichos cambios se verán reflejados en la base de datos.
	Botón Cancelar Cambios	Permite cancelar cualquier cambio hecho en la interfaz, descartando el contenido de la misma y produciendo su cierre.
	Botón Agregar Elemento	Permite crear o agregar nuevos elementos a una lista de entidades.
	Botón Editar Elemento	Permite realizar modificaciones a los atributos del elemento seleccionado de una lista.
	Botón Eliminar Elemento	Permite quitar un elemento de una lista.
	Botón Asignar Elemento	Permite asignar un elemento seleccionado de una lista al elemento padre que lo compone.
	Indicador de Progreso	Le indica al usuario de un bloqueo en la interfaz gráfica, debido a acciones que se ejecutan en segundo plano, como accesos a la base de datos o procesamiento de validación de la ontología.
	Botón Evaluar	Permite crear un proceso de validación para una evaluación, creando la ontología donde va a cargarse la

		misma para luego evaluarla.
	Botón Limpiar	Permite limpiar el área de resultado donde se despliega la validación de una evaluación dada.
	Botón Cargar Elementos	Permite la carga elementos a una lista para su posterior utilización.
	Botón PDF	Permite generar un documento PDF con la evaluación y todos sus elementos que la componen.
	Botón de administración de elementos	Permite administrar los listados de los elementos que tenemos dentro del sistema.
<b>C</b>	Botón de clases de la ontología	Despliega una ventana donde se va a poder seleccionar el nombre de una de las clases que tenemos dentro de la ontología para permitirnos una mejor forma de crear una regla.
<b>P</b>	Botón de propiedades de la ontología	Despliega una ventana donde se va a poder seleccionar el nombre de una de las propiedades de las clases que tenemos dentro de la ontología para permitirnos una mejor forma de crear una regla.
<b>E</b>	Botón de sentencia de error para reglas	Permite autocompletar una regla agregando en la posición donde se encuentra el cursor, la cláusula de error que va a fallar al momento en el que se evalúe la ontología creada con la evaluación.

## 5.2 Interfaces Gráficas

### Acceso al sistema

El sistema se inicia con la ventana de acceso al sistema, como parte de seguridad del sistema. En la figura 5.1 se puede observar dicha ventana.

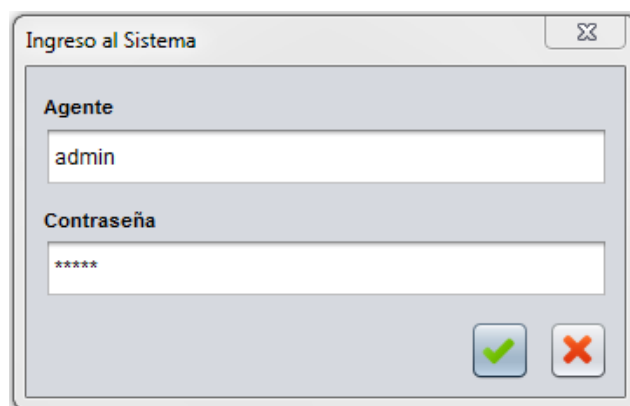


Figura 5.1 - Ingreso al sistema

El usuario ingresa su nombre que se lo identifica como agente del sistema y su contraseña y oprime el botón de aceptar.

Si algún dato no corresponde, ya sea porque el nombre ingresado no es válido o la contraseña es inválida se muestran alguno de los siguientes mensajes:

Nivel de error	Falla	Mensaje
Medio	El nombre del agente no se encuentra registrado en el sistema o es un nombre vacío.	<i>"El nombre del agente es incorrecto"</i>
Medio	La contraseña cargada es vacía o no corresponde con el nombre de usuario cargado.	<i>"La contraseña es incorrecta"</i>



## Selección de Materia

Luego de un ingreso exitoso al sistema por parte del usuario, este debe seleccionar la materia donde va gestionar sus evaluaciones. En la figura 5.2 se observa el menú de selección de materias.

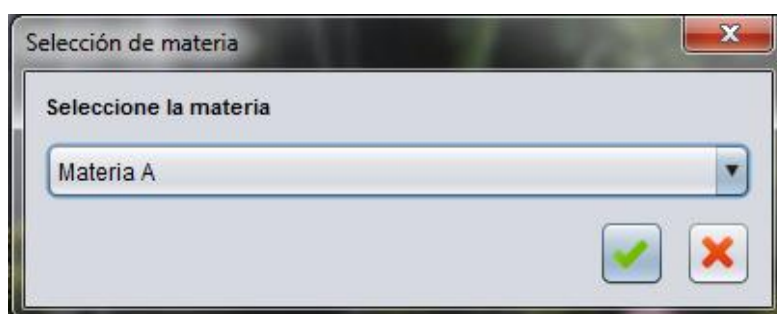


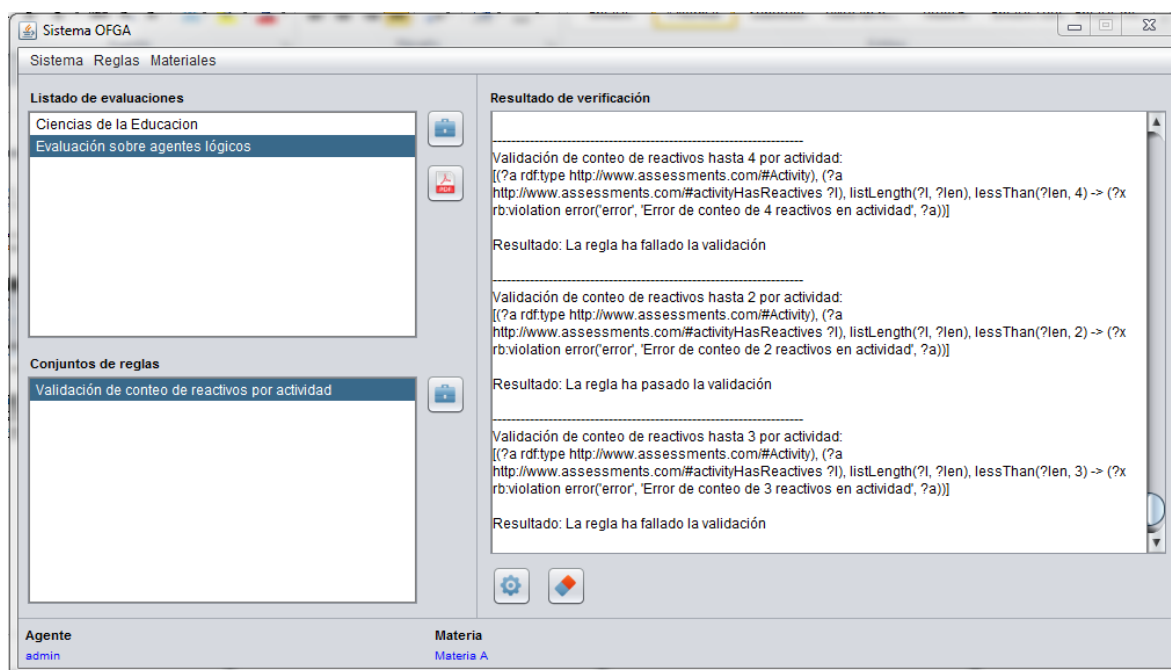
Figura 5.2 - Selección de materia

En caso de que se presione aceptar si seleccionar ninguna materia, se mostrará el siguiente mensaje:

Nivel de error	Falla	Mensaje
Medio	No se selecciona ninguna materia en la ventana.	<i>“Debe seleccionar una materia”</i>

## Menú Principal

Cuando se selecciona una materia se accede de inmediato al menú principal de gestión de evaluaciones, esta ventana se aprecia en la figura 5.3:








**Figura 5.3 - Ventana principal**

En esta ventana se cuenta con una barra de menús que contiene acceso a la mayoría de las partes del sistema, estos menús son los siguientes:

Menú	Submenú	Descripción
<b>Sistema</b>	Cambio de materia	Permite cambiar la materia en la que se encuentra registrado el agente.
	Salir	Permite salir del sistema mediante una confirmación por parte del agente.
<b>Reglas</b>	Administrar conjuntos de reglas	Despliega una ventana donde tenemos el listado de los conjuntos de reglas para poder crear nuevos, editar ya existentes o borrarlos, como así también administrar las reglas del sistema.
<b>Materiales</b>	Administrar evaluaciones	Despliega una ventana donde tenemos un listado de las evaluaciones que hay cargadas en la materia donde ingresamos para poder crear nuevas, editar o borrar evaluaciones existentes.

	Administrar actividades	Despliega una ventana donde tenemos un listado de las actividades que hay cargadas en la materia donde ingresamos para poder crear nuevas, editar o borrar evaluaciones existentes.
	Administrar reactivos	Despliega una ventana donde tenemos un listado de los reactivos que hay cargados en la materia donde ingresamos para poder crear nuevas, editar o borrar evaluaciones existentes.
	Administrar instrumentos	Despliega una ventana donde tenemos un listado de los instrumentos que hay cargados en la materia donde ingresamos para poder crear nuevas, editar o borrar evaluaciones existentes.

Además de la barra de menús, la ventana del menú principal se estructura de la siguiente manera:

- **Listado de evaluaciones:** representan los exámenes creados para la materia en la que el agente eligió ingresar. A la derecha se sitúa el botón de administración de las evaluaciones [  ]. Bajo este botón se encuentra el botón que permite crear un archivo PDF con la evaluación que tenemos seleccionada [  ].
- **Listado de Conjunto de Reglas:** Representan el agrupamiento de ciertas reglas que serán utilizados para validar la evaluación seleccionada en la lista anteriormente mencionada. A su derecha se encuentra el botón de administración de los conjuntos de reglas [  ].
- **Resultado de verificación:** En esta área se va a cargar el detalle informativo de haber aplicado el conjunto de reglas seleccionado del listado de conjuntos a la evaluación seleccionada del conjunto. Para correr el proceso de validación tenemos el botón de validación [  ], al presionarlo este se deshabilita y comienza a correr el proceso de validación y se va cargando esta área de resultado con los datos del proceso. Una vez finalizado el proceso, este botón vuelve a habilitarse. El botón de borrado del área de resultado [  ] limpia el contenido de esta área.

En la parte baja de la ventana principal se encuentra la información de la materia que se seleccionó para administrar.

## Administración de evaluaciones

Para ingresar a la administración de evaluaciones se puede hacer mediante la barra de menú de la ventana principal:

*Material -> Administrar evaluaciones*

O sino también mediante el botón de administración de evaluaciones de la ventana principal.

En la opción de administrar las evaluaciones del sistema se nos presenta la siguiente ventana:

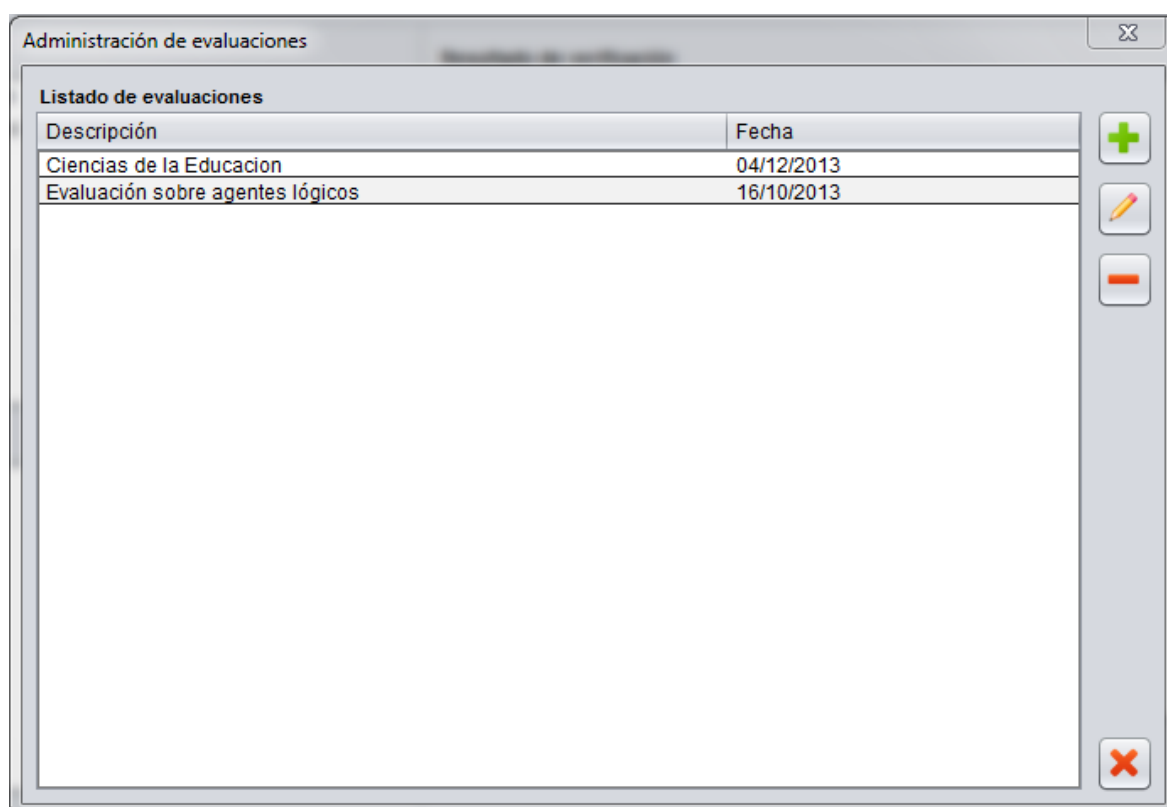


Figura 5.4 - Gestión de evaluaciones

Dentro de la ventana de administración se puede agregar nuevas evaluaciones a través del botón de agregar [ + ], donde nos llevará a una ventana de selección de tipo de evaluación como la siguiente:

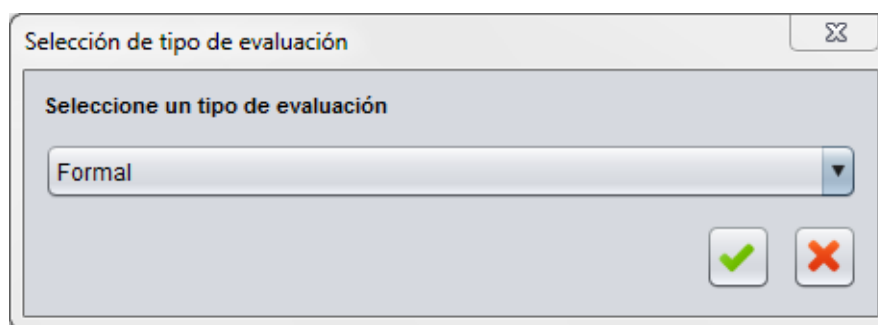


Figura 5.5 - Selección de tipo

Cuando seleccionamos el tipo de evaluación que queremos crear y apretamos aceptar, nos va a aparecer una ventana como la siguiente:

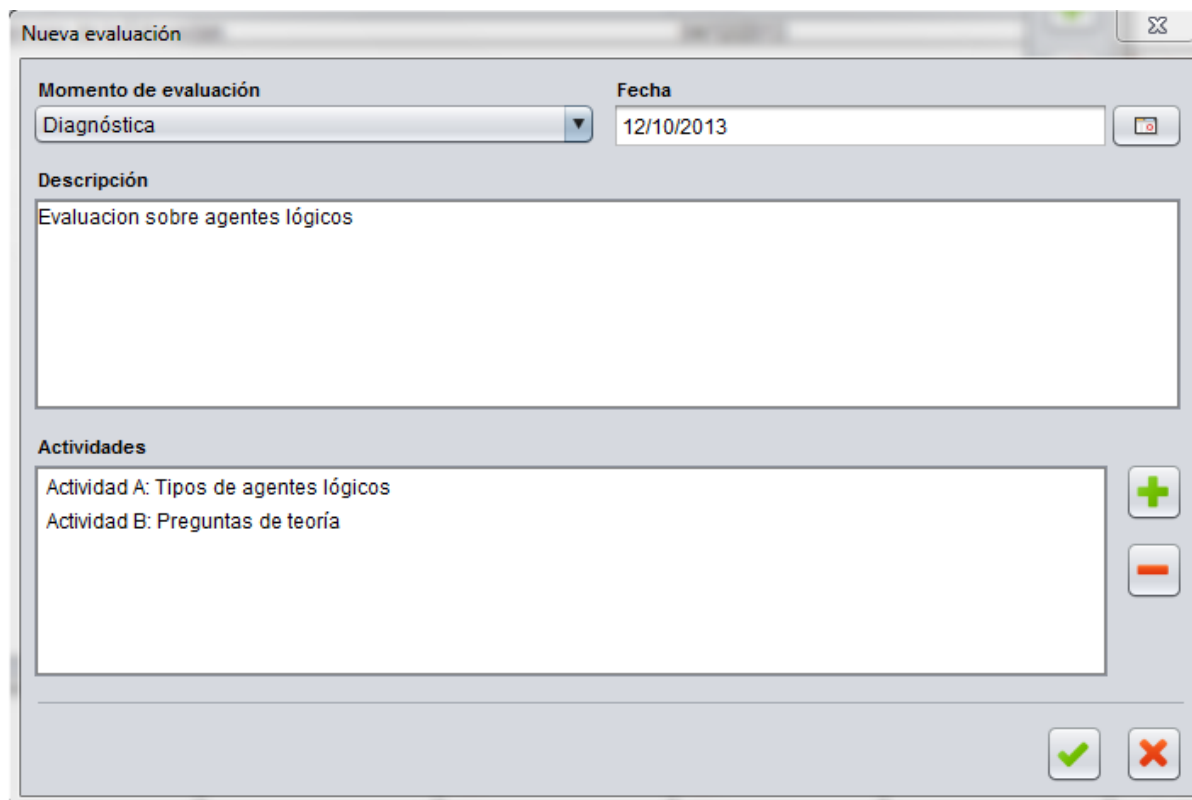




Figura 5.6 - Alta de evaluación

Para editar una evaluación se debe seleccionar una desde el listado de evaluaciones y al presionar el botón de edición [  ] se nos va a desplegar una ventana similar a la de la figura 5.6 pero ya con los campos cargados con los datos de la evaluación a editar.



Para eliminar también se debe seleccionar una evaluación y presionar el botón de borrado [  ], el cual despliega un mensaje de confirmación de borrado para prevenir borrados indeseados.

