



FACULTAD REGIONAL SANTA FE

Aplicación de Software basada en Tecnología
de la Web Semántica para la Asignación de
Recursos Humanos a Eventos de Catering

Proyecto final de carrera
Ingeniería en sistemas de información

2016

Colautti, Alejandro
Mathurin, Andrés

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

ÍNDICE:

CAPÍTULO 1: INTRODUCCIÓN..... PÁG. 4

- 1.1) PROBLEMÁTICA A RESOLVER
- 1.2) FUNDAMENTACIÓN DEL PROYECTO
- 1.3) OBJETIVOS DEL PROYECTO
- 1.4) APORTES DEL PROYECTO
- 1.5) METODOLOGÍA DE TRABAJO

CAPÍTULO 2: MARCO TEÓRICO..... PÁG. 12

- 2.1) SISTEMAS DE RECOMENDACIÓN
 - 2.1.1) FILTRADO BASADO EN CONTENIDO
 - 2.1.2) FILTRADO COLABORATIVO
- 2.2) TECNOLOGÍAS DE LA WEB SEMÁNTICA
 - 2.2.1) LENGUAJE OWL
 - 2.2.2) REGLAS SWRL
 - 2.2.3) CONSULTA DE ONTOLOGÍAS: LENGUAJE SPARQL
- 2.3) METODOLOGÍA DE DESARROLLO DE ONTOLOGÍAS: EDON

CAPÍTULO 3: ANÁLISIS DEL DOMINIO Y REQUERIMIENTOS..... PÁG. 22

- 3.1) LOS EVENTOS
- 3.2) LOS EMPLEADOS
- 3.3) REQUERIMIENTOS FUNCIONALES DEL SISTEMA

CAPÍTULO 4: DISEÑO Y ARQUITECTURA..... PÁG. 29

- 4.1) ARQUITECTURA
- 4.2) ESTRUCTURA: CLASES DEL MODELO DE DOMINIO
- 4.3) ESTRUCTURA: CLASES DE COMPONENTES DE CAPAS DE LA APLICACIÓN
- 4.4) COMPORTAMIENTO

CAPÍTULO 5: CONTRUCCIÓN DE LA ONTOLOGÍA..... PÁG. 48

- 5.1) PRIMER PASO: PROCESO DE SELECCIÓN DE REQUERIMIENTOS
- 5.2) SEGUNDO PASO: DESARROLLO DE LA ONTOLOGÍA

CAPÍTULO 6: PRUEBAS DE FUNCIONALIDADES..... PÁG. 61

- 6.1) AUTENTICAR USUARIO
- 6.2) GESTIÓN DE ROLES Y HABILIDADES
- 6.3) GESTIÓN DE EMPLEADOS
- 6.4) GESTIÓN DE EVENTOS
- 6.5) ASIGNAR EMPLEADOS A EVENTOS
- 6.6) ASIGNAR DESEMPEÑO A EMPLEADOS

CAPÍTULO 7: CONCLUSIÓN..... PÁG. 73

- 7.1) INSTALACIÓN DE APLICACIÓN PARA REALIZAR PRUEBAS
- 7.2) CONCLUSIONES FINALES

BIBLIOGRAFÍA..... PÁG. 75

ANEXO 1: HERRAMIENTAS UTILIZADAS..... PÁG. 77

CAPÍTULO 1: INTRODUCCIÓN

1.1) PROBLEMÁTICA A RESOLVER

En una empresa de servicios, el principal recurso productivo son las personas que los producen. Por lo tanto, los recursos humanos son el componente económico más importante y el factor de éxito de la misma ya que la ventaja competitiva está relacionada positivamente sobre todo con las capacidades de los recursos humanos (Gómez Paz, J. 2010). Así, la correcta asignación de estos recursos a las diferentes tareas de la organización se vuelve fundamental para mantenerse competitiva.






Figura 1: Logo de la empresa Gut Catering

En particular la empresa Gut Catering (figura 1) (<http://www.grupogula.com.ar>) de organización de eventos, ofrece servicios de eventos ya sean particulares, sociales o empresariales (cumpleaños, casamientos, aniversarios, inauguraciones, etc.), que requieren gran cantidad de recursos humanos, altamente sincronizados y suficientemente capacitados para realizar la tarea asignada. Es de destacar que el personal con el que cuenta la empresa es en un 80% temporario siendo solamente un 20% empleados fijos. La asignación de recursos humanos a eventos consiste en asociar a cada una de las etapas que componen un evento, un equipo de recursos humanos disponible cuyos horarios y habilidades se adapten a aquellas que demanda dicha etapa con el fin de que la misma pueda ser llevada a cabo en forma eficiente y eficaz. Dada la gran cantidad de personal afectado y la condición de temporalidad del mismo, hacen que la decisión de este tipo de asignación no sea trivial y conlleva el análisis de grandes volúmenes de datos y de condiciones que deben ser tenidas en cuenta por la persona encargada de tomar esa decisión. Si bien existen

aplicaciones de software cuyo propósito se asemeja al de asignación de recursos humanos, estos son de tipo reclutamiento, es decir, proveen una herramienta útil para incorporar nuevo personal a una organización a partir de la carga de datos correspondiente a los requerimientos de un determinado puesto y las características de los aspirantes que se adecuen a los mismos. Ejemplos de esos sistemas son mostrados en la tabla I.

Tabla I: Software comercial para la asignación de RRHH

Logo	Web	descripción
	http://www.jobvite.com/	Aplicación que permite a través de redes sociales y aplicaciones móviles ofertar puestos de trabajos y postularse como candidato a dichos puestos. Presenta servicios para hacer entrevistas a través de videoconferencia.
	http://www.greenhouse.io/	Herramienta para reclutar personal en una compañía. Encontrar candidatos y realizar entrevistas.
	http://www.icims.com/	Herramienta para reclutar personal, encontrar los mejores candidatos a través de la red y realizar contrataciones.

Sin embargo, para la problemática de la empresa Gut Catering surge la necesidad de poder reubicar, en forma objetiva, recursos humanos ya contratados a los eventos que se desarrollan. Para dar solución a esta situación hace falta implementar un sistema recomendador, que actúe sobre un volumen de datos correspondiente a los recursos humanos actuales y “recomiende” una asignación posible de los mismos a los eventos que la empresa lleva a cabo para sus clientes con el fin de prestar sus servicios, evitando posibles inconsistencias o ambigüedades que puedan surgir a partir de la realización de dicho procedimiento en forma completamente manual por parte de una única persona (Isaias, P., 2010). Para determinar qué empleado es el más indicado para ser

asignado al evento y sus respectivas etapas, el sistema basará su recomendación en los horarios de trabajo, aptitudes, relaciones con sus pares, y experiencia, implementando además un mecanismo de puntajes asociados a cada uno de los empleados que consiste en un coeficiente que se verá favorecido o perjudicado de acuerdo al desempeño en eventos anteriores.