En la figura 5.6 se observan los atributos necesarios para crear la evaluación y al momento de guardarla se verifican que todos los campos tengan valores cargados, en caso de que alguno de ellos no tenga datos se desplegará alguno de los siguientes mensajes:

Nivel de error	Falla	Mensaje
Alto	No se selecciona ningún momento de la evaluación	<i>“Debe seleccionar un tipo de evaluación”</i>
Alto	No se introdujo ninguna fecha a la evaluación	<i>“Debe colocar una fecha de evaluación”</i>
Alto	La descripción de la evaluación está vacía	<i>“La descripción de la evaluación no puede ser vacía”</i>
Alto	No se le cargó ninguna actividad a la evaluación	<i>“Debe al menos seleccionar una actividad para la evaluación”</i>

Volviendo a la ventana de evaluación, figura 5.6, se puede apreciar el botón de agregar



[  ] que nos permite ingresar a la ventana de administración de actividades, figura 5.7, aunque con la diferencia que el botón de selección de elementos [  ] se encuentra habilitado para poder tomar las actividades. El listado de las actividades de la evaluación permite cargar un conjunto no definido de actividades, por lo que podemos seleccionar varias y cargarlas de una sola vez dentro de esta.

## Administración de actividades

Para ingresar a la administración de actividades se puede hacer mediante la barra de menú de la ventana principal:

*Material -> Administrar de actividades*

O mediante el botón de agregar elementos de la ventana de edición de evaluaciones, aunque de esta manera va a tener habilitado el botón de selección de elementos.

La ventana de administración de actividades es la siguiente:

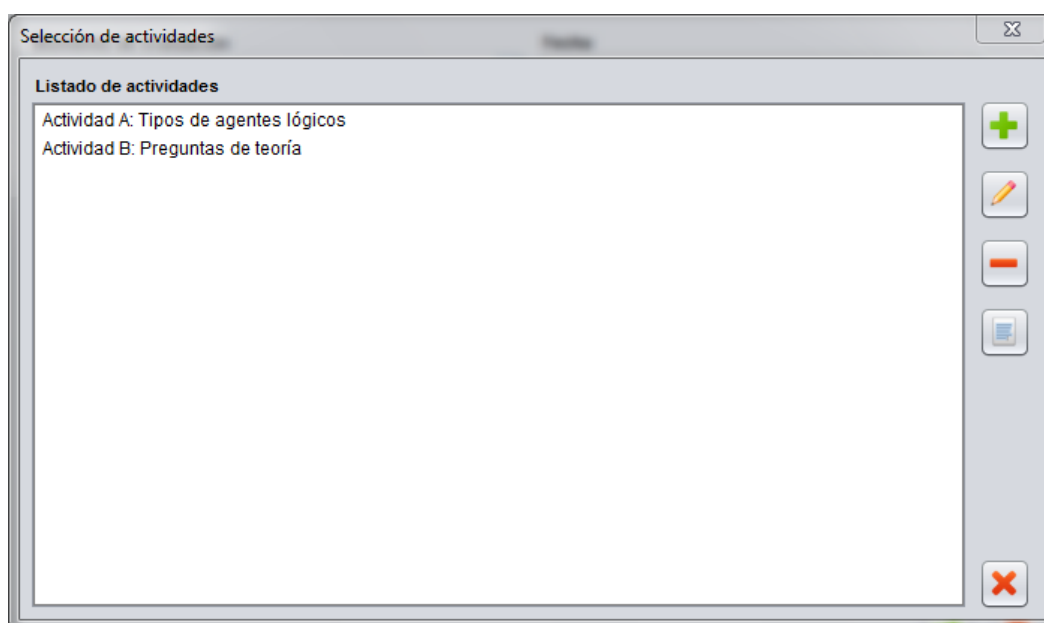


Figura 5.7 - Gestión de actividades

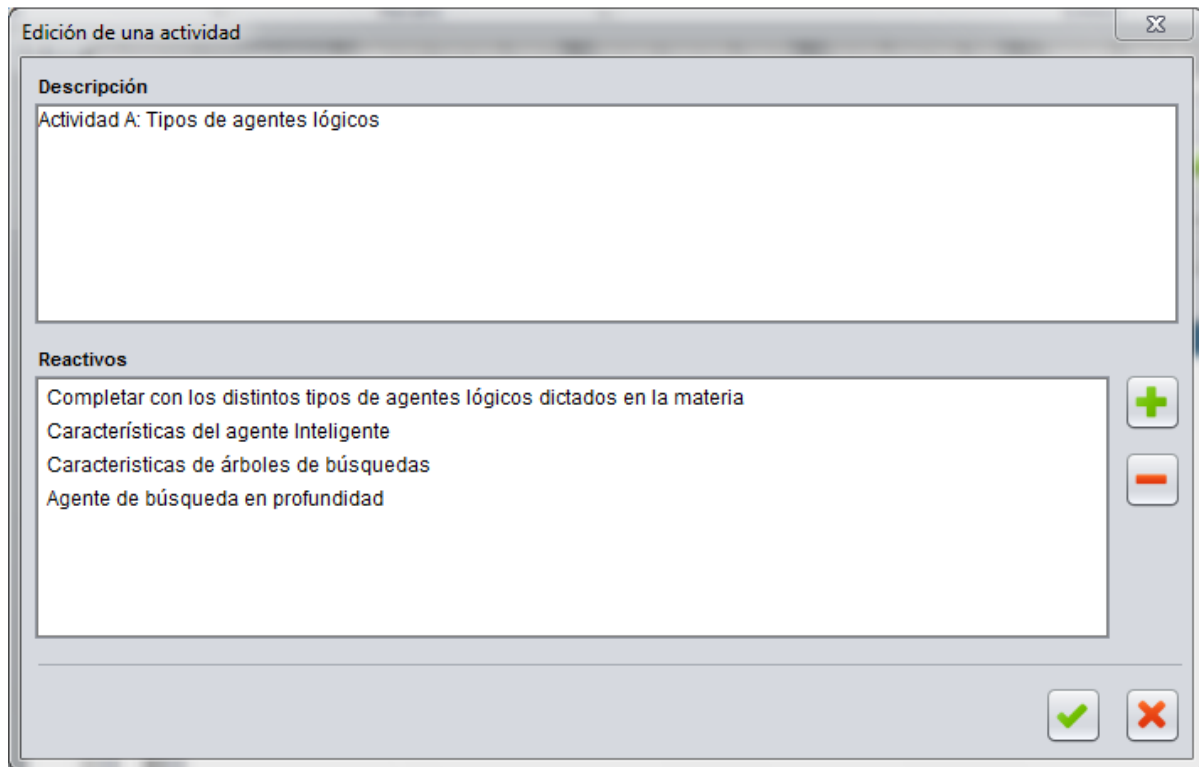
Dentro de la ventana de administración de actividades, se visualiza un botón para agregar elementos [ + ], que tiene por acción enviarnos a la ventana de edición de actividad para dar de alta una nueva. También se observa el botón de edición [ ✎ ], que permite editar los atributos de la actividad previamente seleccionada y cargada dentro de una ventana de edición de actividades. Si se quiere eliminar una actividad, el mecanismo es similar a la eliminación de evaluaciones, pulsando el botón de eliminación [ - ] y aceptando la confirmación de borrado.

El botón de selección de actividades [ 📄 ] se activa solo cuando se accede a la ventana de administración de actividades desde la ventana de edición de evaluaciones y permite tomar





las actividades que tenemos seleccionadas dentro de esta ventana y cargarlas en la evaluación que estamos editando. El listado de las actividades de la evaluación permite cargar un conjunto no definido de actividades, por lo que podemos seleccionar varias y cargarlas de una sola vez dentro de la evaluación.

La ventana de edición de actividades es la siguiente:



**Figura 5.8 - Edición de actividades**

En la interfaz para la creación de una actividad, se observa dos atributos a ser llenados por el usuario, uno es la descripción de la actividad y una lista de al menos 1 reactivo.

El botón para agregar reactivos [  ] de la figura 5.8, nos indica el acceso a la selección de reactivos, y el botón para quitar reactivos [  ] permite eliminar un reactivo previamente cargado. La actividad permite cargar un conjunto de reactivos dentro de la misma, por lo que podemos seleccionar varias dentro del listado de reactivo y cargarlos de una única vez dentro de la actividad.

En caso de que ocurra un error al momento de guardar la actividad se desplegará alguno de los siguientes mensajes:

Nivel de error	Falla	Mensaje
Alto	No se introdujo una descripción a la actividad	<i>“La descripción de la actividad no puede ser vacía”</i>
Alto	No se le cargó al menos 1 reactivo a la actividad	<i>“Debe al menos seleccionar un reactivo para la actividad”</i>

## Administración de reactivos

Para ingresar a la administración de reactivos se puede hacer mediante la barra de menú de la ventana principal:

*Material -> Administrar reactivos*

O mediante el botón de agregar elementos de la ventana de edición de evaluaciones, aunque de esta manera va a tener habilitado el botón de selección de elementos.

La ventana de administración de reactivos es la siguiente:

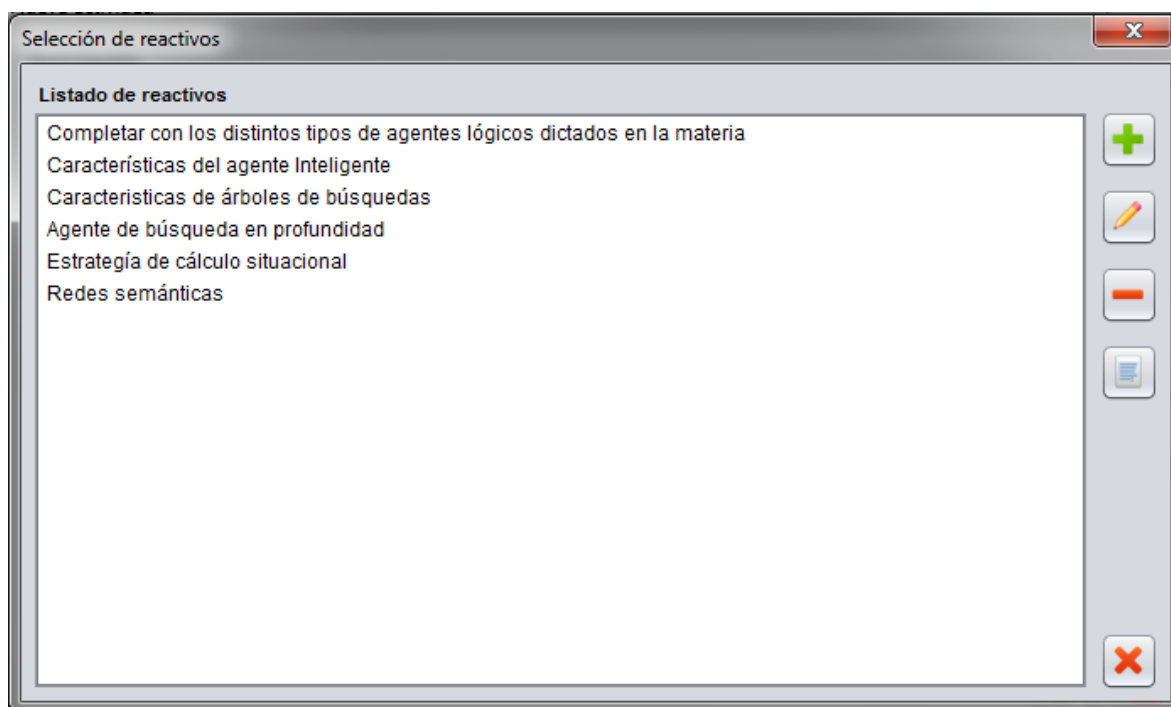


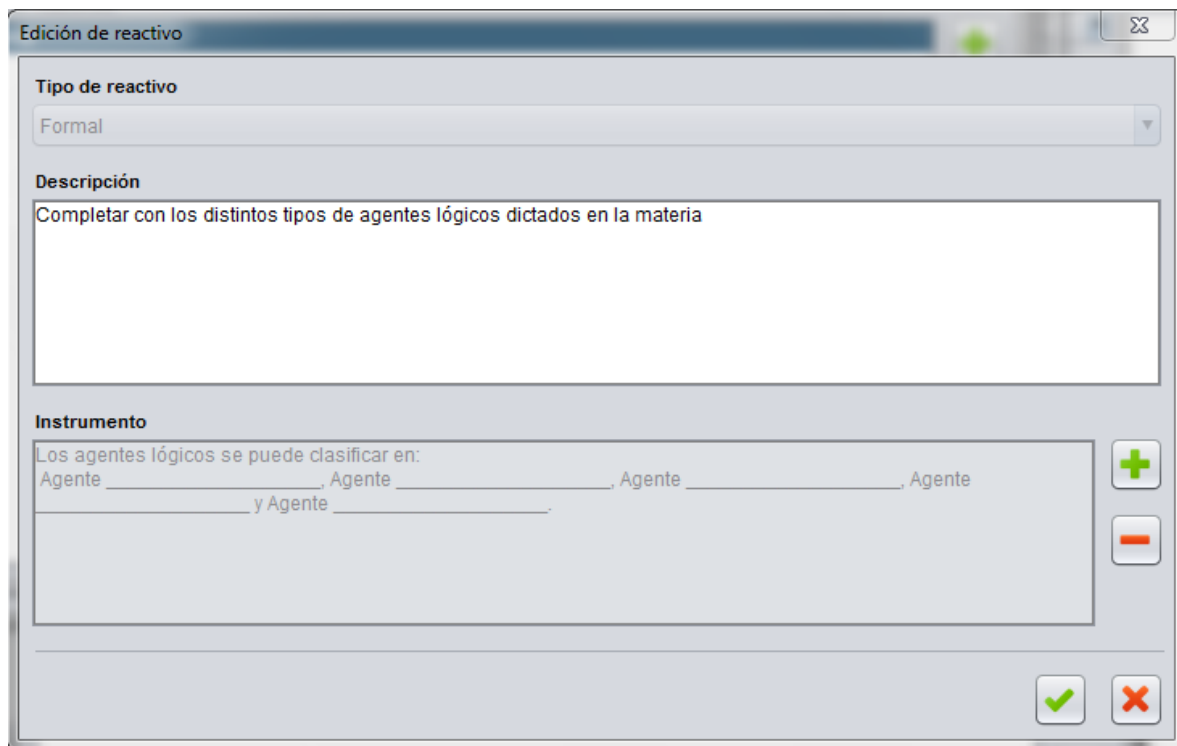
Figura 5.9 - Gestión de reactivos

Dentro de la ventana de administración de reactivos, se visualiza un botón para agregar elementos [ + ], que tiene por acción enviarnos a la ventana de edición de reactivos para dar de alta un nuevo reactivo. También se observa el botón de edición [ ✎ ], que permite editar los atributos del reactivo previamente seleccionado y cargado dentro de una ventana de edición. Si se quiere eliminar un reactivo, el mecanismo es similar a la eliminación de actividades, pulsando el botón de eliminación [ - ] y aceptando la confirmación de borrado.

El botón de selección de reactivos [ 📄 ] se activa solo cuando se accede a la ventana de administración de actividades desde la ventana de edición de actividades y permite tomar los

reactivos que tenemos seleccionados dentro de esta ventana y cargarlos en la actividad que estamos editando. La actividad permite cargar un conjunto de reactivos dentro de la misma, por lo que podemos seleccionar varias dentro del listado de reactivo y cargarlos de una única vez dentro de la actividad.



La ventana de edición de reactivos es la siguiente:



**Figura 5.10 - Edición de reactivo**

Esta ventana presenta un conjunto de atributos propios de los reactivos. Al momento de guardar se verifican que el reactivo tenga los campos obligatorios cargados y en forma, en caso contrario, se muestra alguno de los siguientes errores:

Nivel de error	Falla	Mensaje
Alto	No se introdujo una descripción al reactivo	<i>“La descripción del reactivo no puede ser vacía”</i>
Alto	No se le cargo un instrumento al reactivo	<i>“Debe al menos seleccionar un instrumento”</i>

El botón para agregar instrumento [  ] de la figura 5.10, nos indica el acceso a la selección de instrumentos, y el botón para quitar instrumentos [  ] permite eliminar un instrumento previamente cargado.

## Administración de instrumentos

Para ingresar a la administración de instrumentos se puede hacer mediante la barra de menú de la ventana principal:

*Material -> Administrar instrumentos*

O mediante el botón de agregar elementos de la ventana de edición de reactivos, aunque de esta manera va a tener habilitado el botón de selección de elementos.

La ventana de administración de instrumentos es la siguiente:

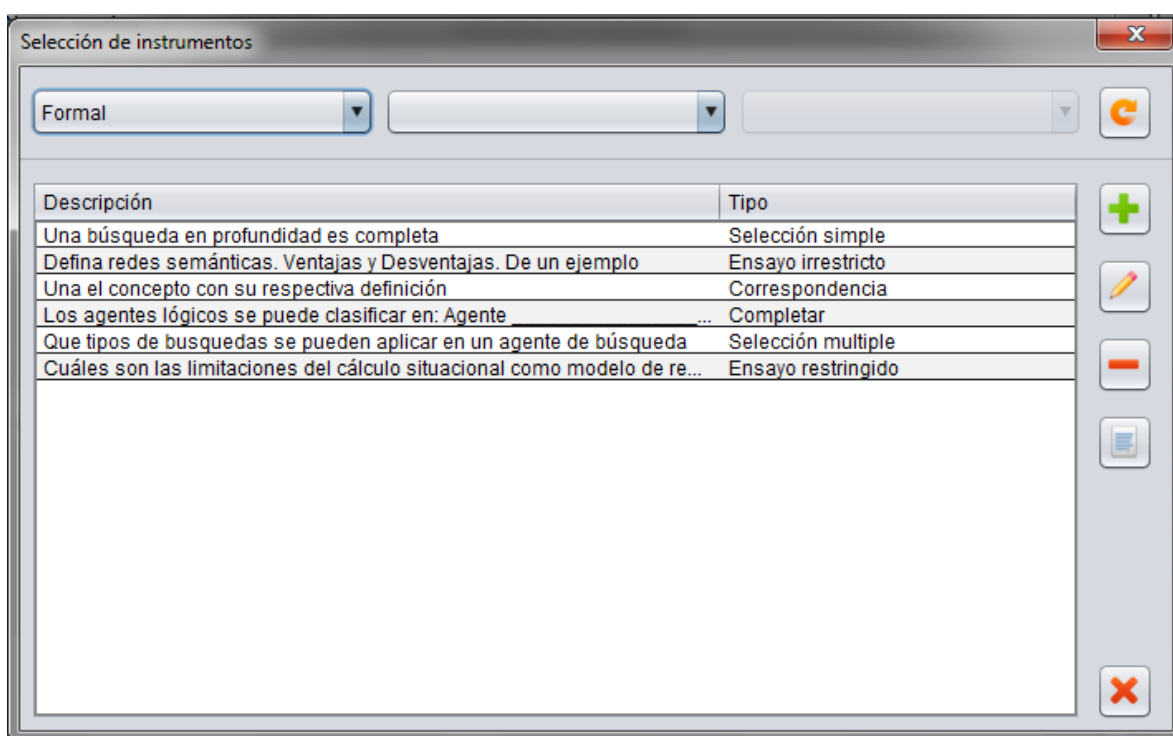







Figura 5.11 - Gestión de instrumentos

Se puede apreciar en la interfaz de instrumento en la parte superior, 3 campos de selección de tipo de instrumentos para realizar un filtrado. Al presionar el botón de actualizar [  ], la lista se carga con los instrumentos de base de datos, y ahí se puede empezar a interactuar con los elementos.

Dentro de la ventana de administración de instrumentos, se visualiza un botón para agregar elementos [  ], que tiene por acción enviarnos a la ventana de selección de tipo de instrumento para dar de alta un nuevo instrumento. También se observa el botón de edición [  ], que permite editar los atributos del instrumento previamente seleccionado y cargado

dentro de una ventana de edición. Si se quiere eliminar un instrumento, el mecanismo es similar a la eliminación de reactivos, pulsando el botón de eliminación [  ] y aceptando la confirmación de borrado.

El botón de selección de instrumento [  ] se activa solo cuando se accede a la ventana de administración de reactivos desde la ventana de edición de actividades y permite tomar los reactivos que tenemos seleccionados dentro de esta ventana y cargarlos en el reactivo que estamos editando. El reactivo solo puede contener un instrumento a la vez, por lo que al seleccionar otro este es reemplazado.

La ventana de selección de tipo de instrumento a crear es la siguiente:

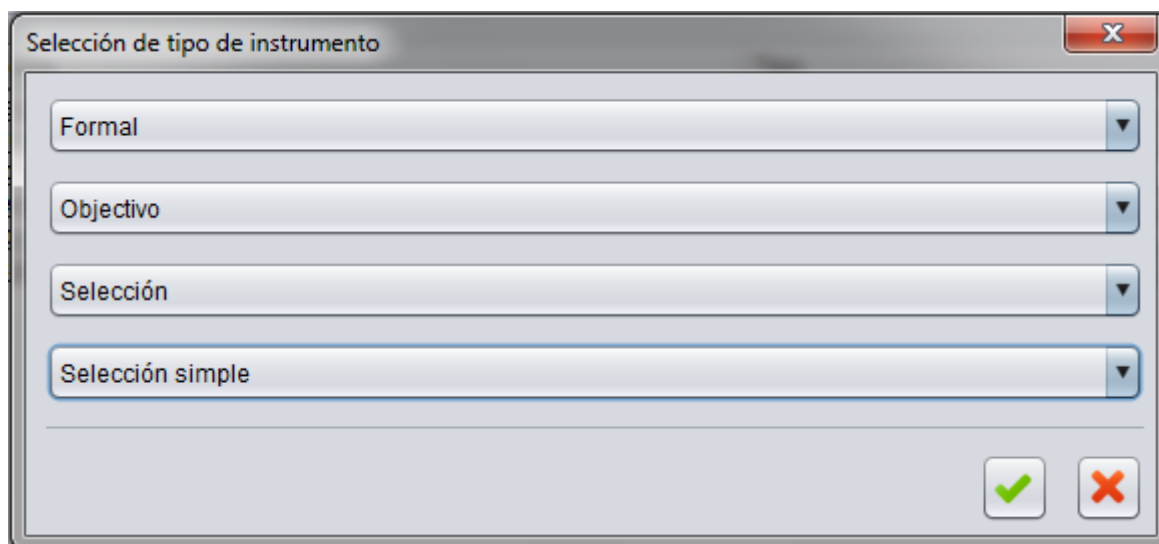


Figura 5.12 - Tipos de instrumentos

Luego de aceptar, dependiendo de lo seleccionado se presentan las siguientes interfaces donde corresponden a un instrumento en particular.

Para el caso de los instrumentos de selección (ya sean de selección simple o multiple) se va a acceder mediante las siguientes cadenas de acuerdo al tipo que queremos crear:

*Formal -> Objetivo -> Selección -> Selección Simple*

*Formal -> Objetivo -> Selección -> Selección Multiple*

La ventana de edición que se nos abre es la siguiente:

Figura 5.13 - Edición instrumentos de selección

El mecanismo para crear instrumento de selección simple o múltiple consiste en escribir la opción en el texto de área que está debajo de todo, tildar por verdadero o falso y apretar ENTER. También existe la posibilidad de tildar las opciones de “Todas las anteriores” o “Ninguna de las anteriores”. El resultado se ve reflejado en la parte de Opciones, donde se van a ir agregando las diferentes opciones que vayamos cargando. En caso que queramos editar una opción de ahí dentro, solo debemos seleccionarla y apretar ENTER o hacerle doble clic a la opción. Para borrar una opción que no queremos tener más en el instrumento, solo debemos seleccionarla y presionar el botón SUPRIMIR.

Al momento de guardar el instrumento, se verifica que los campos cumplan con ciertas reglas, en caso de que falle alguna, se muestra alguno de los siguientes mensajes:

Nivel de error	Falla	Mensaje
Alto	No se introdujo una descripción al instrumento	<i>“La descripción del instrumento no puede ser vacía”</i>



Alto	No se le cargó como mínimo una opción al instrumento	“Debe al menos contener una opción verdadera o una falsa”
------	--	---


Para el caso de los instrumentos de correspondencia se va a acceder mediante las siguientes cadenas de acuerdo al tipo que queremos crear:


*Formal -> Objetivo -> Correspondencia*

La ventana de edición que se nos abre es la siguiente:

Figura 5.14 - Edición de instrumentos de correspondencia

Para crear un instrumento de correspondencia se procede de la siguiente manera:

- Primero se cargan las opciones del lado izquierdo y lado derecho, escribiendo la palabra y presionando ENTER.
- Luego se selecciona un elemento de la parte izquierda y un elemento de la parte derecha y se presiona la tecla para crear una relación [  ]. Para eliminar dicha

relacion, se selecciona la misma y se presiona el botón de borrado [  ] y cada una de las partes de la relación vuelven al lado que corresponden.

- Si se desea editar una de las partes, ya sea derecha o izquierda, se debe seleccionar la misma y apretar ENTER (o haciendole doble clic), lo que saca la opción del listado y la carga en el campo de edición.
- Si se desea borrar una de las partes de las relaciones se debe seleccionar la misma y presionar el botón SUPRIMIR.

Al momento de guardar el instrumento, se verifica que los campos cumplan con ciertas reglas, en caso de que falle alguna, se muestra alguno de los siguientes mensajes:

Nivel de error	Falla	Mensaje
Alto	No se introdujo una descripción al instrumento	<i>“La descripción del instrumento no puede ser vacía”</i>
Alto	No se le cargó como mínimo una relación al instrumento	<i>“Debe al menos seleccionar una relación entre las frases izquierda y derecha”</i>

Para el caso de los instrumentos de completar se va a acceder mediante las siguientes cadenas de acuerdo al tipo que queremos crear:

*Formal -> Objetivo -> Completar*

La ventana de edición que se nos abre es la siguiente:

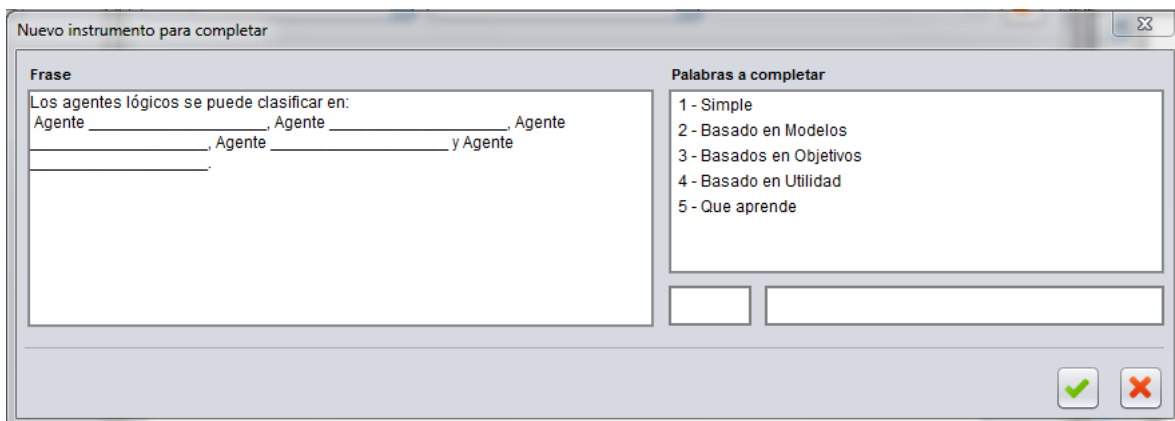


Figura 5.15 - Edición de instrumento de completar

Para la creación de un instrumento de completar, se escribe la frase y se deja un espacio en blanco o alguna simbología que representa la palabra a completar, como por ejemplo una línea de guiones bajos (“\_ \_ \_ \_”). Luego, en la parte derecha, abajo se encuentran dos áreas de texto, que por protocolo, en la primera área se pone un número que va a representar la ubicación de la palabra que corresponde a la ubicación a completar y en el cuadro de al lado se indica la palabra en sí misma. Al presionar el botón ENTER se carga la palabra a completar en el listado. En caso que se quiera editar una palabra a completar, se debe seleccionar la misma y presionar ENTER (o hacerle doble clic) y la misma se va a cargar en los campos de edición. Si se quiere eliminar la palabra, se debe seleccionar la misma desde el listado de palabras a completar y se presiona el botón SUPRIMIR.

Al momento de guardar el instrumento, se verifica que los campos cumplan con ciertas reglas, en caso de que falle alguna, se muestra alguno de los siguientes mensajes:

Nivel de error	Falla	Mensaje
Alto	No se introdujo una descripción al instrumento	<i>“La descripción del instrumento no puede ser vacía”</i>
Alto	No se le cargó como mínimo una palabra a completar al instrumento	<i>“Debe al menos tener una frase para completar”</i>
Alto	No se cargó un valor numérico al índice	<i>“El índice de la frase a completar debe ser un valor numérico”</i>

Para el caso de los instrumentos de ensayo objetivo (ya sean los ensayos restringidos o no restringidos) se va a acceder mediante las siguientes cadenas de acuerdo al tipo que queremos crear:

*Formal -> Ensayo -> Ensayo restringido*

*Formal -> Ensayo -> Ensayo irrestricto*

La ventana de edición que se nos abre es la siguiente:

**Nuevo instrumento de ensayo formal no restringido**

**Pregunta**  
 Defina redes semánticas. Ventajas y Desventajas. De un ejemplo

**Respuesta**  
 Una red semántica es una forma de lógica. Permite visualizar una BC así como algoritmos eficientes para inferir propiedades de un objeto con base en su pertenencia a una categoría. Representa gráficamente objetos individuales, categorías de objetos y relaciones entre objetos. Los objetos o nombres de categorías se representan en óvalos o cajas y los conecta con arcos etiquetados

**Figura 5.16 - Edición de instrumento de ensayo objetivo**

Los instrumentos de Ensayos, ya sea Restringido o Irrestringido, permiten poner la pregunta y su respectiva respuesta.

Al momento de guardar el instrumento, se verifica que los campos cumplan con ciertas reglas, en caso de que falle alguna, se muestra alguno de los siguientes mensajes:

Nivel de error	Falla	Mensaje
Alto	No se introdujo la pregunta al instrumento	<i>“La pregunta del instrumento no puede ser vacía”</i>
Alto	No se introdujo la respuesta al instrumento	<i>“La respuesta del instrumento no puede ser vacía”</i>

### Administración de conjuntos de reglas

Las siguientes interfaces se presentan como herramientas para crear y manipular conjuntos de reglas para validar evaluaciones.

La siguiente ventana nos permite administrar los conjuntos reglas en sí:

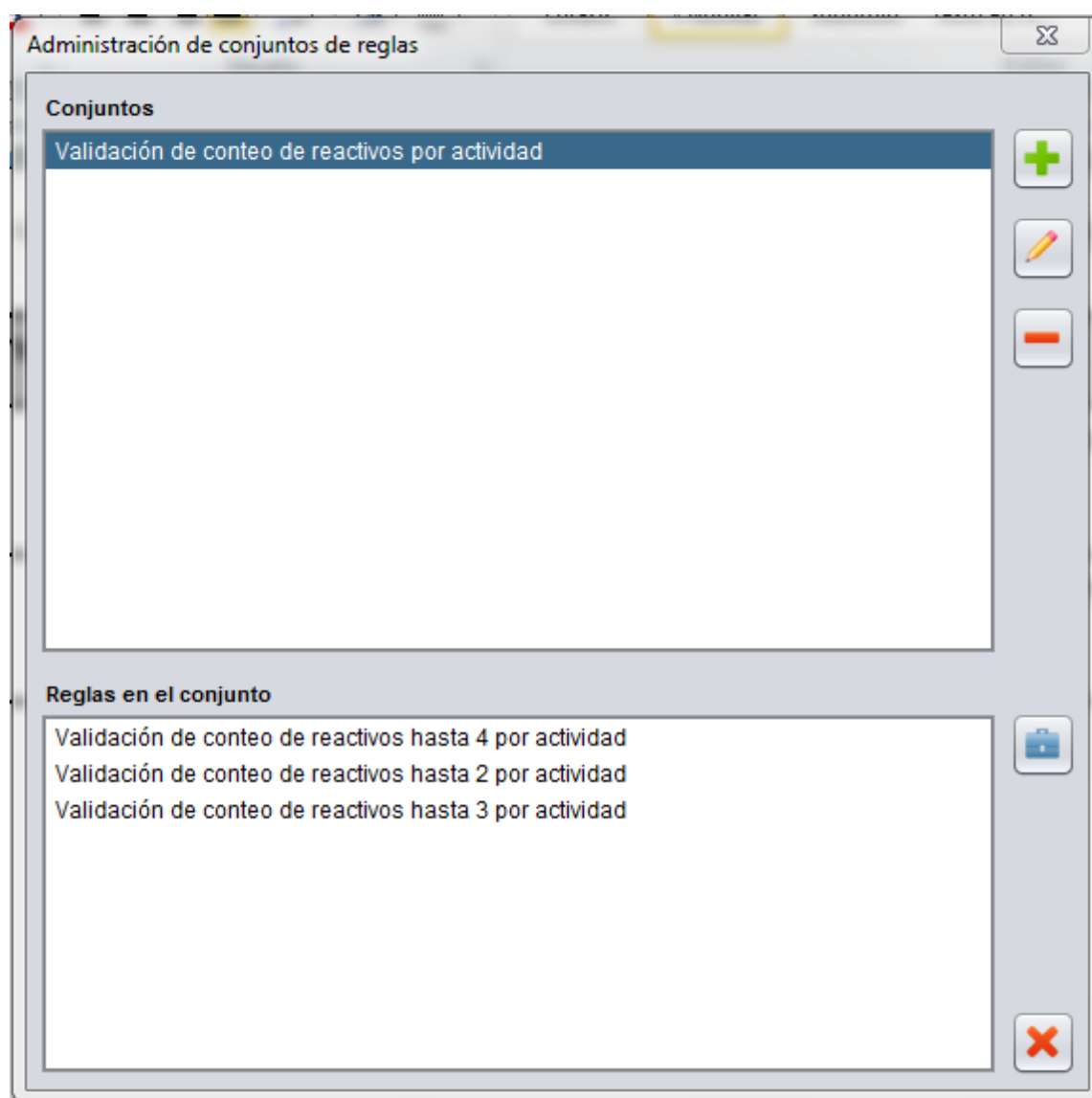





Figura 5.17 - Administración de conjuntos de reglas


Esta interfaz posibilita la generación de conjuntos de reglas acorde a la temática por la materia del agente que ingresó al sistema.

Otra alternativa para acceder a la ventana anterior es a través de la barra de menús.

*Reglas -> Administrar reglas*

Dentro de la ventana de administración de conjuntos de reglas, se visualiza un botón para agregar conjuntos [  ], que tiene por acción enviarnos a la ventana de edición de conjuntos de reglas. También se observa el botón de edición [  ], que permite editar un conjunto de regla previamente seleccionado y cargado dentro de una ventana de edición. Si se quiere eliminar un conjunto de regla, se debe seleccionar el conjunto que quiere eliminarse, pulsar el botón de eliminación [  ] y aceptar la confirmación de borrado.

**NOTA:** El hecho de borrar un conjunto de reglas no indica que también se borren las reglas que este tiene.

En esta misma ventana donde se despliega el listado de las reglas, se puede ver debajo una lista, donde van a desplegarse las reglas que tiene asignadas el conjunto que tenemos seleccionado. Al lado de este listado de reglas tenemos un botón de edición de reglas [  ] que nos permite abrir una ventana de administración de reglas.

La ventana de edición de conjunto es la que se muestra a continuación:

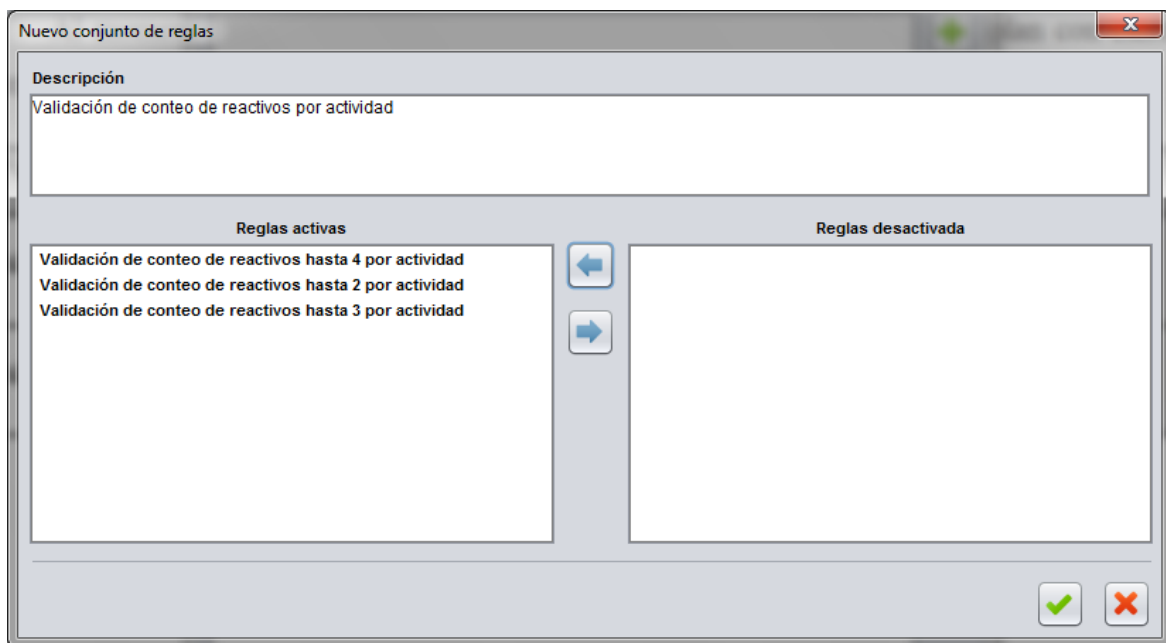




Figura 5.18 - Edición de conjuntos de reglas

Dentro de esta ventana podemos ver que se pueda cargar una descripción al conjunto que estamos editando. También vemos 2 listados de reglas, las mismas representan un listado de reglas activas dentro de este conjunto como también un conjunto de reglas no activas. En

medio de esos 2 listado tenemos 2 botones [   ] que tiene por funcionalidad activar o desactivar reglas para este conjunto moviendo las mismas de una lista a la otra.