Para representar dicho conocimiento, necesario para tomar la decisión de asignación de recursos humanos, es deseable contar con una estructura de representación que sea flexible y que provea mecanismos de razonamiento que sean capaces de evaluar diferentes dimensiones y que sean fáciles de ser modificables. En este sentido, las nuevas tecnologías de la web semántica, cuyo referente principal son las ontologías, surgen como una alternativa de solución ya que permiten describir y comparar las diferentes dimensiones implicadas en la asignación de los recursos humanos con el conocimiento sobre los mismos (Richter, S. y Tolksdorf, R., 2009).

Las ontologías representan conocimiento como un conjunto de conceptos o términos básicos dentro de un área temática y las relaciones entre pares de conceptos (Gómez-Pérez y colab., 2004). Son utilizadas para modelar un dominio y proveer razonamiento sobre los conceptos que lo componen permitiendo definir extensiones del vocabulario, es decir, una ontología no sólo incluye los términos definidos explícitamente en ella, sino también aquellos términos que pueden ser deducidos a través de reglas. Las ontologías proveen un marco estructural para organizar información. Para este caso particular, el dominio del problema incluirá todos los términos necesarios que comprenden los recursos humanos y eventos, así como sus propiedades y relaciones relevantes para llevar a cabo la inferencia que permita a la aplicación realizar la asignación correspondiente. Además el uso de ontologías, reduce los tiempos de búsqueda, lo que da como resultado reducción de costos y tiempo invertido por las organizaciones lo que es un factor importante para el éxito de una organización eficiente.

1.2) FUNDAMENTACIÓN DEL PROYECTO

Este proyecto se basa en la utilización de la ingeniería ontológica como tecnología de la web semántica para la resolución de problemas empresariales. En este caso, la empresa Gut Catering tiene la necesidad de recurrir a una herramienta de software que de soporte a la asignación de recursos humanos a eventos, procedimiento que hasta el momento se lleva a cabo en forma totalmente manual y a cargo de una sola persona, lo que conlleva mayores tiempos, riesgos de desorganización y quizás a la asignación inadecuada de los recursos humanos disponibles. Con este proyecto se pretende dar solución a esta situación, ya que el producto será un sistema recomendador que permita, a quien se encargue de llevar a cabo la asignación definitiva de empleados a un evento, contar con posibles soluciones objetivas que faciliten y agilicen el proceso de decisión.

Como mencionamos anteriormente, para el desarrollo de este sistema recomendador se tomarán elementos de la web semántica. La web semántica es una Web extendida, dotada de mayor significado en la que cualquier usuario en Internet podrá encontrar respuestas a sus preguntas de forma más rápida y sencilla gracias a una información mejor definida. Al dotar a la Web de más significado y, por lo tanto, de más semántica, se pueden obtener soluciones a problemas habituales en la búsqueda de información gracias a la utilización de una infraestructura común, mediante la cual, es posible compartir, procesar y transferir información de forma sencilla. Esta Web extendida y basada en el significado, se apoya en lenguajes universales que resuelven los problemas ocasionados por una Web carente de semántica en la que, en ocasiones, el acceso a la información se convierte en una tarea difícil y frustrante.

Esta semántica se logra al utilizar el concepto de ontologías. Desde un punto de vista pragmático, una ontología puede ser definida como la representación de elementos basada en cuatro tipos de componentes de modelado (Maedche, 2002): conceptos, roles, restricciones e individuos. Los conceptos representan clases de objetos. Los roles describen relaciones binarias entre conceptos, de esta forma también permiten describir propiedades de los conceptos. Las restricciones son usadas para expresar las propiedades de los roles por ejemplo, cardinalidad. Los individuos representan instancias de clases. También es posible usar axiomas y reglas para inferir nueva información. Los axiomas son sentencias lógicas siempre verdaderas que expresan las

propiedades del modelo. Las reglas son sentencias lógicas que expresan características del dominio, por ejemplo reglas de negocio.

Gracias a la semántica en la Web, el software es capaz de procesar su contenido, razonar con este, combinarlo y realizar deducciones lógicas para resolver problemas cotidianos automáticamente.

OWL (Web Ontology Language) (W3C OWL Working Group, 2012) es el lenguaje estándar para definir ontologías, sin embargo este lenguaje no aporta los elementos necesarios para realizar deducciones sobre las definiciones ontológicas, se hace necesario combinar OWL con otros formalismos de representación de reglas lógicas. Uno de estos formalismos es SWRL (Semantic Web Rule Language) (O'Connor y col., 2005), el cual provee la habilidad de expresar reglas como cláusulas de HORN en términos de conceptos OWL. Así, para poder extraer información de las ontologías OWL es necesario contar con un lenguaje de consulta. El lenguaje más usado y con mayor poder es SQWRL (O'Connor y Das, 2009) el cual está basado en el lenguaje SWRL de reglas lógicas utilizando los fundamentos semánticos de SWRL como también su sustento formal. Este lenguaje contiene un conjunto de operadores novedosos que pueden ser utilizados para ejecutar operaciones de clausura para permitir formas limitadas de negación como ser verdadero-falso, agregación y conteos.

La ingeniería Ontológica se refiere al conjunto de actividades que son necesarias en el proceso de desarrollo de una ontología, el ciclo de vida de una ontología, así como a las metodologías, herramientas y lenguajes necesarios para la construcción de ontologías.

El sistema contará con una ontología subyacente conteniendo las definiciones y relaciones necesarias para realizar las inferencias lógicas pertinentes que permitan a la aplicación llevar a cabo una asignación eficiente de recursos a eventos.

1.3) OBJETIVOS DEL PROYECTO

El objetivo general de este proyecto consiste en diseñar e implementar una aplicación de software basada tecnologías de la web semántica, que de soporte a la asignación eficiente de recursos humanos a eventos de catering organizados por la empresa GUT CATERING.