Al momento de guardar el conjunto de reglas, se verifica que los campos cumplan con ciertos criterios, en caso de que falle alguno, se muestra alguno de los siguientes mensajes:

Nivel de error	Falla	Mensaje
Alto	No se introdujo la descripción del conjunto de regla	<i>“La descripción del conjunto de reglas no puede estar vacía”</i>
Alto	No se seleccionó ninguna regla para el conjunto	<i>“El conjunto de reglas debe al menos tener una regla activa”</i>
Alto	Ocurrió un fallo al momento de guardar la regla	<i>“Falla en la guardado del conjunto de reglas”</i>

## Administración de reglas

A continuación se detallara el funcionamiento de las interfaces que ayudan al usuario a crear y manipular reglas para validar evaluaciones.

Ventana de administracion de reglas en si:

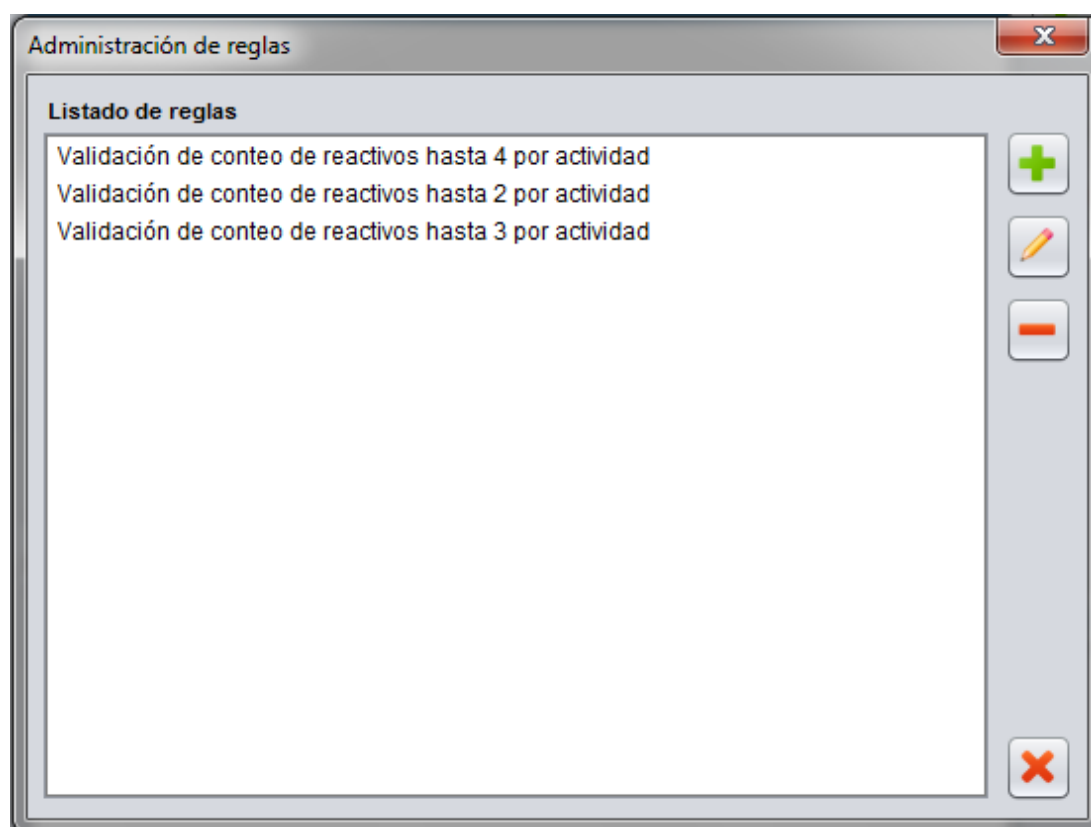


Figura 5.19 - Administración de reglas

Dentro de la ventana de administración de reglas, se visualiza un botón para agregar elementos [ + ], que tiene por acción enviarnos a la ventana de edición de reglas. También se observa el botón de edición [ ✎ ], que facilita la edición de una regla previamente seleccionada y cargada dentro de una ventana de edición. Si se quiere eliminar una regla, se debe seleccionar la que quiere eliminarse, pulsar el botón de eliminación [ - ] y aceptar la confirmación de borrado.

Para editar las reglas, tenemos a continuación la ventana:



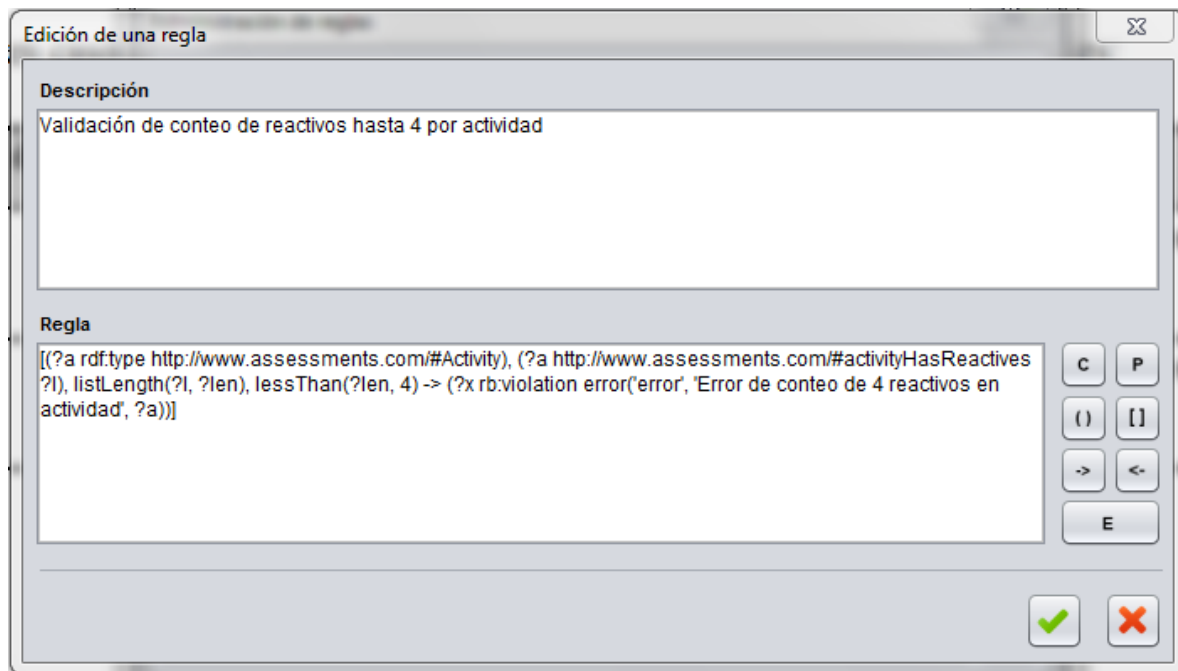



Figura 5.20 - Edición de reglas

En esta ventana vemos que disponemos de un área de texto donde introducimos una pequeña descripción que detalla lo que la regla va a validar, como así también un campo donde vamos a poder escribir la regla en sí misma.

Las reglas que podemos manejar dentro del sistema son una derivación de reglas SWRL que implementa el FRAMEWORK JENA (**Documentación JENA**).

Al momento de aceptar los cambios en la regla, el sistema valida la misma y su sintaxis y en caso de que ocurra un error al guardar se mostrara alguno de los siguientes mensajes:

Nivel de error	Falla	Mensaje
Alto	No se introdujo la descripción de la regla	<i>“La descripción de la regla no puede estar vacía”</i>
Alto	La regla cargada no cumple con la sintaxis de las reglas de JENA	<i>“La regla no está correctamente formada”</i>
Alto	Ocurrió un fallo al momento de guardar la regla	<i>“Falla en el guardado de la regla”</i>

Para poder escribir las reglas de manera más fácil y cometer menos errores, se dispone de un pequeño panel donde podemos abrir una ventana donde tenemos el listado de las clases que tenemos dentro de la ontología (botón [  ]), como también una ventana donde tenemos

desplegado un listado de las propiedades de las entidades de la ontología (botón [ **P** ]). También disponemos de un par de botones para colocar símbolos propios de la sintaxis de las reglas, como son el botón de paréntesis (botón [ ( ) ]), el botón de corchetes (botón [ [ ] ]) y los conectores de reglas forward (botón [ → ]) y backward (botón [ ← ]). El último botón del panel de ayuda nos carga la sentencia que nos permite marcar como error la ontología para poder validar la misma (botón [ **E** ]).

Toda la ayuda que ocupemos se va a cargar dentro del campo de reglas en la posición donde tenemos colocada el cursor del mouse.

Las ventanas de listado que nos ayudan a la creación de reglas son:

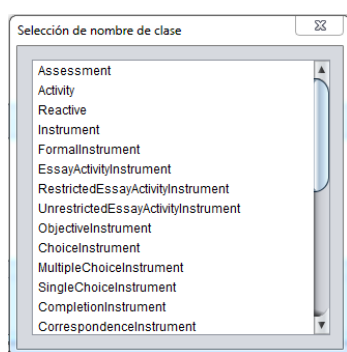


Figura 5.21 - Nombres de clases

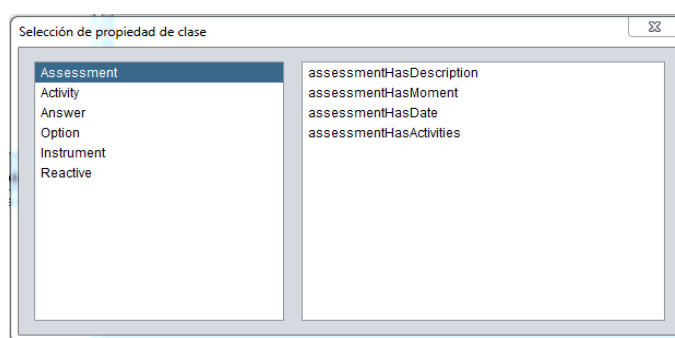


Figura 5.22 - Nombre de propiedades

Para poder seleccionar el nombre de una clase o de una propiedad solo debemos hacer doble clic sobre la misma, el nombre de la misma junto al NAMESPACE de la ontología, se va a cargar dentro del campo regla.

## Capítulo 6 - Conclusiones y Perspectivas futuras

### 6.1 Puntos de vistas y logros obtenidos gracias a la investigación e implementación de la tesis

Habiendo finalizado este trabajo, detallamos a continuación las principales contribuciones del presente Proyecto Final de Carrera, tanto como proyecto de investigación como al desarrollo de un sistema de información:

- **Herramienta OFGA para el CIDISI:** Se ha desarrollado la herramienta de software de uso libre, que será utilizada en el CIDISI para dar soporte a la generación de evaluaciones junto con sus componentes como actividades, reactivos, instrumentos y reglas de validación. Este proyecto nos permitió incorporar nuevos conocimientos y afianzar los adquiridos a lo largo de la carrera, en cuanto a la necesidad de contar con especificaciones de requerimientos que permitan el correcto desarrollo de un sistema desde su etapa inicial, como así también al uso y los aportes que las ontologías están brindando a su definición.
- **Uso de herramientas y técnicas para el manejo de ontologías:** Se trabajó sobre la carga, manipulación, gestión en memoria, gestión en base de datos, y almacenamiento de ontologías con extensión OWL para permitir la generación de las evaluaciones.
- **Implementación de la herramienta:** Mediante la implementación de OFGA se deriva la creación de evaluaciones por parte de usuarios del mismo que tengan la necesidad de conocer a través de la implementación de reglas sobre las ontologías que son usadas como base. También permite chequear la consistencia de la ERS en todo momento, evitando así que se produzcan errores por cambios que puedan haber sido efectuados por los stakeholders.

### 6.2 Perspectivas futuras de implementaciones sobre el sistema realizado

Podemos destacar que el sistema al ser un prototipo con funcionalidad a medida, permite el análisis de futuros desarrollos que surgen con un estudio más amplio de requerimientos y funcionalidades genéricas como son:

- Desarrollo de todas las jerarquías de componentes de una evaluación en lo que respecta a lo no informal. Actividades, reactivos, instrumentos que son de uso general y no protocolar.
- Analizador semántico de reglas del framework JENA. Estas reglas se componen de varias formas, lo cual lleva a generar diferentes alternativas de composición, donde un análisis de la construcción de las mismas sería muy eficiente y de ayuda para los futuros usuarios.

### 6.3 Entorno donde va a ser utilizado el software y usuarios potenciales

El sistema como primera instancia va a ser utilizado por profesores familiarizados con la herramienta, que darán el uso adecuado de acuerdo a sus necesidades. Al ser un prototipo de sistema se carece de funcionalidades que vayan más allá de los límites que la aplicación soporte, siempre manteniéndose en el ámbito de ontologías de evaluaciones, sin embargo no deja de ser una herramienta fácil de utilizar, lo que permite que personas con conocimientos mínimos en ontologías, puedan definir reglas en JENA para ser utilizadas sobre las evaluaciones creadas.

## Referencias

(Allocca, y col., 2009) Allocca, C., D'aquin, M., and Motta, E. "DOOR-towards a formalization of ontology relations". Proc. Int. Conference on Knowledge Engineering and Ontology Development, 2009, pp 13-20.

(Ashton, 2008). Ashton H Beevers C Thomas R. "Can e-Assessment become Mainstream?" Proc. 12th Internacional Computer Assisted Assessment Conference. 8th & 9th July 2008 at Loughborough University (United Kingdom). pp 13-24. 2008.

(Bloom, 1956) Bloom, B., Krathwohl, D., Taxonomy of educational objectives. "The classification of educational goals by a committee of college and university examiners". Handbook 1. Cognitive domain. New York: Addison-Wesley.

(Bolivar, 2011) C. Bolivar. "Pruebas de rendimiento académico" Technical report. Programa interinstitucional doctorado en educación. 2011.

(Castellanos-Nieves, 2011) Castellanos-Nieves, D., Fernandez-Breis, J., Valencia-Garcia, R., Martinez-Bejar, R. Semantic Web Technologies for supporting learning assessment. Journal Information Science an International Journal archive. Volume 181, Issue 9, May 2011. Elsevier.

(Chan y Chen, 2009) Chan, T. & Chen, Y. (2009). Cooperative learning in E-learning: A peer assessment of student-centered using consistent fuzzy preference. Expert system with application. Elsevier Vol 36(4), pp. 8342–8349.

(Chung y colab., 2003) Chung, G., Niami, D., BewLey, W. Assessment Applications of Ontologies. 23 p. Annual Meeting of the American Educational Research Association. 2003.

(Cubric y colab., 2010) Cubric, M., M & Tasic, M 2010. Towards automatic generation of e-assessment using semantic web technologies. In: Proceedings of the 2010 International Computer Assisted Assessment Conference, Jul, vol 2010.

(Davis, 1993) Davis, R., Sorbe, H., Szolovits, P. (1993). What is a Knowledge Representation? AI Magazine. Spring, 17-33.

**(Díaz y col., 2012)** Díaz, A., Motz, R., Rohrer, E., and Tansini, L. “An Ontology Network for Educational Recommender Systems”. Educational recommender systems and technologies. Practices and challenges. IGI Global, 2012, Chapter 4, pp. 67-93.

**(Frutos y Morales, 2010)** Frutos-Morales, F., Sanchez-Vera, M., Castellanos-Nieves, D., Esteban-Gil, A., Cruz-Corona, C., Prendes-Espinosa, M. & Fernandez-Breis, J. (2010). An extension of the OeLE platform for generating semantic feedback for students and teachers. Procedia. Social and behavioral sciences. Elsevier Ltd. Volume 2, (2), Pp. 527–531.

**(Gama, 2005)** Gama, J., Gomes de Olivera, Z., da Silva, E. La evaluación del aprendizaje en los programas de educación a la distancia: Construcción de procesos y alcance de calidad. Universidade do Estado de Rio de Janeiro. Etic@net. ISSN: 1695-324X. Año II Número 4.

**(Gladun, 2009)** Gladun, A. Rogushina, J., García-Sánchez, F. Martínez-Béjar, R. & Fernández-Breis, J. (2009). An application of intelligent techniques and semantic web technologies in e-learning environments. Expert Systems with Applications, Elsevier, 36, pp, 1922–1931

**(Gómez-Pérez, 2004)** Gómez-Pérez, A., Fernández-López, M., Corcho, O. (2004). Ontological Engineering with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. Springer. 2004.

**(Guardia Ortiz, 2005)** Guàrdia Ortiz, L., Sangrà Morer, A. Red Diseño instruccional y objetos de aprendizaje; hacia un modelo para el diseño de actividades de evaluación del aprendizaje online. Revista de educación a distancia, julio, año/vol. IV, número monográfico OIV. Universidad de Murcia. (2005).

**(Lindberg, 1993)** Lindberg, D.A.B., Humphrey, B.L., McCray, A.T. (1993). The Unified Medical Language System. Methods of Information in Medicine. 32, 281-291.

**(Maedche, 2002)** Maedche, A. Ontology Learning for the Semantic Web. Kluwer Academic Publishers (2002).

**(Pierce 1935)** The Collected Papers of Charles Sanders Peirce. Electronic edition reproducing Vols. I–VI [C. Hartshorne & P. Weiss (eds.), Cambridge: Harvard University

Press, 1931–1935]; Vols. VII—VIII [A. W. Burks (ed.), same publisher, 1958]. Charlottesville: Intelix Corporation, 1931–1935. (Quoted as CP).

**(Romero y col, 2012)** Romero, L., Gutiérrez, M. y Caliusco, M. 2012. Towards an ontology network for conceptualizing e-assessment domain. Conferencia Ibérica de Sistemas y Tecnologías de Información. Madrid, España.