También enumeramos a continuación los siguientes objetivos específicos a cumplir:

- Analizar e Investigar herramientas que permitan implementar ontologías dentro de una aplicación de software.
- Analizar algoritmos de asignación de recursos implementados en los sistemas recomendadores.
- Analizar los requerimientos del sistema para determinar características y atributos de los eventos y recursos humanos a gestionar.
- Identificar usuarios potenciales y definir roles para los mismos.
- Diseñar la gestión de eventos, empleados y la asignación correspondiente entre ambos.
- Implementar y probar el sistema diseñado generando casos de prueba.

1.4) APORTES DE ESTE PROYECTO

Con la implementación de este sistema la empresa mejora la gestión de recursos humanos asignados a los eventos. Al automatizarse la recomendación de personal teniendo en cuenta las características del evento como también las habilidades propias de cada postulante se agiliza la selección del personal más apto para cada evento. Además el sistema cuenta con un mecanismo de “scoring” (puntaje) que permite puntuar a los empleados de acuerdo a su desempeño logrando que los mismos actúen en función de alcanzar mayor “scoring” a la vez que le brinda al responsable de la asignación, contar con más datos para lograr más calidad en el servicio prestado al utilizar los mejores empleados. Finalmente, la implementación del sistema trae la ventaja de realizar la tarea de asignación del personal a los eventos de una manera rápida y simple mejorando la calidad de todo el proceso de la empresa.

1.5) METODOLOGÍA DE TRABAJO

Para realizar este proyecto optamos por usar una metodología que consiste en una serie de etapas, comenzando por una investigación previa con el fin de informarnos más acerca de las ontologías y de cómo emplearlas haciendo un uso óptimo de la misma. Además, como el sistema tiene un objetivo similar al de un sistema recomendador, se investigó sobre estos, conociendo los mecanismos que se utilizan en ellos y así, poder implementar uno haciendo uso de las ontologías. Los temas investigados forman parte de nuestro marco teórico para la realización del proyecto el cual se encuentra descrito en el capítulo 2.

Como este sistema utiliza una ontología, la cual constituye una herramienta novedosa para ser utilizada en una aplicación de gestión de recursos humanos, decidimos apuntar a un desarrollo en forma de cascada sin comprometernos a emplear un proceso de desarrollo riguroso que no haya sido pensado para tener en cuenta la especificación de una ontología subyacente al sistema.

Para llevar a cabo el módulo de este sistema tenemos una necesidad del cliente bien definida. En este caso la automatización de la gestión de recursos humanos. Utilizamos entonces un proceso de desarrollo de cascada donde las etapas se desarrollan de manera secuencial, dando lugar a bucles de retroalimentación entre cada una de las fases para que a medida que se descubran problemas sea posible “volver atrás” y realizar las modificaciones pertinentes. A continuación describimos brevemente las etapas en que se fue desarrollando el proyecto, las que además, serán profundizadas a lo largo del informe:

- Primera Etapa, Análisis de Requerimientos: Como la mayoría de las variantes de esquemas de desarrollo en cascada, nuestro proyecto comienza con una etapa de análisis de requerimientos. En esta primera etapa se recopila toda la información acerca de lo que el cliente necesita y se define en términos claros la problemática que el sistema debe ser capaz de resolver. En el análisis se deben afianzar las reglas de negocio, restricciones y funciones que se deben soportar en el sistema. Para llevar a cabo la captura de los requerimientos del sistema de información se llevaron a cabo entrevistas con las personas responsables de Gut Catering con el fin de que nos planteen sus necesidades y expectativas de los posibles usuarios. Para la especificación de los requerimientos

funcionales soportados por el sistema se definieron escenarios y los casos de uso correspondientes. El análisis se encuentra detallado en el capítulo 3 del informe. El mismo está orientado a un análisis de conceptos, ya que son claves para el desarrollo de un sistema de información basado en ontología.

- Segunda Etapa, Diseño: Esta etapa consiste en definir la arquitectura de hardware y software, los componentes, módulos, interfaces y datos para soportar los requerimientos especificados en la etapa anterior. También abarca a la selección de un entorno de desarrollo y el lenguaje de programación. La salida de esta etapa es una o más especificaciones de diseño las cuales son utilizadas en la etapa de implementación. La arquitectura y las clases resultantes se exponen en el capítulo 4 de este informe. Finalizando esta etapa y antes de comenzar con la siguiente es donde aplicamos la metodología EDON, la cual utilizamos especificar y desarrollar la ontología. Para dar comienzo a la aplicación de la misma hace falta definir previamente cuál de los requerimientos especificados en la primera etapa será soportado por la ontología. Esta parte del proyecto se desarrolla en el capítulo 5 del informe.
- Tercera Etapa, Implementación: Esta etapa consiste en construir el sistema a partir de las especificaciones realizadas en las etapas anteriores. La salida de esta etapa es el módulo del sistema construido acorde a estándares de codificación, probado e integrado para satisfacer los requerimientos de la arquitectura. En el capítulo 6 exponemos algunas de las interfaces gráficas más importantes que fueron desarrolladas y se explican cómo son soportadas las funcionalidades.
- Cuarta Etapa, Testing: En esta etapa, tanto componentes individuales como integrados se verifican metódicamente con el fin de encontrar errores y asegurar que se alcanzan los requerimientos especificados en la primera etapa. Para este proyecto se realizaron pruebas sobre el módulo implementado para verificar el funcionamiento confiable del sistema y la correcta inferencia a partir de las reglas para la asignación de empleados a eventos, entre otras. Con la etapa de testing se da por terminado este proyecto.

CAPÍTULO 2: MARCO TEÓRICO

El sistema desarrollado funciona como una herramienta de software que se ocupa de almacenar datos conocidos y a partir de los mismos, mediante las inferencias que provee una ontología, se crea el conocimiento necesario para dar soporte al proceso de toma de decisiones en donde se deben elegir alternativas de acción, con el propósito de alcanzar uno o varios objetivos definidos. Los sistemas que se ocupan de tareas de esta naturaleza son conocidos como Sistemas de Soporte a Decisiones (DSS: Decision Support System).

En esta etapa del informe, nos ocuparemos de desarrollar el grupo central de conceptos que fueron necesarios para el desarrollo de este proyecto.

2.1) SISTEMAS DE RECOMENDACIÓN

Como mencionamos en la problemática a resolver, existen sistemas llamados sistemas de reclutamiento que resuelven la problemática de buscar recursos humanos para asignarlos a puestos de trabajo. Si bien en nuestro caso el problema consiste en considerar empleados que ya son parte de la empresa en lugar de buscar cubrir nuevos puestos, podemos ver que en ambos casos juega un papel importante la recomendación, ya sea de un posible nuevo empleado para un determinado puesto, como de un empleado existente para participar de una actividad en la que se buscan determinadas características. Por eso comenzamos nuestra investigación acerca de cómo funcionan los llamados sistemas de recomendación con el fin de incorporar algunas de sus estrategias para poder realizar recomendaciones en nuestra propia aplicación.

Los sistemas de recomendación son aquellos sistemas que buscan predecir qué nivel de valoración o preferencia le asignará un usuario a algo en particular (Ricci, Rokach, Shapira, 2011) e ir aprendiendo a partir de las selecciones que haga el usuario. Estos se consideran una subclase de los llamados sistemas de filtrado de información, que son aquellos cuyo propósito consiste en, a partir de un flujo de información, eliminar aquello que sea redundante o considerado como información no deseada utilizando métodos (semi)automatizados o computarizados para así poder presentar al usuario aquello que solo sea relevante.

Los sistemas de recomendación son una buena alternativa frente al uso de algoritmos de búsqueda cuando se espera que el resultado de la misma arroje gran cantidad de información, ya que al filtrar y seleccionar parte de estos resultados, ayudan a los usuarios a descubrir ítems que podrían no haber encontrado de otra manera.

Existen dos alternativas por las cuales estos sistemas producen listas de recomendación: (i) mecanismos de filtrado colaborativo y (ii) filtrado basado en contenidos (Jafarkarimi, Sim, Saadatdoost, 2012).

Los enfoques de filtrado colaborativo construyen un modelo a partir de comportamientos previos de un usuario (por ejemplo teniendo en cuenta productos que fueron comprados o seleccionados anteriormente y/o un puntaje de valoración que se le ha dado a dichos productos) así como también de decisiones similares tomadas por otros usuarios; luego, se hace uso de dicho modelo con el fin de predecir ítems (o valoraciones de los mismos) que pueden ser de interés para el usuario.

Por otro lado, los enfoques de filtrado basado en contenidos utilizan una serie de características discretas de un ítem en particular con el fin de recomendar ítems adicionales con propiedades similares.

Dada la naturaleza de este proyecto, donde la recomendación de empleados se realiza en base tanto a sus características, como a su buen o mal desempeño en ejercicios anteriores, resulta conveniente para este caso adoptar un método donde se combinan ambas estrategias de filtrado, resultando en un enfoque híbrido para obtener el mejor provecho.

2.1.1) FILTRADO BASADO EN CONTENIDO

Los métodos de filtrado basado en contenido se fundamentan en una descripción del ítem en cuestión y en un perfil de las preferencias del usuario. En los sistemas de recomendación basados en contenidos, se utilizan palabras claves para describir los ítems que se someten al proceso de recomendación; además, se arma un perfil de usuario para indicar el tipo de ítem que dicho usuario prefiere.

Para poder abstraer las características de los ítems se debe utilizar algún algoritmo de presentación, mientras que para la creación de los perfiles de usuario, puede optarse por utilizar

un modelo de sus preferencias o un historial de las interacciones del usuario con el sistema de recomendación.

Básicamente, estos métodos utilizan un perfil del ítem (compuesto por un conjunto discreto de atributos) que caracterizan dicho ítem dentro del sistema. El sistema crea un perfil basado en contenido de los usuarios implementado en un vector ponderado de características de los ítems. Estas ponderaciones denotan la importancia que le da el usuario a cada característica. Los enfoques más simples usan valores promedio de vector de ítems clasificado, mientras que otros métodos más sofisticados utilizan técnicas de aprendizaje automático como por ejemplo inferencia Bayesiana, algoritmos de agrupamiento, árboles de decisión y redes neuronales artificiales para poder estimar la probabilidad de que a un usuario le guste un ítem en particular.

2.1.2) FILTRADO COLABORATIVO

Otro enfoque de diseño de sistemas de recomendación ampliamente usado es el filtrado colaborativo (Breese, Heckerman, Kadie, 1998).

Los métodos de filtrado colaborativo se basan en la colección y análisis de grandes cantidades de información relacionada al comportamiento de los usuarios, sus actividades o preferencias y en predecir lo que le gusta al usuario basándose en su similitud con otros usuarios. Una principal ventaja del filtrado colaborativo es que no depende de análisis automático de contenido y por lo tanto es capaz de recomendar en forma precisa ítems complejos sin necesidad de conocer y entender características propias del ítem. Sin embargo, los enfoques de filtrado colaborativo deben lidiar con tres problemas:

- Arranque en frío: Estos sistemas normalmente requieren grandes cantidades de datos existentes sobre un usuario para poder hacer recomendaciones acertadas.
- Escalabilidad: En muchos de los entornos que estos sistemas hacen recomendaciones, hay millones de usuarios y productos. Por lo tanto, se requiere mucha capacidad de procesamiento para calcular las recomendaciones.
- Escasez: En el caso de sistemas e-commerce, existe una gran cantidad de ítems a la venta. La mayoría de los usuarios alcanzan a puntuar un subconjunto muy reducido de la base de datos en su totalidad. Por lo tanto, hasta los ítems más populares suelen tener pocas valoraciones.

2.2) TECNOLOGIAS DE LA WEB SEMÁNTICA

La Web Semántica es una Web extendida, dotada de mayor significado en la que cualquier usuario en Internet podrá encontrar respuestas a sus preguntas de forma más rápida y sencilla gracias a una información mejor definida. Esta Web extendida y basada en el significado, se apoya en lenguajes universales que resuelven los problemas ocasionados por una Web carente de semántica en la que, muchas veces, el acceso a la información se convierte en una tarea difícil y frustrante.

La Web tiene la capacidad de construir una base de conocimiento sobre las preferencias de los usuarios y que, a través de una combinación entre su capacidad de conocimiento y la información disponible en Internet, sea capaz de atender de forma exacta las demandas de información por parte de los usuarios.

La forma en la que se procesará esta información no sólo será en términos de entrada y salida de parámetros sino en términos de su semántica. La Web Semántica como infraestructura basada en metadatos aporta un camino para razonar en la Web, extendiendo así sus capacidades. Las ontologías son consideradas unos de los pilares de la Web Semántica. Una definición de ontología concreta la ofrece Weigand (1997):

Una ontología es una base de conocimiento que describe conceptos del mundo o de algún dominio en particular, alguna de sus propiedades y cómo los conceptos se relacionan entre ellos.