**(Suárez-Figueroa, 2010)** Suárez-Figueroa, M. C. "NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse ". Thesis Doctoral, Facultad de Informática - Universidad Politécnica de Madrid. 2010.

**(Van Heijst, 1997)** Van Heijst, G., Schreiber, A. T., Wielinga, B. J. (1997). Using explicit ontologies in KBS development, International Journal of Human-Computer Studies, 45: 183-292.

**(Web semántica, 2007)**

<http://www.maestrosdelweb.com/editorial/web-semantica-y-sus-principales-caracteristicas/>

**(Documentación Kunagi)**

<http://jesuslc.com/2013/01/12/documentacion-de-kunagi-en-espanol/>

**(Documentación Jena)**

<http://jena.apache.org/documentation/inference/index.html>

**(Documentación Eclipse)**

[http://es.wikipedia.org/wiki/Eclipse\\_\(software\)](http://es.wikipedia.org/wiki/Eclipse_(software))

**(Documentación Maven)**

<http://es.wikipedia.org/wiki/Maven>

**(Documentación Hibernate)**

<http://es.wikipedia.org/wiki/Hibernate>

**(Documentación Spring)**

[http://es.wikipedia.org/wiki/Spring\\_Framework](http://es.wikipedia.org/wiki/Spring_Framework)

**(Documentación Tortoise SVN)**

<http://softlibrewin.org.ar/Instalaciones/TortoiseSVN/TortoiseSVN-1.4.0-es.pdf>

**(Documentación MySQL)**

<http://www.esepestudio.com/noticias/que-es-mysql>



## Anexos

### Escenarios de Casos de usos

#### El módulo de login

<b>Módulo:</b> Login		<b>Iteración:</b> 1	
<b>Nombre del Caso de Uso:</b> Login al sistema.		<b>Versión:</b> 1.0	<b>Nro.:</b> 001
<b>Actor Principal:</b> Agente del Sistema		<b>Actor Secundario:</b> No Aplica	
<b>Criticidad:</b> <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja			
<b>Prioridad:</b> <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Útil <input type="checkbox"/> Deseable			
<b>Complejidad:</b> <input type="checkbox"/> Extrem. Complejo <input type="checkbox"/> Muy Complejo <input checked="" type="checkbox"/> Complejo <input type="checkbox"/> Mediano <input type="checkbox"/> Simple			
<b>Frecuencia:</b> <input checked="" type="checkbox"/> Siempre <input type="checkbox"/> Frecuentemente <input type="checkbox"/> A veces <input type="checkbox"/> Raramente <input type="checkbox"/> Una vez			
<b>Tipo de Caso de Uso:</b> <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto			
<b>Descripción:</b> El usuario ingresa dentro del sistema OFGA para hacer uso del mismo.			
<b>Precondiciones:</b> <ul style="list-style-type: none"> <li>▪ El usuario debe estar registrado en base de datos.</li> </ul>			
<b>Postcondiciones:</b>	Éxito: <ul style="list-style-type: none"> <li>▪ Ingreso Correcto a la aplicación.</li> </ul>		
	Fracaso: <ul style="list-style-type: none"> <li>▪ Mensaje de error al intentar ingresar a la aplicación.</li> </ul>		
<b>Curso Normal</b>		<b>Cursos Alternativos</b>	
1. El sistema muestra la pantalla inicial Login, donde se presentan los campos “Agente” [Texto Alfanumérico] y “contraseña” [Texto Alfanumérico con máscara “*”]. Ambos son requeridos.			
2. El actor ingresa el nombre de agente y su respectiva contraseña y presiona el botón “Aceptar cambios”.			
3. El sistema valida la existencia del usuario y la coincidencia con su contraseña.		3.a - Los datos de ingreso están incompletos o no son correctos. 3.a.1 - El usuario no ha ingresado el valor del campo “Agente” o el mismo es incorrecto. 3.a.2 - El sistema informa mediante el mensaje <b>CU001-MSG01</b> . 3.a.3 - El usuario no ha ingresado el valor de la “Contraseña” o la misma no es correcta.	

	3.a.4 - El sistema informa mediante el mensaje <b>CU001- MSG02.</b> 3.a.5 - El caso de uso continúa en el punto 2.
4. El sistema despliega la ventana de selección de materia en la que está registrado el agente.	
5. El agente selecciona la materia.	
6. El sistema despliega la ventana de menú principal.	
7. El caso de uso finaliza.	
<b>Requerimientos No funcionales Especiales:</b> No aplica.	
<b>Asociaciones de Extensión:</b> No aplica.	
<b>Asociaciones de Inclusión:</b> No aplica.	
<b>Caso de Uso de Generalización:</b> No aplica.	
<b>Referencias:</b>  <u><b>Mensajes empleados</b></u>  <b>CU001- MSG01</b> - <i>El nombre del agente es incorrecto.</i>  <b>CU001- MSG02</b> - <i>La contraseña es incorrecta.</i>	
<b>Observaciones:</b> No aplica.	

## El módulo de instrumentos

<b>Módulo:</b> ABM Instruments		<b>Iteración:</b> 1
<b>Nombre del Caso de Uso:</b> Crear Instrumento	<b>Versión:</b> 1.0	<b>Nro.:</b> 002
<b>Actor Principal:</b> Agente del sistema.		<b>Actor Secundario:</b> No Aplica.
<b>Criticidad:</b> <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja		
<b>Prioridad:</b> <input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Útil <input type="checkbox"/> Deseable		
<b>Complejidad:</b> <input type="checkbox"/> Extrem. Complejo <input type="checkbox"/> Muy Complejo <input checked="" type="checkbox"/> Complejo <input type="checkbox"/> Mediano <input type="checkbox"/> Simple		
<b>Frecuencia:</b> <input type="checkbox"/> Siempre <input type="checkbox"/> Frecuentemente <input checked="" type="checkbox"/> A veces <input type="checkbox"/> Raramente <input type="checkbox"/> Una vez		
<b>Tipo de Caso de Uso:</b> <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
<b>Descripción:</b> El caso de uso que detalla los pasos que realiza el usuario para crear un nuevo instrumento dentro del sistema.		
<b>Precondiciones:</b> <ul style="list-style-type: none"> <li>El agente debe haber ingresado al sistema.</li> </ul>		
<b>Postcondiciones:</b>	<b>Éxito:</b> <ul style="list-style-type: none"> <li>Creación correcta del instrumento.</li> </ul>	
	<b>Fracaso:</b> <ul style="list-style-type: none"> <li>Mensaje de error al intentar crear el instrumento.</li> </ul>	
<b>Curso Normal</b>		<b>Cursos Alternativos</b>
1. El agente presiona el botón "Agregar Elemento" para crear nuevo instrumento.		
2. El agente selecciona el tipo de instrumento.		
3. El agente carga los datos correspondientes al tipo de instrumento que quiere utilizar y presiona el botón "Aceptar Cambios".		
4. El sistema verifica y valida que estén todos los datos necesarios para crear el instrumento.		4.a - El sistema informa al agente por la información faltantes dependiendo del tipo de instrumento. 4.b - El caso de uso continúa en el punto 3.
5. El sistema guardar el instrumento en la base de datos.		
6. El caso de uso finaliza.		
<b>Requerimientos No funcionales Especiales:</b> No aplica.		
<b>Asociaciones de Extensión:</b> No aplica.		
<b>Asociaciones de Inclusión:</b> No aplica.		
<b>Caso de Uso de Generalización:</b> No aplica.		

<b>Referencias:</b> No aplica.
<b>Observaciones:</b> No aplica.

<b>Módulo:</b> ABM Instruments		<b>Iteración:</b> 1
<b>Nombre del Caso de Uso:</b> Buscar instrumentos.	<b>Versión:</b> 1.0	<b>Nro.:</b> 003
<b>Actor Principal:</b> Agente del sistema.	<b>Actor Secundario:</b> No Aplica.	
<b>Criticidad:</b> <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja		
<b>Prioridad:</b> <input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Útil <input type="checkbox"/> Deseable		
<b>Complejidad:</b> <input type="checkbox"/> Extrem. Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Complejo <input checked="" type="checkbox"/> Mediano <input type="checkbox"/> Simple		
<b>Frecuencia:</b> <input type="checkbox"/> Siempre <input type="checkbox"/> Frecuentemente <input checked="" type="checkbox"/> A veces <input type="checkbox"/> Raramente <input type="checkbox"/> Una vez		
<b>Tipo de Caso de Uso:</b> <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
<b>Descripción:</b> Caso de uso que realiza la búsqueda de los instrumentos que tenemos dentro del sistema.		
<b>Precondiciones:</b> <ul style="list-style-type: none"> <li>▪ Debe haber un agente logueado.</li> </ul>		
<b>Postcondiciones:</b>	<b>Éxito:</b> <ul style="list-style-type: none"> <li>▪ Búsqueda exitosa de los instrumentos.</li> </ul>	
	<b>Fracaso:</b> <ul style="list-style-type: none"> <li>▪ Búsqueda no exitosa de los instrumentos.</li> </ul>	
<b>Curso Normal</b>		<b>Cursos Alternativos</b>
1. El agente selecciona el tipo de instrumento y presiona el botón “Cargar elementos”.		
2. El sistema lista todos los instrumentos encontrados.		2.a - El instrumento buscado no se encuentra 2.a.1 - Se invoca al <b>CU002</b> . 2.a.2 - Se vuelve al paso 2.
3. El caso de uso finaliza.		
<b>Requerimientos No funcionales Especiales:</b> No aplica.		
<b>Asociaciones de Extensión:</b> <ul style="list-style-type: none"> <li>▪ CU002 - Crear instrumento.</li> </ul>		
<b>Asociaciones de Inclusión:</b> No aplica.		
<b>Caso de Uso de Generalización:</b> No aplica.		

**Referencias:** No aplica.

**Observaciones:** No aplica.

<b>Módulo:</b> ABM Instruments		<b>Iteración:</b> 1
<b>Nombre del Caso de Uso:</b> Modificar instrumento.	<b>Versión:</b> 1.0	<b>Nro.:</b> 004
<b>Actor Principal:</b> Agente del sistema.	<b>Actor Secundario:</b> No Aplica	
<b>Criticidad:</b> <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja		
<b>Prioridad:</b> <input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Útil <input type="checkbox"/> Deseable		
<b>Complejidad:</b> <input type="checkbox"/> Extrem. Complejo <input type="checkbox"/> Muy Complejo <input checked="" type="checkbox"/> Mediano <input type="checkbox"/> Simple		
<b>Frecuencia:</b> <input type="checkbox"/> Siempre <input type="checkbox"/> Frecuentemente <input checked="" type="checkbox"/> A veces <input type="checkbox"/> Raramente <input type="checkbox"/> Una vez		
<b>Tipo de Caso de Uso:</b> <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
<b>Descripción:</b> Caso de uso que detalla cómo se realiza la modificación de un instrumento dentro del sistema.		
<b>Precondiciones:</b> <ul style="list-style-type: none"> <li>Debe haber un agente logueado.</li> </ul>		
<b>Postcondiciones:</b>	<b>Éxito:</b> <ul style="list-style-type: none"> <li>Modificación exitosa del instrumento.</li> </ul>	
	<b>Fracaso:</b> <ul style="list-style-type: none"> <li>Mensaje de error al intentar modificar el instrumento.</li> </ul>	
<b>Curso Normal</b>		<b>Cursos Alternativos</b>
1. El sistema despliega en pantalla todos los instrumentos almacenados mediante la invocación al <b>CU003</b> .		
2. El agente selecciona el instrumento a modificar y presiona el botón "Editar Elemento".		
3. El sistema abre la ventana de edición con el instrumento ya cargado.		
4. El agente realiza los cambios de los datos.		
5. El sistema verifica la consistencia de la información modificada.		5.a - El sistema informa al agente por la información faltantes dependiendo del tipo de instrumento. 5.b - El caso de uso continúa en el punto 4.
6. El sistema guarda los cambios del instrumento en la base de datos.		
7. El caso de uso finaliza.		

<b>Requerimientos No funcionales Especiales:</b> No aplica.
<b>Asociaciones de Extensión:</b> No aplica.
<b>Asociaciones de Inclusión:</b> <ul style="list-style-type: none"> <li>▪ CU003 - Buscar instrumentos.</li> </ul>
<b>Caso de Uso de Generalización:</b> No aplica.
<b>Referencias:</b> No aplica.
<b>Observaciones:</b> No aplica.

<b>Módulo:</b> ABM Instruments		<b>Iteración:</b> 1
<b>Nombre del Caso de Uso:</b> Eliminar instrumento.		<b>Versión:</b> 1.0
<b>Nro.:</b> 005		
<b>Actor Principal:</b> Agente del sistema.	<b>Actor Secundario:</b> No Aplica.	
<b>Criticidad:</b> <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja		
<b>Prioridad:</b> <input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Útil <input type="checkbox"/> Deseable		
<b>Complejidad:</b> <input type="checkbox"/> Extrem. Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Complejo <input checked="" type="checkbox"/> Mediano <input type="checkbox"/> Simple		
<b>Frecuencia:</b> <input type="checkbox"/> Siempre <input type="checkbox"/> Frecuentemente <input checked="" type="checkbox"/> A veces <input type="checkbox"/> Raramente <input type="checkbox"/> Una vez		
<b>Tipo de Caso de Uso:</b> <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
<b>Descripción:</b> Caso de uso que describe como eliminar un instrumento del sistema.		
<b>Precondiciones:</b> <ul style="list-style-type: none"> <li>▪ Debe haber un agente logueado.</li> </ul>		
<b>Postcondiciones:</b>	<b>Éxito:</b> <ul style="list-style-type: none"> <li>▪ Borrado exitoso del instrumento.</li> </ul>	
	<b>Fracaso:</b> <ul style="list-style-type: none"> <li>▪ Borrado no exitoso del instrumento.</li> </ul>	
<b>Curso Normal</b>		<b>Cursos Alternativos</b>
1. El sistema despliega en pantalla todos los instrumentos almacenados mediante la invocación del <b>CU003</b> .		
2. El agente selecciona el instrumento a eliminar y presiona el botón "Eliminar elemento".		

3. El sistema solicita una confirmación de eliminación del instrumento.	
4. El agente acepta la confirmación.	4.a – El usuario cancela la eliminación. 4.b - El caso de uso finaliza.
5. El sistema procede a eliminar el instrumento de la base de datos.	
6. El caso de uso finaliza.	
<b>Requerimientos No funcionales Especiales:</b> No aplica.	
<b>Asociaciones de Extensión:</b> No aplica.	
<b>Asociaciones de Inclusión:</b> <ul style="list-style-type: none"> <li>▪ CU003 - Buscar instrumentos.</li> </ul>	
<b>Caso de Uso de Generalización:</b> No aplica.	
<b>Referencias:</b> No aplica.	
<b>Observaciones:</b> No aplica.	

### El módulo de reactivos

<b>Módulo:</b> ABM Reactives		<b>Iteración:</b> 1
<b>Nombre del Caso de Uso:</b> Crear reactivo.	<b>Versión:</b> 1.0	<b>Nro.:</b> 006
<b>Actor Principal:</b> Agente del sistema.	<b>Actor Secundario:</b> No Aplica	
<b>Criticidad:</b> <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja		
<b>Prioridad:</b> <input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Útil <input type="checkbox"/> Deseable		
<b>Complejidad:</b> <input type="checkbox"/> Extrem. Complejo <input type="checkbox"/> Muy Complejo <input checked="" type="checkbox"/> Complejo <input type="checkbox"/> Mediano <input type="checkbox"/> Simple		
<b>Frecuencia:</b> <input type="checkbox"/> Siempre <input type="checkbox"/> Frecuentemente <input checked="" type="checkbox"/> A veces <input type="checkbox"/> Raramente <input type="checkbox"/> Una vez		
<b>Tipo de Caso de Uso:</b> <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
<b>Descripción:</b> Caso de uso que detalla los pasos que realiza el usuario para crear un nuevo reactivo dentro del sistema.		
<b>Precondiciones:</b> ■ Debe haber un agente logueado.		
<b>Postcondiciones:</b>	Éxito: ■ Creación correcta del reactivo.	
	Fracaso: ■ Mensaje de error al intentar crear el reactivo.	
<b>Curso Normal</b>		<b>Cursos Alternativos</b>
1. El agente presiona el botón “Agregar Elemento” para crear nuevo reactivo.		
2. El agente carga la descripción correspondiente para la creación del reactivo.		
3. El sistema busca todos los instrumentos almacenados mediante la invocación del <b>CU003</b> .		
4. El agente selecciona el instrumento.		
5. El sistema verifica y valida que estén todos los datos necesarios para crear el reactivo.		5.a - La descripción se encuentra vacía. 5.a.1 – El agente no introdujo información al campo “Descripción”. 5.a.2 – El sistema informa con el mensaje <b>CU006-MSG01</b> . 5.b – No sé cargo el instrumento al reactivo. 5.b.1 – El agente no selecciono un instrumento para el reactivo. 5.b.2 - El sistema informa con el mensaje <b>CU006-MSG02</b> . 5.c - El caso de uso continúa en el punto 2.
6. El sistema guardar el reactivo en la base de		



datos.	
7. El caso de uso finaliza.	
<b>Requerimientos No funcionales Especiales:</b> No aplica.	
<b>Asociaciones de Extensión:</b> No aplica.	
<b>Asociaciones de Inclusión:</b>	
<ul style="list-style-type: none"> <li>CU003 - Buscar instrumentos.</li> </ul>	
<b>Caso de Uso de Generalización:</b> No aplica.	
<b>Referencias:</b>	
<u><b>Mensajes empleados</b></u>	
CU006-MSG01 - La descripción del reactivo no puede ser vacía.	
CU006-MSG02 - Debe al menos seleccionar un instrumento.	
<b>Observaciones:</b> No aplica.	