Según la definición de Gruber (1993:199), una ontología constituye "*una especificación formal y explícita de una conceptualización compartida*". En esta definición, que ya se utiliza como un estándar, cuando habla de conceptualización se refiere a un modelo abstracto de algún fenómeno del mundo del que se identifican los conceptos que son relevantes; el término explícito hace referencia a la necesidad de especificar de forma consciente los distintos conceptos que conforman una ontología, el término formal indica que la especificación debe representarse por medio de un lenguaje de representación formalizado. El término compartida refleja que una ontología debe, en el mejor de los casos, dar cuenta de conocimiento aceptado.

Según Gruber, las ontologías se componen de:

- **Conceptos:** son las ideas básicas que se intentan formalizar. Los conceptos pueden ser clases de objetos, métodos, planes, estrategias, procesos de razonamiento, etc.
- **Relaciones:** representan la interacción y enlace entre los conceptos de un dominio. Suelen formar la taxonomía del dominio. Por ejemplo: subclase-de, parte-de, parte-exhaustiva-de, conectado-a, etc.
- **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, pueden aparecer funciones como: asignar-fecha, categorizar-clase, etc.
- **Instancias:** se utilizan para representar objetos determinados de un concepto.
- **Reglas de restricción o axiomas:** son teoremas que se declaran sobre relaciones que deben cumplir los elementos de la ontología. Por ejemplo: "Si A y B son de la clase C, entonces A no es subclase de B", "Para todo A que cumpla la condición B1, A es C", etc. Los axiomas, junto con la herencia de conceptos, permiten inferir conocimiento que no esté indicado explícitamente en la taxonomía de conceptos.

2.2.1) LENGUAJE OWL

OWL (Ontology Web Language) es un lenguaje de marcado (incorpora etiquetas) para publicar y compartir datos usando ontologías en la World Wide Web. OWL tiene como objetivo facilitar un modelo de marcado construido sobre RDF y codificado en XML. El modelo de datos RDF es similar a los enfoques de modelado conceptual clásicos como entidad-relación o diagramas de clases, ya que se basa en la idea de hacer declaraciones sobre los recursos (en particular, recursos web) en forma de tripletas compuestas por sujeto-predicado-objeto. El sujeto indica el recurso y el predicado denota rasgos o aspectos del recurso y expresa una relación entre el sujeto y el objeto.

Las Ontologías OWL son conjuntos de axiomas que proveen afirmaciones sobre tres tipos de elementos: clases, individuos y propiedades. Utilizando una herramienta de software llamada **razonador**, se puede inferir nuevo conocimiento.

Existen dos tipos de propiedades en las Ontologías OWL. Por un lado, las **Data Properties**, son relaciones binarias que asocian un individuo (un sujeto) con algún tipo de dato simple como una fecha o una cadena de caracteres. Por otro lado, las **Object Properties** son relaciones binarias que asocian un individuo con otro individuo.

2.2.2) REGLAS SWRL

El lenguaje OWL no tiene la capacidad de expresar todo tipo de relaciones entre los conceptos que forman una ontología, ya que no hay forma de expresar una relación entre individuos con los que otro individuo tiene relaciones.

El poder de expresividad de OWL puede extenderse al agregar reglas SWRL (Semantic Web Rule Language) a una ontología. Las reglas SWRL son parecidas a las reglas de lenguajes como Prolog o DATALOG. De hecho, las reglas SWRL son reglas DATALOG con predicados unarios para describir clases y tipos de datos, predicados binarios para las propiedades y algunos predicados n-arios especiales incorporados.

El editor OWL de Protegé (editor Open Source de ontologías utilizado en este proyecto) soporta reglas SWRL, así como también los razonadores Pellet y Hermit, de estos hemos utilizado el primero.

2.2.3) CONSULTA DE ONTOLOGIAS: LENGUAJE SPARQL

Para consultar acerca del contenido que abarca una ontología, existen lenguajes de consulta específicos. Entre estos, tenemos a SPARQL (SPARQL Protocol and RDF Query Language), que es un lenguaje estándar para recuperación de información en W3C. La importancia de SPARQL radica en gran parte en su capacidad de expresar sus consultas en un lenguaje de alto nivel sin importar el formato en que estos datos han sido almacenados. Tim Berners Lee (2002) lo explica así:

“Pretender usar la Web Semántica sin SPARQL es como pretender usar una base de datos relacional sin SQL”, “SPARQL hace posible consultar información desde bases de datos y otros orígenes de datos en sus estados primitivos a través de la web”.

La mayoría de las formas de consulta en SPARQL contienen un conjunto de patrones de tripleta denominadas *patrón de grafo básico*. Los patrones de tripleta son similares a las tripletas RDF, excepto que cada sujeto, predicado y objeto puede ser una variable. Un patrón de grafo básico concuerda con un subgrafo de datos RDF cuando los términos RDF de dicho subgrafo pueden ser sustituidos por las variables y el resultado es un grafo RDF equivalente al subgrafo en cuestión.

Utilizar el lenguaje de consultas SPARQL, nos permite realizar consultas orientadas puramente a contenidos semánticos, y no basándonos en la teoría de conjuntos como el modelo relacional, el cual no tiene capacidades para representar la ontología de manera óptima.

Para realizar modificaciones de tipo inserción, eliminación y actualización (equivalentes a INSERT, DELETE, UPDATE de SQL) se utiliza una extensión de SPARQL denominada SPARUL (SPARQL/Update) creada por Hewlett-Packard.

2.3) METODOLOGÍA DE DESARROLLO DE ONTOLOGÍAS: EDON

Para el desarrollo de la ontología que se utilizó en este proyecto se aplicó la metodología EDON (Reynares y otros., 2012). La elección de esta metodología se basa en que, al haber sido desarrollada en el CIDISI de la FRSF, esperamos contribuir al refinamiento de la misma aplicándola y determinando, a nuestro criterio, sus ventajas y desventajas como han hecho otros proyectos de carrera de temática similar.

En esta metodología se trabaja conjuntamente con personas que cumplen los siguientes roles:

Expertos del Dominio: Son aquellos que están en constante contacto con el dominio del problema por lo que conocen el alcance y saben qué necesidades deben ser modeladas. Para este proyecto de asignación de recursos humanos a eventos se habló con personal de la empresa Gut Cátering con conocimiento acerca del dominio y que son las encargadas de realizar la asignación del personal a eventos en forma manual hoy en día.

Ingenieros del Conocimiento: Son aquellos que logran transformar la realidad del dominio en sentencias que puedan ser leídas por algún software.

Estos roles tienen que estar vinculados por canales de comunicación activos y dinámicos que permitan mantener la ontología actualizada, con los datos necesarios y mejorarla con cada iteración.

A EDON podríamos ubicarla entre la educación de requerimientos con el cliente y el comienzo del desarrollo del software, como vemos en la figura 5.1:

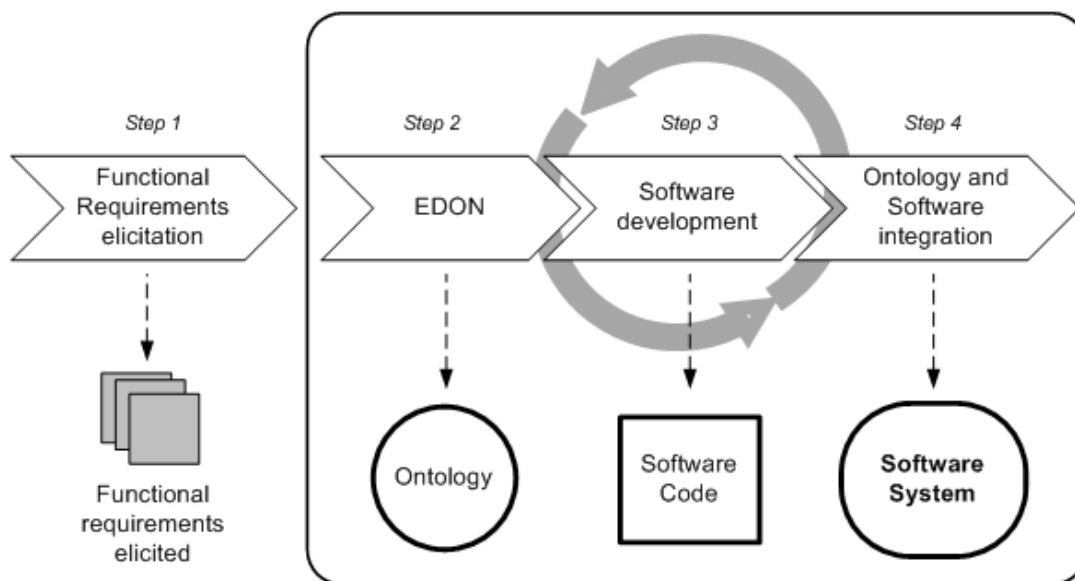


Figura 5.1 EDON en el desarrollo de software.

Una vez concluida la etapa de educación de los requerimientos comienza la etapa donde aplicamos la metodología EDON para comenzar a definir la ontología que es el marco conceptual del sistema desarrollado. Esta metodología consiste de una secuencia de etapas que se muestran en la figura 5.2:

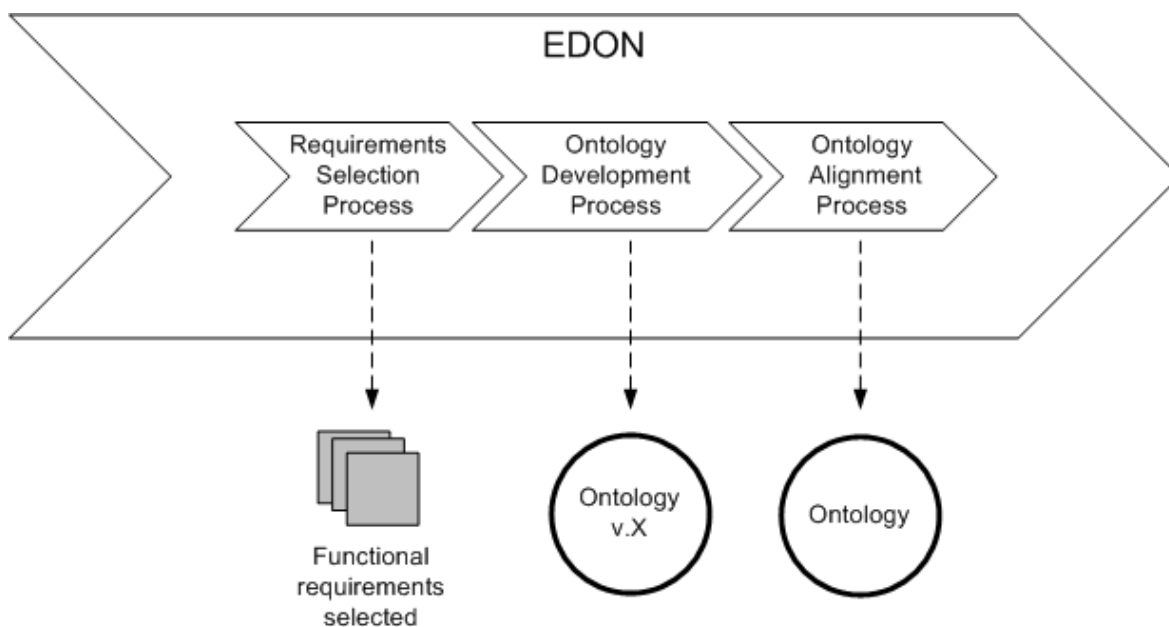


Figura 5.2 Etapas de EDON.

En primer lugar se lleva a cabo el Proceso de Selección de Requerimientos que consiste en determinar cuáles de los requerimientos funcionales del sistema serán soportados por la ontología teniendo en cuenta que:

- La ontología debe poder razonar a partir de reglas de negocio.
- El usuario no interactúa directamente con la ontología sino que lo hará mediante la aplicación de software.

Luego de la selección de requerimientos se lleva a cabo el Proceso de Desarrollo de la Ontología que abarca actividades que van desde la especificación de un modelo abstracto hasta una ontología específica que pueda ser entendida por un software.

Cada aplicación de EDON produce una nueva Ontología que soporta un conjunto inconexo de los requisitos funcionales, aquellos seleccionados en la especificación de la iteración. La alineación de la versión actual con la anterior es necesaria para apoyar el conjunto de requerimientos. La alineación de la Ontología, es el proceso de identificar diferentes tipos de relaciones entre los términos. Luego se realiza la actividad de evaluación de la alineación, para ver

la calidad. Si la calidad necesaria no es alcanzada las relaciones establecidas en la actividad de alineación deben ser formuladas y refinadas.

CAPÍTULO 3: ANÁLISIS DEL DOMINIO Y REQUERIMIENTOS

El alcance de este sistema abarca hasta las funcionalidades necesarias para poder llevar a cabo la asignación de recursos humanos de la empresa de organización de eventos GUT catering. Estas funcionalidades consisten en la gestión de recursos humanos, la gestión de los eventos y la asignación que asocie ambas partes.