<b>Módulo:</b> ABM Reactivos		<b>Iteración:</b> 1
<b>Nombre del Caso de Uso:</b> Buscar reactivos.	<b>Versión:</b> 1.0	<b>Nro.:</b> 007
<b>Actor Principal:</b> Sistema.	<b>Actor Secundario:</b> No Aplica.	
<b>Criticidad:</b> <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja		
<b>Prioridad:</b> <input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Útil <input type="checkbox"/> Deseable		
<b>Complejidad:</b> <input type="checkbox"/> Extrem. Complejo <input type="checkbox"/> Muy Complejo <input checked="" type="checkbox"/> Mediano <input type="checkbox"/> Simple <input type="checkbox"/> Complejo		
<b>Frecuencia:</b> <input type="checkbox"/> Siempre <input type="checkbox"/> Frecuentemente <input checked="" type="checkbox"/> A veces <input type="checkbox"/> Raramente <input type="checkbox"/> Una vez		
<b>Tipo de Caso de Uso:</b> <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
<b>Descripción:</b> Caso de uso que detalla la búsqueda de los reactivos que tenemos almacenados dentro del sistema.		
<b>Precondiciones:</b> <ul style="list-style-type: none"> <li>Debe haber un agente logueado.</li> </ul>		
<b>Postcondiciones:</b>	Éxito:	
	<ul style="list-style-type: none"> <li>Búsqueda exitosa de los reactivos.</li> </ul>	
<b>Postcondiciones:</b>	Fracaso:	
	<ul style="list-style-type: none"> <li>Búsqueda no exitosa de los reactivos.</li> </ul>	

Curso Normal	Cursos Alternativos
1. El sistema busca todos los reactivos almacenados en el sistema de acuerdo al agente logueado.	
2. El sistema lista todos los reactivos encontrados.	2.a - El reactivo buscado no se encuentra listado. 2.a.1 - Se invoca al <b>CU006</b> . 2.a.2 - Se vuelve al paso 2.
3. El caso de uso finaliza.	
<b>Requerimientos No funcionales Especiales:</b> No aplica.	
<b>Asociaciones de Extensión:</b>  ▪ CU006 - Crear reactivo.	
<b>Asociaciones de Inclusión:</b> No aplica.	
<b>Caso de Uso de Generalización:</b> No aplica.	
<b>Referencias:</b> No aplica.	
<b>Observaciones:</b> No aplica.	

<b>Módulo:</b> ABM Reactives		<b>Iteración:</b> 1
<b>Nombre del Caso de Uso:</b> Modificar reactivo.	<b>Versión:</b> 1.0	<b>Nro.:</b> 008
<b>Actor Principal:</b> Agente del sistema.	<b>Actor Secundario:</b> No Aplica.	
<b>Criticidad:</b> <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja		
<b>Prioridad:</b> <input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Útil <input type="checkbox"/> Deseable		
<b>Complejidad:</b> <input type="checkbox"/> Extrem. Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Complejo <input checked="" type="checkbox"/> Mediano <input type="checkbox"/> Simple		
<b>Frecuencia:</b> <input type="checkbox"/> Siempre <input type="checkbox"/> Frecuentemente <input checked="" type="checkbox"/> A veces <input type="checkbox"/> Raramente <input type="checkbox"/> Una vez		
<b>Tipo de Caso de Uso:</b> <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
<b>Descripción:</b> Caso de uso que detalla cómo se realiza la modificación de un reactivo dentro del sistema.		
<b>Precondiciones:</b>  ▪ Debe haber un agente logueado.		
<b>Postcondiciones:</b>	Éxito: ▪ Modificación exitosa del reactivo.	

	<b>Fracaso:</b> <ul style="list-style-type: none"> <li>▪ Mensaje de error al intentar modificar el reactivo.</li> </ul>
<b>Curso Normal</b>	<b>Cursos Alternativos</b>
1. El sistema despliega en pantalla todos los reactivos almacenados mediante la invocación del <b>CU007</b> .	
2. El agente selecciona el reactivo a modificar y presiona el botón “Editar Elemento”.	
3. El sistema abre la ventana de edición con el reactivo ya cargado.	
4. El agente realiza los cambios de los datos.	4.a - Se requiere modificar el instrumento. 4.a.1 - Se invoca al <b>CU003</b> . 4.a.2 - Se vuelve al paso 4.
5. El sistema verifica la consistencia de la información modificada.	5.a - La descripción se encuentra vacía. 5.a.1 - El agente no introdujo información al campo “Descripción”. 5.a.2 - El sistema informa con el mensaje <b>CU008-MSG01</b> . 5.b - No sé cargo el instrumento al reactivo. 5.b.1 - El agente no selecciono un instrumento al reactivo. 5.b.2 - El sistema informa con el mensaje <b>CU008-MSG02</b> . 5.c - El caso de uso continúa en el punto 4.
6. El sistema guarda los cambios del reactivo en la base de datos.	
7. El caso de uso finaliza.	
<b>Requerimientos No funcionales Especiales:</b> No aplica.	
<b>Asociaciones de Extensión:</b> <ul style="list-style-type: none"> <li>▪ CU003 - Buscar instrumentos.</li> </ul>	
<b>Asociaciones de Inclusión:</b> <ul style="list-style-type: none"> <li>▪ CU007 - Buscar reactivos.</li> </ul>	
<b>Caso de Uso de Generalización:</b> No aplica.	
<b>Referencias:</b>  <u><b>Mensajes empleados</b></u>  <b>CU008-MSG01</b> - <i>La descripción del reactivo no puede ser vacía.</i>  <b>CU008-MSG02</b> - <i>Debe al menos seleccionar un instrumento.</i>	
<b>Observaciones:</b> No aplica.	

<b>Módulo:</b> ABM Reactives		<b>Iteración:</b> 1
<b>Nombre del Caso de Uso:</b> Eliminar reactivo.	<b>Versión:</b> 1.0	<b>Nro.:</b> 009
<b>Actor Principal:</b> Agente del sistema.	<b>Actor Secundario:</b> No Aplica	
<b>Criticidad:</b> <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja		
<b>Prioridad:</b> <input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Útil <input type="checkbox"/> Deseable		
<b>Complejidad:</b> <input type="checkbox"/> Extrem. Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Complejo <input checked="" type="checkbox"/> Mediano <input type="checkbox"/> Simple		
<b>Frecuencia:</b> <input type="checkbox"/> Siempre <input type="checkbox"/> Frecuentemente <input checked="" type="checkbox"/> A veces <input type="checkbox"/> Raramente <input type="checkbox"/> Una vez		
<b>Tipo de Caso de Uso:</b> <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
<b>Descripción:</b> Caso de uso que describe como eliminar un reactivo del sistema.		
<b>Precondiciones:</b> <ul style="list-style-type: none"> <li>▪ Debe haber un agente logueado.</li> </ul>		
<b>Postcondiciones:</b>	<b>Éxito:</b> <ul style="list-style-type: none"> <li>▪ Borrado exitoso del reactivo.</li> </ul>	
	<b>Fracaso:</b> <ul style="list-style-type: none"> <li>▪ Borrado no exitoso del reactivo.</li> </ul>	
<b>Curso Normal</b>		<b>Cursos Alternativos</b>
1. El sistema despliega en pantalla todos los reactivos almacenados mediante la invocación al <b>CU007</b> .		
2. El agente selecciona el reactivo a eliminar y presiona el botón "Eliminar elemento".		
3. El sistema solicita una confirmación de eliminación del reactivo.		
4. El agente acepta la confirmación.		4.a - El usuario cancela la eliminación. 4.a.1 - El caso de uso finaliza.
5. El sistema procede a eliminar el reactivo de la base de datos.		
6. El caso de uso finaliza.		
<b>Requerimientos No funcionales Especiales:</b> No aplica.		
<b>Asociaciones de Extensión:</b> No aplica.		
<b>Asociaciones de Inclusión:</b> <ul style="list-style-type: none"> <li>▪ CU007 - Buscar reactivo.</li> </ul>		
<b>Caso de Uso de Generalización:</b> No aplica.		

<b>Referencias:</b> No aplica.
<b>Observaciones:</b> No aplica.

### El módulo de actividades

<b>Módulo:</b> ABM Activities		<b>Iteración:</b> 1
<b>Nombre del Caso de Uso:</b> Crear actividad.	<b>Versión:</b> 1.0	<b>Nro.:</b> 010
<b>Actor Principal:</b> Agente del sistema.	<b>Actor Secundario:</b> No Aplica.	
<b>Criticidad:</b> <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja		
<b>Prioridad:</b> <input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Útil <input type="checkbox"/> Deseable		
<b>Complejidad:</b> <input type="checkbox"/> Extrem. Complejo <input type="checkbox"/> Muy Complejo <input checked="" type="checkbox"/> Complejo <input type="checkbox"/> Mediano <input type="checkbox"/> Simple		
<b>Frecuencia:</b> <input type="checkbox"/> Siempre <input type="checkbox"/> Frecuentemente <input checked="" type="checkbox"/> A veces <input type="checkbox"/> Raramente <input type="checkbox"/> Una vez		
<b>Tipo de Caso de Uso:</b> <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
<b>Descripción:</b> Caso de uso que detalla los pasos para crear una nueva actividad dentro del sistema.		
<b>Precondiciones:</b> <ul style="list-style-type: none"> <li>▪ Debe haber un agente logueado.</li> </ul>		
<b>Postcondiciones:</b>	<b>Éxito:</b> <ul style="list-style-type: none"> <li>▪ Creación correcta de la actividad.</li> </ul>	
	<b>Fracaso:</b> <ul style="list-style-type: none"> <li>▪ Mensaje de error al intentar crear la actividad.</li> </ul>	
<b>Curso Normal</b>		<b>Cursos Alternativos</b>
1. El agente presiona el botón “Agregar Elemento” para crear una nueva actividad.		
2. El agente carga la descripción de la actividad.		
3. El sistema busca todo los reactivos almacenados mediante la invocación al <b>CU007</b> .		
4. El agente selecciona los reactivos a cargar en la actividad a través del botón “Asignar elemento” y luego presiona el botón “Aceptar Cambios”.		
5. El sistema verifica y valida que estén todos los datos necesarios para crear la actividad.		5.a - La descripción se encuentra vacía. 5.a.1 - El agente no introdujo información al campo “Descripción”. 5.a.2 - El sistema informa con el mensaje <b>CU010-MSG01</b> . 5.b - No sé cargo por lo menos un reactivo para la actividad. 5.b.1 - El agente no selecciono ningún reactivo. 5.b.2 - El sistema informa con el mensaje <b>CU010-MSG02</b> .

	5.c - El caso de uso continúa en el punto 2.
6. El sistema guardar la actividad en la base de datos.	
7. El caso de uso finaliza.	
<b>Requerimientos No funcionales Especiales:</b> No aplica.	
<b>Asociaciones de Extensión:</b> No aplica.	
<b>Asociaciones de Inclusión:</b>	
<ul style="list-style-type: none"> <li>CU007 - Buscar reactivos.</li> </ul>	
<b>Caso de Uso de Generalización:</b> No aplica.	
<b>Referencias:</b>	
<u><b>Mensajes empleados</b></u>	
CU010-MSG01 - La descripción de la actividad no puede ser vacía.	
CU010-MSG02 - Debe al menos seleccionar un reactivo para la actividad.	
<b>Observaciones:</b> No aplica.	

<b>Módulo:</b> ABM Activities		<b>Iteración:</b> 1
<b>Nombre del Caso de Uso:</b> Buscar actividades	<b>Versión:</b> 1.0	<b>Nro.:</b> 011
<b>Actor Principal:</b> Sistema.	<b>Actor Secundario:</b> No Aplica.	
<b>Criticidad:</b>	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	
<b>Prioridad:</b>	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Útil <input type="checkbox"/> Deseable	
<b>Complejidad:</b>	<input type="checkbox"/> Extrem. Complejo <input type="checkbox"/> Muy Complejo <input checked="" type="checkbox"/> Mediano <input type="checkbox"/> Simple <input type="checkbox"/> Complejo	
<b>Frecuencia:</b>	<input type="checkbox"/> Siempre <input type="checkbox"/> Frecuentemente <input checked="" type="checkbox"/> A veces <input type="checkbox"/> Raramente <input type="checkbox"/> Una vez	
<b>Tipo de Caso de Uso:</b>	<input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto	
<b>Descripción:</b>		
Caso de uso que detalla la búsqueda de las actividades que tenemos almacenados dentro del sistema.		
<b>Precondiciones:</b>		
<ul style="list-style-type: none"> <li>Debe haber un agente logueado.</li> </ul>		

<b>Postcondiciones:</b>	Éxito:	<ul style="list-style-type: none"> <li>Búsqueda exitosa de las actividades.</li> </ul>
	Fracaso:	<ul style="list-style-type: none"> <li>Búsqueda no exitosa de las actividades.</li> </ul>
<b>Curso Normal</b>		<b>Cursos Alternativos</b>
1. El sistema busca todos las actividades almacenados en el sistema de acuerdo al agente logueado.		
2. El sistema lista todas las actividades encontradas.		2.a - La actividad requerida no se encuentra listada. 2.a.1 - Se invoca al <b>CU010</b> . 2.a.2 - Se vuelve al paso 2.
3. El caso de uso finaliza.		
<b>Requerimientos No funcionales Especiales:</b> No aplica.		
<b>Asociaciones de Extensión:</b>		
<ul style="list-style-type: none"> <li>CU010 - Crear actividad.</li> </ul>		
<b>Asociaciones de Inclusión:</b> No aplica.		
<b>Caso de Uso de Generalización:</b> No aplica.		
<b>Referencias:</b> No aplica.		
<b>Observaciones:</b> No aplica.		

<b>Módulo:</b> ABM Activities		<b>Iteración:</b> 1
<b>Nombre del Caso de Uso:</b> Modificar actividad.	<b>Versión:</b> 1.0	<b>Nro.:</b> 012
<b>Actor Principal:</b> Agente del sistema.	<b>Actor Secundario:</b> No Aplica.	
<b>Criticidad:</b>	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	
<b>Prioridad:</b>	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Útil <input type="checkbox"/> Deseable	
<b>Complejidad:</b>	<input type="checkbox"/> Extrem. Complejo <input type="checkbox"/> Muy Complejo <input checked="" type="checkbox"/> Mediano <input type="checkbox"/> Simple <input type="checkbox"/> Complejo	
<b>Frecuencia:</b>	<input type="checkbox"/> Siempre <input type="checkbox"/> Frecuentemente <input checked="" type="checkbox"/> A veces <input type="checkbox"/> Raramente <input type="checkbox"/> Una vez	
<b>Tipo de Caso de Uso:</b>	<input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto	
<b>Descripción:</b>		
Caso de uso que detalla cómo se realiza la modificación de una actividad dentro del sistema.		
<b>Precondiciones:</b>		



<ul style="list-style-type: none"> <li>▪ Debe haber un agente logueado.</li> </ul>	
<b>Postcondiciones:</b>	Éxito: <ul style="list-style-type: none"> <li>▪ Modificación exitosa de la actividad.</li> </ul>
	Fracaso: <ul style="list-style-type: none"> <li>▪ Mensaje de error al intentar modificar la actividad.</li> </ul>
<b>Curso Normal</b>	<b>Cursos Alternativos</b>
1. El sistema despliega en pantalla todas las actividades almacenadas mediante la invocación al <b>CU011</b> .	
2. El agente selecciona la actividad a modificar y presiona el botón “Editar Elemento”.	
3. El sistema abre la ventana de edición con la actividad ya cargada.	
4. El agente realiza los cambios de los datos y presiona el botón “Aceptar Cambios”.	4.a - Se requiere editar los reactivos de la actividad. 4.a.1 - Se invoca al <b>CU007</b> . 4.a.2 - Se vuelve al paso 4.
5. El sistema verifica la consistencia de la información modificada.	5.a - La descripción se encuentra vacía. 5.a.1 - El agente no introdujo información al campo “Descripción”. 5.a.2 - El sistema informa con el mensaje <b>CU012-MSG01</b> . 5.b - No sé cargo por lo menos un reactivo. 5.b.1 - El agente no selecciono ningún reactivo. 5.b.2 - El sistema informa con el mensaje <b>CU012-MSG02</b> . 5.c - El caso de uso continúa en el punto 4.
6. El sistema guarda los cambios de la actividad en la base de datos.	
7. El caso de uso finaliza.	
<b>Requerimientos No funcionales Especiales:</b> No aplica.	
<b>Asociaciones de Extensión:</b> <ul style="list-style-type: none"> <li>▪ CU007 - Buscar reactivos.</li> </ul>	
<b>Asociaciones de Inclusión:</b> <ul style="list-style-type: none"> <li>▪ CU011 - Buscar actividades.</li> </ul>	
<b>Caso de Uso de Generalización:</b> No aplica.	
<b>Referencias:</b>  <b><u>Mensajes empleados</u></b>  <b>CU012-MSG01</b> - <i>La descripción de la actividad no puede ser vacía.</i>  <b>CU012-MSG02</b> - <i>Debe al menos seleccionar un reactivo para la actividad.</i>	
<b>Observaciones:</b> No aplica.	

<b>Módulo:</b> ABM Activities		<b>Iteración:</b> 1	
<b>Nombre del Caso de Uso:</b> Eliminar actividad.		<b>Versión:</b> 1.0	<b>Nro.:</b> 013
<b>Actor Principal:</b> Agente del sistema.		<b>Actor Secundario:</b> No Aplica.	
<b>Criticidad:</b> <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja			
<b>Prioridad:</b> <input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Útil <input type="checkbox"/> Deseable			
<b>Complejidad:</b> <input type="checkbox"/> Extrem. Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Complejo <input checked="" type="checkbox"/> Mediano <input type="checkbox"/> Simple			
<b>Frecuencia:</b> <input type="checkbox"/> Siempre <input type="checkbox"/> Frecuentemente <input checked="" type="checkbox"/> A veces <input type="checkbox"/> Raramente <input type="checkbox"/> Una vez			
<b>Tipo de Caso de Uso:</b> <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto			
<b>Descripción:</b> Caso de uso que describe como eliminar una actividad del sistema.			
<b>Precondiciones:</b> <ul style="list-style-type: none"> <li>▪ Debe haber un agente logueado.</li> </ul>			
<b>Postcondiciones:</b>	Éxito: <ul style="list-style-type: none"> <li>▪ Borrado exitoso de la actividad.</li> </ul>		
	Fracaso: <ul style="list-style-type: none"> <li>▪ Borrado no exitoso de la actividad.</li> </ul>		
<b>Curso Normal</b>		<b>Cursos Alternativos</b>	
1. El sistema despliega en pantalla todas las actividades almacenadas mediante la invocación al <b>CU011</b> .			
2. El agente selecciona la actividad a eliminar y presiona el botón "Eliminar elemento".			
3. El sistema solicita una confirmación de eliminación de la actividad.			
4. El agente acepta la confirmación.		4.a - El usuario cancela la eliminación. 4.a.1 - El caso de uso finaliza.	
5. El sistema procede a eliminar la actividad de la base de datos.			
6. El caso de uso finaliza.			
<b>Requerimientos No funcionales Especiales:</b> No aplica.			
<b>Asociaciones de Extensión:</b> No aplica.			

<b>Asociaciones de Inclusión:</b>
<ul style="list-style-type: none"><li>▪ CU011 - Buscar actividades.</li></ul>
<b>Caso de Uso de Generalización:</b> No aplica.
<b>Referencias:</b> No aplica.
<b>Observaciones:</b> No aplica.