3.1) LOS EVENTOS

Los servicios ofrecidos por la empresa Gut Catering consisten en organizar eventos de distinto tipo. Para registrar un evento se debe indicar el tipo de evento, la fecha y hora del mismo, la duración aproximada, la cantidad de asistentes y nombre de cliente. Estos eventos pueden estar compuestos de hasta tres etapas. Las etapas que puede incluir un evento son: (i) la recepción , (ii) servicio de mesa y (iii) trasnoche. No necesariamente el evento debe incluir todas las etapas, pero en caso de que se incluyan más de una, deben desarrollarse en el orden mencionado.

Los eventos a su vez están compuestos por una serie de actividades donde cada una requiere uno o más empleados que puedan desempeñar algún rol en particular y además pueden o no necesitar que los empleados que participen de dicha actividad tengan determinadas habilidades y que se adecuen a un perfil en caso de preferir personas de un sexo o edad para el desarrollo de una actividad.

Las actividades son la fuente de donde un organizador de eventos determina los requisitos en base a los que se recomendará un empleado para ejecutar la misma. Para este caso, los eventos y sus etapas son análogos a los usuarios de sistemas recomendadores que mencionamos en el marco teórico, y los empleados son los ítems a recomendar. Mientras que para un sistema recomendador tradicional se recomienda algún tipo de ítem de acuerdo al perfil de un usuario, en este sistema se recomiendan empleados de acuerdo a los roles y habilidades que se configuren como necesarios para una actividad que compone un evento, haciendo uso de un enfoque híbrido que ordene a los empleados según sus características (filtrado basado en contenidos) así como también del puntaje obtenido por sus desempeños en eventos en los que hayan participado anteriormente (filtrado colaborativo). Los desempeños deberán poder ser cargados una vez finalizado el evento, exponiendo a un supervisor todos los empleados que participaron del mismo.

3.2) LOS EMPLEADOS

Los empleados son el recurso humano de la empresa Gut Catering. En su gran mayoría son temporarios, esto quiere decir que los empleados no tienen una relación de dependencia duradera con la empresa sino que se rigen por contratos para trabajar en los eventos para los que fueron seleccionados. En esta temporalidad es donde radica la mayor dificultad para seleccionar su participación en eventos. Hay una gran disponibilidad de personas dispuestas a trabajar, sin embargo todas ellas tienen restricciones de días y horarios a la vez que se suman las restricciones en cuanto a las habilidades necesarias para realizar el trabajo, todo esto hace que seleccionar un empleado no sea una tarea fácil.

Los empleados deben ser capaces de desempeñar un rol y además pueden poseer determinadas habilidades que les serán útiles para ser recomendados en eventos donde se necesite de ellos.

Actualmente la empresa cuenta con información de sus empleados en planillas. Esta información consiste en el legajo, apellido, nombres, DNI, CUIL, fecha de nacimiento, dirección, teléfono, mail y habilidades que cumplen (actualmente se registran las siguientes habilidades: carnet de conducir, coctelería, cafetería, islas, bocados, emplatado y cocina).

Además de ser recomendados en eventos a partir de sus roles y habilidades, se tendrá en cuenta si el empleado ya ha participado de eventos realizando actividades similares o si es nuevo para llevar a cabo determinadas tareas, así también como su desempeño en caso de haber trabajado en eventos anteriores. Los empleados pueden estar relacionados entre ellos, como ser por ejemplo, estar en pareja, tener afinidades, tener relaciones de parentesco, tener problemas personales entre ellos, etc. Estas relaciones se pueden conocer a priori o bien descubrirse durante la ejecución de un evento y el supervisor determinará si es necesaria tenerla en cuenta a la hora de asignar los recursos ya sea por considerarla beneficiosas o malignas. Estas relaciones deben ser tenidas en cuenta a la hora de asignar los empleados a un evento con el fin de evitar situaciones conflictivas durante el desarrollo del mismo.

3.3) REQUERIMIENTOS FUNCIONALES DEL SISTEMA

Para poder realizar la captura de los requerimientos del sistema de información se llevaron a cabo entrevistas con las personas responsables de la gestión de eventos de Gut Catering con el fin de que nos planteen sus necesidades y expectativas de los posibles usuarios. Para la especificación de los requerimientos funcionales que serán soportados por el sistema hemos definido los casos de uso correspondientes los cuales se muestran en la figura 3.1.