## El módulo de evaluaciones

<b>Módulo:</b> ABM Assessments		<b>Iteración:</b> 1
<b>Nombre del Caso de Uso:</b> Crear evaluación.	<b>Versión:</b> 1.0	<b>Nro.:</b> 014
<b>Actor Principal:</b> Agente del sistema.	<b>Actor Secundario:</b> No Aplica.	
<b>Criticidad:</b> <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
<b>Prioridad:</b> <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Útil <input type="checkbox"/> Deseable		
<b>Complejidad:</b> <input type="checkbox"/> Extrem. Complejo <input type="checkbox"/> Muy Complejo <input checked="" type="checkbox"/> Complejo <input type="checkbox"/> Mediano <input type="checkbox"/> Simple		
<b>Frecuencia:</b> <input type="checkbox"/> Siempre <input checked="" type="checkbox"/> Frecuentemente <input type="checkbox"/> A veces <input type="checkbox"/> Raramente <input type="checkbox"/> Una vez		
<b>Tipo de Caso de Uso:</b> <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
<b>Descripción:</b> Caso de uso que detalla los pasos para crear una nueva evaluación dentro del sistema.		
<b>Precondiciones:</b> ■ Debe haber un agente logueado.		
<b>Postcondiciones:</b>	Éxito: ■ Creación correcta de la evaluación.	
	Fracaso: ■ Mensaje de error al intentar crear la evaluación.	
<b>Curso Normal</b>		<b>Cursos Alternativos</b>
1. El agente presiona el botón “Agregar Elemento” para crear una nueva evaluación.		
2. El sistema despliega los tipos de evaluaciones posibles.		
3. El agente selecciona el tipo de evaluación.		
4. El agente selecciona el momento de la evaluación, la fecha e ingresa la descripción.		
5. El sistema despliega todas las actividades almacenadas en base de datos mediante la invocación al <b>CU011</b> .		
6. El agente selecciona las actividades a cargar a través del botón “Asignar elemento” y luego presiona el botón “Aceptar Cambios”.		
7. El sistema verifica y valida que estén todos los datos necesarios para crear la evaluación.		7.a - No se seleccionó ningún momento de evaluación. 7.a.1 - El agente no selecciono el momento de la evaluación. 7.a.2 - El sistema informa con el mensaje <b>CU014-MSG01</b> . 7.b - No selecciono o introdujo una fecha para la evaluación.

	7.b.1 - El agente no selecciono ni introdujo una fecha. 7.b.2 - El sistema informa con el mensaje <b>CU014-MSG02</b> . 7.c - No se ingresó ninguna descripción. 7.c.1 - El agente dejo el campo "Descripción" vacío. 7.c.2 - El sistema informa con el mensaje <b>CU014-MSG03</b> . 7.d - No se cargó una actividad. 7.d.1 - El agente no selecciono por lo menos una actividad. 7.d.2 - El sistema informa con el mensaje <b>CU014-MSG04</b> . 7.e - El caso de uso continúa en el punto 4.
8. El sistema guardar la evaluación en la base de datos.	
9. El caso de uso finaliza.	
<b>Requerimientos No funcionales Especiales:</b> No aplica.	
<b>Asociaciones de Extensión:</b> No aplica.	
<b>Asociaciones de Inclusión:</b> <ul style="list-style-type: none"> <li>▪ CU011 - Buscar actividades.</li> </ul>	
<b>Caso de Uso de Generalización:</b> No aplica.	
<b>Referencias:</b> <u><b>Mensajes empleados</b></u> <b>CU014-MSG01</b> - <i>Debe seleccionar un tipo de evaluación.</i> <b>CU014-MSG02</b> - <i>Debe colocar una fecha de evaluación.</i> <b>CU014-MSG03</b> - <i>La descripción de la evaluación no puede ser vacía.</i> <b>CU014-MSG04</b> - <i>Debe al menos seleccionar una actividad para la evaluación.</i>	
<b>Observaciones:</b> No aplica.	

<b>Módulo:</b> ABM Assessments		<b>Iteración:</b> 1
<b>Nombre del Caso de Uso:</b> Buscar evaluaciones.	<b>Versión:</b> 1.0	<b>Nro.:</b> 015
<b>Actor Principal:</b> Sistema.	<b>Actor Secundario:</b> No Aplica.	
<b>Críticidad:</b> <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja		
<b>Prioridad:</b> <input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Útil <input type="checkbox"/> Deseable		

<b>Complejidad:</b> <input type="checkbox"/> Extrem. Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Complejo <input checked="" type="checkbox"/> Mediano <input type="checkbox"/> Simple	
<b>Frecuencia:</b> <input type="checkbox"/> Siempre <input type="checkbox"/> Frecuentemente <input checked="" type="checkbox"/> A veces <input type="checkbox"/> Raramente <input type="checkbox"/> Una vez	
<b>Tipo de Caso de Uso:</b> <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto	
<b>Descripción:</b> Caso de uso que detalla la búsqueda de las evaluaciones que tenemos almacenados dentro del sistema.	
<b>Precondiciones:</b>  ▪ Debe haber un agente logueado.	
<b>Postcondiciones:</b>	Éxito: ▪ Búsqueda exitosa de las evaluaciones.
	Fracaso: ▪ Búsqueda no exitosa de las evaluaciones.
<b>Curso Normal</b>	
1. El sistema busca todas las evaluaciones almacenadas en el sistema de acuerdo al agente logueado.	
2. El sistema lista todas las evaluaciones encontradas.	2.a - La evaluación requerida no se encuentra listada. 2.a.1 - Se invoca el <b>CU014</b> . 2.a.2 - Se vuelve al paso 2.
3. El caso de uso finaliza.	
<b>Requerimientos No funcionales Especiales:</b> No aplica.	
<b>Asociaciones de Extensión:</b>  ▪ CU014 - Crear evaluación.	
<b>Asociaciones de Inclusión:</b> No aplica.	
<b>Caso de Uso de Generalización:</b> No aplica.	
<b>Referencias:</b> No aplica.	
<b>Observaciones:</b> No aplica.	

<b>Módulo:</b> ABM Assessments		<b>Iteración:</b> 1
<b>Nombre del Caso de Uso:</b> Modificar evaluación.	<b>Versión:</b> 1.0	<b>Nro.:</b> 016
<b>Actor Principal:</b> Agente del sistema.	<b>Actor Secundario:</b> No Aplica.	
<b>Criticidad:</b> <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja		
<b>Prioridad:</b> <input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Útil <input type="checkbox"/> Deseable		

<b>Complejidad:</b> <input type="checkbox"/> Extrem. Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Complejo <input checked="" type="checkbox"/> Mediano <input type="checkbox"/> Simple	
<b>Frecuencia:</b> <input type="checkbox"/> Siempre <input type="checkbox"/> Frecuentemente <input checked="" type="checkbox"/> A veces <input type="checkbox"/> Raramente <input type="checkbox"/> Una vez	
<b>Tipo de Caso de Uso:</b> <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto	
<b>Descripción:</b> Caso de uso que detalla cómo se realiza la modificación de una evaluación dentro del sistema.	
<b>Precondiciones:</b> <ul style="list-style-type: none"> <li>Debe haber un agente logueado.</li> </ul>	
<b>Postcondiciones:</b>	<b>Éxito:</b> <ul style="list-style-type: none"> <li>Modificación exitosa de la evaluación.</li> </ul>
	<b>Fracaso:</b> <ul style="list-style-type: none"> <li>Mensaje de error al intentar modificar la evaluación.</li> </ul>
Curso Normal	Cursos Alternativos
1. El sistema despliega en pantalla todas las evaluaciones almacenadas mediante la invocación al <b>CU015</b> .	
2. El agente selecciona la evaluación a modificar y presiona el botón "Editar Elemento".	
3. El sistema abre la ventana de edición con la evaluación ya cargada.	
4. El agente realiza los cambios de los datos y presiona el botón "Aceptar Cambios".	4.a - Se requiere editar las actividades de la evaluación. 4.a.1 - Se invoca al <b>CU011</b> . 4.a.2 - Se vuelve al paso 4.
5. El sistema verifica la consistencia de la información modificada.	5.a - No se seleccionó ningún momento de evaluación. 5.a.1 - El agente no selecciono el momento de la evaluación. 5.a.2 - El sistema informa con el mensaje <b>CU016-MSG01</b> . 5.b - No selecciono o introdujo una fecha para la evaluación. 5.b.1 - El agente no selecciono ni introdujo una fecha. 5.b.2 - El sistema informa con el mensaje <b>CU016-MSG02</b> . 5.c - No se ingresó ninguna descripción. 7.c.1 - El agente dejo el campo "Descripción" vacío. 7.c.2 - El sistema informa con el mensaje <b>CU016-MSG03</b> . 5.d - No se cargó una actividad. 7.d.1 - El agente no selecciono por lo menos una actividad. 5.d.2 - El sistema informa con el mensaje <b>CU016-MSG04</b> .

	5.e - El caso de uso continúa en el punto 4.
6. El sistema guarda los cambios de la evaluación en la base de datos.	
7. El caso de uso finaliza.	
<b>Requerimientos No funcionales Especiales:</b> No aplica.	
<b>Asociaciones de Extensión:</b>	
<ul style="list-style-type: none"> <li>CU011 - Buscar actividades.</li> </ul>	
<b>Asociaciones de Inclusión:</b>	
<ul style="list-style-type: none"> <li>CU015 - Buscar evaluaciones.</li> </ul>	
<b>Caso de Uso de Generalización:</b> No aplica.	
<b>Referencias:</b>	
<u><b>Mensajes empleados</b></u>	
CU016-MSG01 - Debe seleccionar un tipo de evaluación.	
CU016-MSG02 - Debe colocar una fecha de evaluación.	
CU016-MSG03 - La descripción de la evaluación no puede ser vacía.	
CU016-MSG04 - Debe al menos seleccionar una actividad para la evaluación.	
<b>Observaciones:</b> No aplica.	

<b>Módulo:</b> ABM Assessments		<b>Iteración:</b> 1
<b>Nombre del Caso de Uso:</b> Eliminar evaluación.	<b>Versión:</b> 1.0	<b>Nro.:</b> 017
<b>Actor Principal:</b> Agente del sistema.	<b>Actor Secundario:</b> No Aplica.	
<b>Criticidad:</b>	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja	
<b>Prioridad:</b>	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Útil <input type="checkbox"/> Deseable	
<b>Complejidad:</b>	<input type="checkbox"/> Extrem. Complejo <input checked="" type="checkbox"/> Mediano <input type="checkbox"/> Simple <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Complejo	
<b>Frecuencia:</b>	<input type="checkbox"/> Siempre <input type="checkbox"/> Frecuentemente <input checked="" type="checkbox"/> A veces <input type="checkbox"/> Raramente <input type="checkbox"/> Una vez	
<b>Tipo de Caso de Uso:</b>	<input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto	
<b>Descripción:</b>		
Caso de uso que describe como eliminar una evaluación del sistema.		



<b>Precondiciones:</b>	
<ul style="list-style-type: none"> <li>▪ Debe haber un agente logueado.</li> </ul>	
<b>Postcondiciones:</b>	<b>Éxito:</b> <ul style="list-style-type: none"> <li>▪ Borrado exitoso de la evaluación.</li> </ul>
	<b>Fracaso:</b> <ul style="list-style-type: none"> <li>▪ No se realiza el borrado de la evaluación.</li> </ul>
<b>Curso Normal</b>	<b>Cursos Alternativos</b>
1. El sistema despliega en pantalla todas las evaluaciones almacenadas mediante la invocación del <b>CU015</b> .	
2. El agente selecciona la evaluación a eliminar y presiona el botón “Eliminar elemento”.	
3. El sistema solicita una confirmación de eliminación de la evaluación.	
4. El agente acepta la confirmación.	4.a - El usuario cancela la eliminación. 4.a.1 - El caso de uso finaliza.
5. El sistema procede a eliminar la evaluación de la base de datos.	
6. El caso de uso finaliza.	
<b>Requerimientos No funcionales Especiales:</b> No aplica.	
<b>Asociaciones de Extensión:</b> No aplica.	
<b>Asociaciones de Inclusión:</b> <ul style="list-style-type: none"> <li>▪ CU015 - Buscar evaluaciones.</li> </ul>	
<b>Caso de Uso de Generalización:</b> No aplica.	
<b>Referencias:</b> No aplica.	
<b>Observaciones:</b> No aplica.	

## El módulo de reglas

<b>Módulo:</b> ABM Rules		<b>Iteración:</b> 1
<b>Nombre del Caso de Uso:</b> Crear regla.	<b>Versión:</b> 1.0	<b>Nro.:</b> 018
<b>Actor Principal:</b> Agente del sistema.	<b>Actor Secundario:</b> No Aplica.	
<b>Criticidad:</b> <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
<b>Prioridad:</b> <input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Útil <input type="checkbox"/> Deseable		
<b>Complejidad:</b> <input type="checkbox"/> Extrem. Complejo <input type="checkbox"/> Muy Complejo <input checked="" type="checkbox"/> Complejo <input type="checkbox"/> Mediano <input type="checkbox"/> Simple		
<b>Frecuencia:</b> <input type="checkbox"/> Siempre <input checked="" type="checkbox"/> Frecuentemente <input type="checkbox"/> A veces <input type="checkbox"/> Raramente <input type="checkbox"/> Una vez		
<b>Tipo de Caso de Uso:</b> <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
<b>Descripción:</b> Caso de uso que describe como crear una nueva regla para validar una evaluación dentro del sistema.		
<b>Precondiciones:</b> ■ Debe haber un agente logueado.		
<b>Postcondiciones:</b>	Éxito: ■ Creación correcta de la regla.	
	Fracaso: ■ Mensaje de error al intentar crear la regla.	
<b>Curso Normal</b>		<b>Cursos Alternativos</b>
1. El agente presiona el botón “Agregar Elemento” para crear una nueva regla.		
2. El sistema despliega la ventana de carga de datos de la nueva regla.		
3. El agente introduce los datos correspondientes y presiona el botón “Aceptar cambios”.		
4. El sistema verifica y valida que estén todos los datos necesarios para crear la regla.		4.a – La descripción se encuentra vacía. 4.a.1 – La descripción no fue ingresada por el agente. 4.a.2 – El sistema informa con el mensaje <b>CU018-MSG01</b> . 4.b – La regla no está bien construida. 4.b.1 – El usuario ingreso mal la estructura de la regla. 4.b.2 – El sistema informa con el mensaje <b>CU018-MSG02</b> . 4.c – Se produce una excepción al guardar la regla. 4.c.1 – El sistema informa con el mensaje <b>CU018-MSG03</b> . 4.d - El caso de uso continúa en el punto 3.

5. El sistema guardar la regla en la base de datos.	
6. El caso de uso finaliza.	
<b>Requerimientos No funcionales Especiales:</b> No aplica.	
<b>Asociaciones de Extensión:</b> No aplica.	
<b>Asociaciones de Inclusión:</b> No aplica.	
<b>Caso de Uso de Generalización:</b> No aplica.	
<b>Referencias:</b>	
<u><b>Mensajes empleados</b></u>	
CU018-MSG01 - La descripción de la regla no puede estar vacía.	
CU018-MSG02 - La regla no está correctamente formada.	
CU018-MSG03 - Falla en el guardado de la regla.	
<b>Observaciones:</b> No aplica.	

<b>Módulo:</b> ABM Rules		<b>Iteración:</b> 1
<b>Nombre del Caso de Uso:</b> Buscar reglas.	<b>Versión:</b> 1.0	<b>Nro.:</b> 019
<b>Actor Principal:</b> Sistema.	<b>Actor Secundario:</b> No Aplica.	
<b>Criticidad:</b> <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja		
<b>Prioridad:</b> <input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Útil <input type="checkbox"/> Deseable		
<b>Complejidad:</b> <input type="checkbox"/> Extrem. Complejo <input checked="" type="checkbox"/> Mediano <input type="checkbox"/> Simple <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Complejo		
<b>Frecuencia:</b> <input type="checkbox"/> Siempre <input type="checkbox"/> Frecuentemente <input checked="" type="checkbox"/> A veces <input type="checkbox"/> Raramente <input type="checkbox"/> Una vez		
<b>Tipo de Caso de Uso:</b> <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
<b>Descripción:</b> Caso de uso que detalla la búsqueda de las reglas que tenemos almacenados dentro del sistema.		
<b>Precondiciones:</b>  ▪ Debe haber un agente logueado.		
<b>Postcondiciones:</b>	Éxito: ▪ Búsqueda exitosa de las reglas.	
	Fracaso: ▪ Búsqueda no exitosa de las reglas.	

Curso Normal	Cursos Alternativos
1. El sistema busca todas las reglas almacenadas en el sistema.	
2. El sistema lista todas las reglas encontradas.	2.a - La regla buscada no se encuentra en la lista. 2.a.1 - Se llama al <b>CU018</b> . 2.a.2 - Se vuelve al paso 2.
3. El caso de uso finaliza.	
<b>Requerimientos No funcionales Especiales:</b> No aplica.	
<b>Asociaciones de Extensión:</b>  ▪ CU018 - Crear Regla.	
<b>Asociaciones de Inclusión:</b> No aplica.	
<b>Caso de Uso de Generalización:</b> No aplica.	
<b>Referencias:</b> No aplica.	
<b>Observaciones:</b> No aplica.	

<b>Módulo:</b> ABM Rules		<b>Iteración:</b> 1
<b>Nombre del Caso de Uso:</b> Modificar regla.	<b>Versión:</b> 1.0	<b>Nro.:</b> 020
<b>Actor Principal:</b> Agente del sistema.	<b>Actor Secundario:</b> No Aplica.	
<b>Criticidad:</b> <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja		
<b>Prioridad:</b> <input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Útil <input type="checkbox"/> Deseable		
<b>Complejidad:</b> <input type="checkbox"/> Extrem. Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Complejo <input checked="" type="checkbox"/> Mediano <input type="checkbox"/> Simple		
<b>Frecuencia:</b> <input type="checkbox"/> Siempre <input type="checkbox"/> Frecuentemente <input checked="" type="checkbox"/> A veces <input type="checkbox"/> Raramente <input type="checkbox"/> Una vez		
<b>Tipo de Caso de Uso:</b> <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
<b>Descripción:</b> Caso de uso que detalla cómo se realiza la modificación de una regla dentro del sistema.		
<b>Precondiciones:</b>  ▪ Debe haber un agente logueado.		
<b>Postcondiciones:</b>	Éxito: ▪ Modificación exitosa de la regla.	
	Fracaso:	

<p align="center">▪ Mensaje de error al intentar modificar la regla.</p>	
Curso Normal	Cursos Alternativos
1. El sistema despliega en pantalla todas las reglas almacenadas mediante el <b>CU019</b> .	
2. El agente selecciona la regla a modificar y presiona el botón “Editar Elemento”.	
3. El sistema abre la ventana de edición con la regla ya cargada.	
4. El agente realiza los cambios de los datos y presiona el botón “Aceptar Cambios”.	
5. El sistema verifica la consistencia de la información modificada.	5.a – La descripción se encuentra vacía. 5.a.1 – La descripción no fue ingresada por el agente. 5.a.2 – El sistema informa con el mensaje <b>CU020-MSG01</b> . 5.b – La regla no está bien construida. 5.b.1 – El usuario ingreso mal la estructura de la regla. 5.b.2 – El sistema informa con el mensaje <b>CU020-MSG02</b> . 5.c – Se produce una excepción al guardar la regla. 5.c.1 – El sistema informa con el mensaje <b>CU020-MSG03</b> . 5.d - El caso de uso continúa en el punto 4.
6. El sistema guarda los cambios de la regla en la base de datos.	
7. El caso de uso finaliza.	
<b>Requerimientos No funcionales Especiales:</b> No aplica.	
<b>Asociaciones de Extensión:</b> No aplica.	
<b>Asociaciones de Inclusión:</b> <ul style="list-style-type: none"> <li>▪ CU019 - Buscar reglas.</li> </ul>	
<b>Caso de Uso de Generalización:</b> No aplica.	
<b>Referencias:</b>  <u><b>Mensajes empleados</b></u>  <b>CU020-MSG01</b> - <i>La descripción de la regla no puede estar vacía.</i>  <b>CU020-MSG02</b> - <i>La regla no está correctamente formada.</i>  <b>CU020-MSG03</b> - <i>Falla en el guardado de la regla.</i>	
<b>Observaciones:</b> No aplica.	

<b>Módulo:</b> ABM Rules		<b>Iteración:</b> 1
<b>Nombre del Caso de Uso:</b> Eliminar regla.	<b>Versión:</b> 1.0	<b>Nro.:</b> 021
<b>Actor Principal:</b> Agente del sistema.		<b>Actor Secundario:</b> No Aplica.
<b>Criticidad:</b> <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja		
<b>Prioridad:</b> <input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Útil <input type="checkbox"/> Deseable		
<b>Complejidad:</b> <input type="checkbox"/> Extrem. Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Complejo <input checked="" type="checkbox"/> Mediano <input type="checkbox"/> Simple		
<b>Frecuencia:</b> <input type="checkbox"/> Siempre <input type="checkbox"/> Frecuentemente <input checked="" type="checkbox"/> A veces <input type="checkbox"/> Raramente <input type="checkbox"/> Una vez		
<b>Tipo de Caso de Uso:</b> <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
<b>Descripción:</b> Caso de uso que describe como eliminar una regla del sistema.		
<b>Precondiciones:</b> <ul style="list-style-type: none"> <li>Debe haber un agente logueado.</li> </ul>		
<b>Postcondiciones:</b>	<b>Éxito:</b> <ul style="list-style-type: none"> <li>Borrado exitoso de la regla.</li> </ul>	
	<b>Fracaso:</b> <ul style="list-style-type: none"> <li>Borrado no exitoso de la regla.</li> </ul>	
<b>Curso Normal</b>		<b>Cursos Alternativos</b>
1. El sistema despliega en pantalla todas las reglas almacenadas, mediante el caso de uso <b>CU019</b> .		
2. El agente selecciona la regla a eliminar y presiona el botón "Eliminar elemento".		
3. El sistema solicita una confirmación de eliminación de la regla.		
4. El agente acepta la confirmación.		4.a - El usuario cancela la eliminación. 4.a.1 - El caso de uso finaliza.
5. El sistema procede a eliminar la regla de la base de datos.		
6. El caso de uso finaliza.		
<b>Requerimientos No funcionales Especiales:</b> No aplica.		
<b>Asociaciones de Extensión:</b> No aplica.		
<b>Asociaciones de Inclusión:</b> <ul style="list-style-type: none"> <li>CU019 - Buscar reglas.</li> </ul>		
<b>Caso de Uso de Generalización:</b> No aplica.		

<b>Referencias:</b> No aplica.
<b>Observaciones:</b> No aplica.

## El módulo de conjuntos de reglas

<b>Módulo:</b> ABM Rules Set		<b>Iteración:</b> 1
<b>Nombre del Caso de Uso:</b> Creación de conjunto de reglas.	<b>Versión:</b> 1.0	<b>Nro.:</b> 022
<b>Actor Principal:</b> Agente del Sistema.		<b>Actor Secundario:</b> No Aplica.
<b>Críticidad:</b> <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja		
<b>Prioridad:</b> <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Útil <input type="checkbox"/> Deseable		
<b>Complejidad:</b> <input type="checkbox"/> Extrem. Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Complejo <input checked="" type="checkbox"/> Mediano <input type="checkbox"/> Simple		
<b>Frecuencia:</b> <input type="checkbox"/> Siempre <input checked="" type="checkbox"/> Frecuentemente <input type="checkbox"/> A veces <input type="checkbox"/> Raramente <input type="checkbox"/> Una vez		
<b>Tipo de Caso de Uso:</b> <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
<b>Descripción:</b> Caso de uso que describe como crear un nuevo conjunto de reglas para validar una evaluación dentro del sistema.		
<b>Precondiciones:</b> ■ El usuario debe estar logueado.		
<b>Postcondiciones:</b>	Éxito: ■ Creación correcta del conjunto de reglas.	
	Fracaso: ■ Mensaje de error al intentar crear un conjunto de reglas.	
<b>Curso Normal</b>		<b>Cursos Alternativos</b>
1. El agente presiona el botón “Agregar Elemento” para crear un nuevo conjunto de reglas.		
2. El sistema despliega la ventana de alta de conjuntos de reglas.		
3. El sistema despliega el listado de reglas disponibles para cargar dentro del conjunto mediante la llamada al <b>CU019</b> .		
4. El agente carga la descripción del conjunto y las reglas que va a contener este.		
5. El agente presiona el botón “Aceptar Cambios”.		5.a - El agente no cargo la descripción del conjunto. 5.a.1 - El sistema informa del error de falta de descripción con el mensaje <b>CU022-MSG01</b> . 5.a.2 - El flujo vuelve al paso 4. 5.b - El agente no cargo como mínimo una regla valida al conjunto. 5.b.1 - El sistema informa del error de falta de reglas con el mensaje <b>CU022-MSG02</b> . 5.b.2 - El flujo vuelve al paso 4.



6. El sistema guarda el conjunto en la base de datos.	
7. El caso de uso finaliza.	
<b>Requerimientos No funcionales Especiales:</b> No aplica.	
<b>Asociaciones de Extensión:</b> No aplica.	
<b>Asociaciones de Inclusión:</b> <ul style="list-style-type: none"> <li>CU019 - Buscar reglas.</li> </ul>	
<b>Caso de Uso de Generalización:</b> No aplica.	
<b>Referencias:</b> <b>Mensajes empleados</b> <p>CU022-MSG01 - La descripción del conjunto de reglas no puede ser vacío.</p> <p>CU022- MSG02 - Debe seleccionar al menos una regla para el conjunto.</p>	
<b>Observaciones:</b> No aplica.	

<b>Módulo:</b> ABM Rules Set		<b>Iteración:</b> 1
<b>Nombre del Caso de Uso:</b> Buscar conjuntos de reglas.	<b>Versión:</b> 1.0	<b>Nro.:</b> 023
<b>Actor Principal:</b> Sistema.	<b>Actor Secundario:</b> No Aplica.	
<b>Criticidad:</b> <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja		
<b>Prioridad:</b> <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Útil <input type="checkbox"/> Deseable		
<b>Complejidad:</b> <input type="checkbox"/> Extrem. Complejo <input type="checkbox"/> Muy Complejo <input checked="" type="checkbox"/> Mediano <input type="checkbox"/> Simple		
<b>Frecuencia:</b> <input type="checkbox"/> Siempre <input checked="" type="checkbox"/> Frecuentemente <input type="checkbox"/> A veces <input type="checkbox"/> Raramente <input type="checkbox"/> Una vez		
<b>Tipo de Caso de Uso:</b> <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
<b>Descripción:</b> Caso de uso que detalla la búsqueda de los conjuntos de reglas que tenemos almacenados dentro del sistema		
<b>Precondiciones:</b> <ul style="list-style-type: none"> <li>El usuario debe estar logueado.</li> </ul>		

<b>Postcondiciones:</b>	Éxito:	<ul style="list-style-type: none"> <li>La lista de conjuntos de reglas desplegada.</li> </ul>
	Fracaso:	<ul style="list-style-type: none"> <li>Mensaje de error al intentar buscar los conjuntos de reglas.</li> </ul>
<b>Curso Normal</b>		<b>Cursos Alternativos</b>
1. El sistema busca todos los conjuntos de reglas almacenados de acuerdo al agente logueado.		
2. El sistema lista todos los conjuntos de reglas encontrados.		2.a - No se encuentra el conjunto requerido. 2.a.1 - Se llama a <b>CU022</b> para dar de alta un nuevo conjunto. 2.a.2 - Se vuelve al paso 2.
3. El caso de uso finaliza.		
<b>Requerimientos No funcionales Especiales:</b> No aplica.		
<b>Asociaciones de Extensión:</b>		
<ul style="list-style-type: none"> <li>CU022 - Crear conjunto de reglas.</li> </ul>		
<b>Asociaciones de Inclusión:</b> No aplica.		
<b>Caso de Uso de Generalización:</b> No aplica.		
<b>Referencias:</b> No aplica.		
<b>Observaciones:</b> No aplica.		

<b>Módulo:</b> ABM Rule Set		<b>Iteración:</b> 1
<b>Nombre del Caso de Uso:</b> Modificación de conjunto de reglas.	<b>Versión:</b> 1.0	<b>Nro.:</b> 024
<b>Actor Principal:</b> Agente del Sistema.	<b>Actor Secundario:</b> No Aplica.	
<b>Criticidad:</b> <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
<b>Prioridad:</b> <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Útil <input type="checkbox"/> Deseable		
<b>Complejidad:</b> <input type="checkbox"/> Extrem. Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Complejo <input checked="" type="checkbox"/> Mediano <input type="checkbox"/> Simple		
<b>Frecuencia:</b> <input type="checkbox"/> Siempre <input checked="" type="checkbox"/> Frecuentemente <input type="checkbox"/> A veces <input type="checkbox"/> Raramente <input type="checkbox"/> Una vez		
<b>Tipo de Caso de Uso:</b> <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
<b>Descripción:</b> Caso de uso que detalla cómo se realiza la modificación de un conjunto de reglas dentro del sistema.		
<b>Precondiciones:</b>		

<ul style="list-style-type: none"> <li>El usuario debe estar logueado.</li> </ul>	
<b>Postcondiciones:</b>	<b>Éxito:</b> <ul style="list-style-type: none"> <li>Modificación correcta del conjunto de reglas.</li> </ul>
	<b>Fracaso:</b> <ul style="list-style-type: none"> <li>Mensaje de error al intentar modificar un conjunto de reglas.</li> </ul>
<b>Curso Normal</b>	<b>Cursos Alternativos</b>
1. El sistema despliega en pantalla todos los conjuntos de reglas almacenados mediante la invocación al <b>CU023</b> .	
2. El agente selecciona el conjunto de regla a modificar y presiona el botón “Editar Elemento”.	
3. El sistema abre la ventana de edición con el conjunto de regla ya cargado.	
4. El agente carga la descripción del conjunto o las reglas que va a contener este.	4.a - La regla no se encuentra en el listado. 4.a.1 - El agente llama al <b>CU019</b> para buscar la regla. 4.a.2 - El flujo retorna al paso 3.
5. El agente presiona el botón “Aceptar Cambios”.	5.a - El agente no cargo la descripción del conjunto. 5.a.1 - El sistema informa del error de falta de descripción con el mensaje <b>CU024-MSG01</b> . 5.a.2 - El flujo vuelve al paso 4. 5.b - El agente no cargo como mínimo una regla valida al conjunto. 5.b.1 - El sistema informa del error de falta de reglas con el mensaje <b>CU024-MSG02</b> . 5.b.2 - El flujo vuelve al paso 4.
6. El sistema guarda el conjunto en la base de datos.	
7. El caso de uso finaliza.	
<b>Requerimientos No funcionales Especiales:</b> No aplica.	
<b>Asociaciones de Extensión:</b>	
<ul style="list-style-type: none"> <li>CU019 - Buscar reglas.</li> </ul>	
<b>Asociaciones de Inclusión:</b>	
<ul style="list-style-type: none"> <li>CU023 - Buscar conjuntos de reglas.</li> </ul>	
<b>Caso de Uso de Generalización:</b> No aplica.	
<b>Referencias:</b>	
<u><b>Mensajes empleados</b></u>	
<b>CU022-MSG01</b> - La descripción del conjunto de reglas no puede ser vacío.	
<b>CU022- MSG02</b> - Debe seleccionar al menos una regla para el conjunto.	

**Observaciones:** No aplica.

<b>Módulo:</b> ABM Rule Set		<b>Iteración:</b> 1
<b>Nombre del Caso de Uso:</b> Eliminar conjunto de reglas.	<b>Versión:</b> 1.0	<b>Nro.:</b> 025
<b>Actor Principal:</b> Agente del Sistema.	<b>Actor Secundario:</b> No Aplica.	
<b>Criticidad:</b> <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja		
<b>Prioridad:</b> <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Útil <input type="checkbox"/> Deseable		
<b>Complejidad:</b> <input type="checkbox"/> Extrem. Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Complejo <input checked="" type="checkbox"/> Mediano <input type="checkbox"/> Simple		
<b>Frecuencia:</b> <input type="checkbox"/> Siempre <input checked="" type="checkbox"/> Frecuentemente <input type="checkbox"/> A veces <input type="checkbox"/> Raramente <input type="checkbox"/> Una vez		
<b>Tipo de Caso de Uso:</b> <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
<b>Descripción:</b> Caso de uso que describe como eliminar un conjunto de reglas del sistema.		
<b>Precondiciones:</b> ■ El usuario debe estar logueado.		
<b>Postcondiciones:</b>	<b>Éxito:</b> ■ Eliminación correcta del conjunto de reglas.	
	<b>Fracaso:</b> ■ Mensaje de error al intentar eliminar un conjunto de reglas.	
<b>Curso Normal</b>		<b>Cursos Alternativos</b>
1. El sistema despliega el listado de los conjuntos de reglas que tenemos dentro del sistema mediante la invocación a <b>CU023</b> .		
2. El agente selecciona el conjunto a eliminar y presiona el botón "Eliminar elemento".		
3. El sistema solicita una confirmación de eliminación del conjunto.		
4. El agente acepta la confirmación.		4.a - El usuario cancela la eliminación. 4.a.1 - El caso de uso finaliza.
5. El sistema procede a eliminar el conjunto de la base de datos.		
6. El caso de uso finaliza.		
<b>Requerimientos No funcionales Especiales:</b> No aplica.		

<b>Asociaciones de Extensión:</b> No aplica.
<b>Asociaciones de Inclusión:</b> <ul style="list-style-type: none"><li>CU023 - Buscar conjunto de reglas.</li></ul>
<b>Caso de Uso de Generalización:</b> No aplica.
<b>Referencias:</b> No aplica.
<b>Observaciones:</b> No aplica.

### El módulo de validación

<b>Módulo:</b> Validate Assessment		<b>Iteración:</b> 1
<b>Nombre del Caso de Uso:</b> Validar una evaluación.	<b>Versión:</b> 1.0	<b>Nro.:</b> 026
<b>Actor Principal:</b> Agente del sistema.		<b>Actor Secundario:</b> No Aplica.
<b>Críticidad:</b> <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
<b>Prioridad:</b> <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Útil <input type="checkbox"/> Deseable		
<b>Complejidad:</b> <input checked="" type="checkbox"/> Extrem. Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Complejo <input type="checkbox"/> Mediano <input type="checkbox"/> Simple		
<b>Frecuencia:</b> <input type="checkbox"/> Siempre <input checked="" type="checkbox"/> Frecuentemente <input type="checkbox"/> A veces <input type="checkbox"/> Raramente <input type="checkbox"/> Una vez		
<b>Tipo de Caso de Uso:</b> <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
<b>Descripción:</b> Caso de uso que detalla los pasos para validar una evaluación dado un conjunto de reglas seleccionadas por el agente.		
<b>Precondiciones:</b> <ul style="list-style-type: none"> <li>▪ Debe haber un agente logueado.</li> </ul>		
<b>Postcondiciones:</b>	<b>Éxito:</b> <ul style="list-style-type: none"> <li>▪ Reporte de validación de la evaluación.</li> </ul>	
	<b>Fracaso:</b> <ul style="list-style-type: none"> <li>▪ Mensaje de error al intentar validar una evaluación o de parsear una regla.</li> </ul>	
<b>Curso Normal</b>		<b>Cursos Alternativos</b>
1. El agente selecciona una evaluación a validar mediante la invocación al <b>CU015</b> y un conjunto de reglas mediante la invocación al <b>CU023</b> que van a aplicarse en la validación de la evaluación seleccionada.		
2. El sistema limpia el área donde van a desplegarse los resultados de la validación.		
3. El sistema toma la evaluación y la crea un modelo ontológico en memoria conteniendo los datos de la evaluación.		
4. El sistema toma cada una de las reglas que tenemos en el conjunto y las parsea para crear una regla propia del framework a utilizar.		3.a - La regla no parsea como una regla propia del framework. 3.a.1 - El sistema informa del error de parseo con el mensaje <b>CU026-MSG01</b> . 3.a.2 - Se vuelve al paso 4.
5. El sistema toma la regla y la aplica al modelo ontológico creado.		
6. El sistema valida el modelo de inferencia creado a partir de aplicar la regla.		

7. El sistema muestra el resultado de aplicar la regla en el área de resultado establecida.	
8. El caso de uso finaliza.	
<b>Requerimientos No funcionales Especiales:</b> No aplica.	
<b>Asociaciones de Extensión:</b> No aplica.	
<b>Asociaciones de Inclusión:</b> <ul style="list-style-type: none"> <li>▪ CU015 - Buscar evaluaciones.</li> <li>▪ CU023 - Buscar conjuntos de reglas.</li> </ul>	
<b>Caso de Uso de Generalización:</b> No aplica.	
<b>Referencias:</b>  <u><b>Mensajes empleados</b></u>  <b>CU026-MSG01</b> - <i>Falla en el parseo de la regla.</i>	
<b>Observaciones:</b> No aplica.	