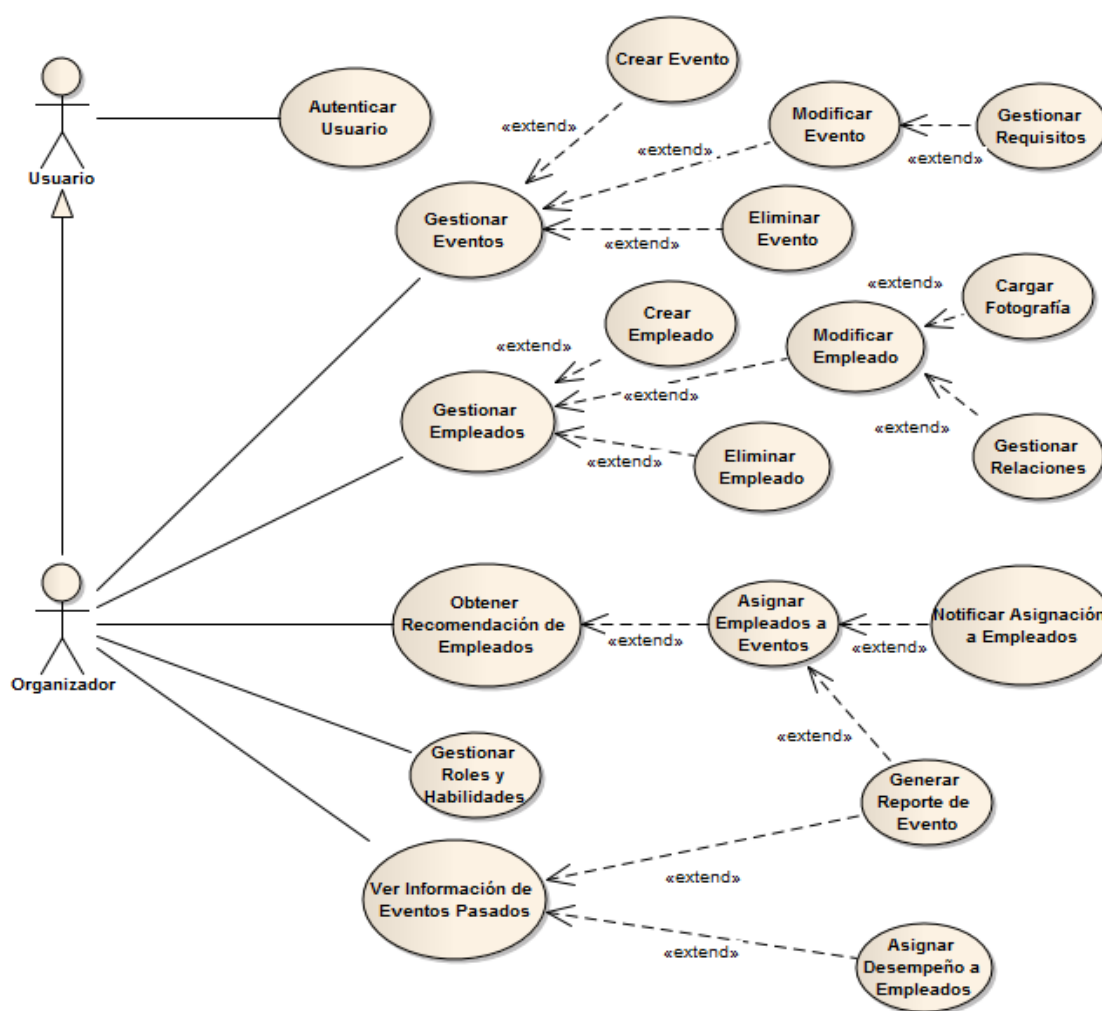


Figura 3.1: Casos de uso del sistema.

A continuación describiremos brevemente los casos de uso y los actores.

Actores:

El actor usuario representa el rol que cumple cualquier persona designada para utilizar el sistema. El usuario puede, por un lado, ser un administrador, que es aquel encargado de gestionar a los demás usuarios del sistema, sus nombres de usuario y claves de ingreso así como su tipo de acceso. Por otro lado un usuario puede ser un organizador, que es aquel encargado de gestionar toda la información relacionada a la lógica de negocios de la empresa, los eventos, los empleados, obtener recomendaciones y realizar asignaciones.

Casos de Uso:

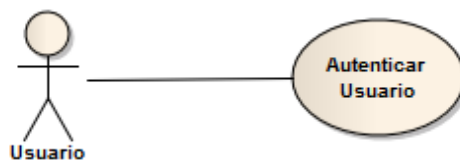


Figura 3.2: Caso de uso Autenticar Usuario.

Autenticar Usuario: es el primer caso de uso que se ejecuta siempre al comenzar a utilizar el sistema. El usuario recibe una indicación para ingresar su nombre de usuario y contraseña para poder autenticarse como usuario del sistema. Una vez ingresado sus datos, el sistema verificará los mismos, dando acceso, en caso de ser válidos, a las funcionalidades correspondientes según el tipo de usuario que está ingresando.

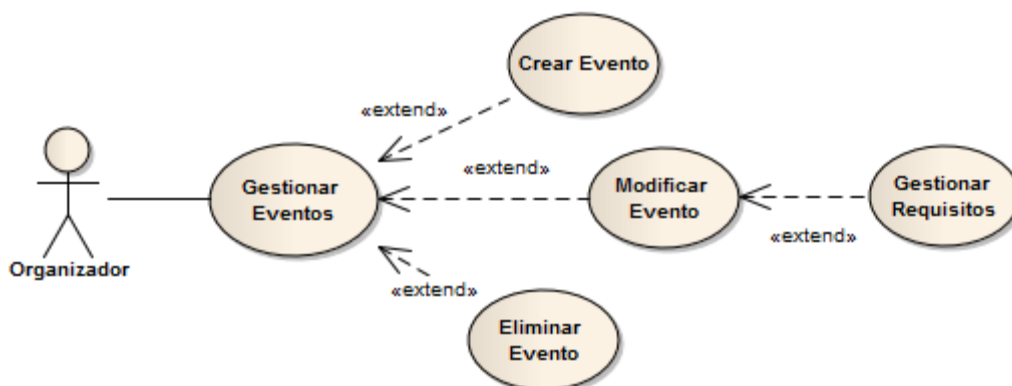


Figura 3.3: Caso de uso Gestionar Eventos.

Gestionar Eventos: este caso de uso representa una de las funcionalidades accesible por el organizador. Consiste en registrar en el sistema los eventos que organiza la empresa, modificarlos o eliminarlos. Se configuran todos los aspectos relacionados al evento, como la fecha, las etapas que lo componen, las actividades que se desarrollarán, etc. Para los eventos creados se podrá acceder a la funcionalidad de gestionar los requisitos del mismo, indicando para qué rol se deberá tener en cuenta cada uno. Por cada requisito se podrá incluir un rango de edad, sexo y una habilidad deseados.

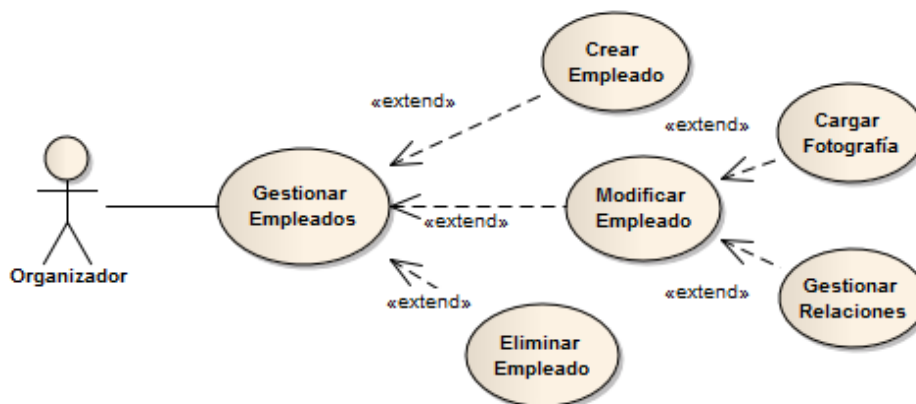


Figura 3.4: Caso de uso Gestionar Empleados.

Gestionar Empleados: este caso de uso corresponde a otra de las funcionalidades a las que puede acceder un organizador. Consiste en realizar tareas correspondientes al alta, baja y modificación

de empleados de la empresa. Los datos que se manejan en este caso son principalmente para identificar a los empleados dentro del sistema así como el rol que desempeñan y las habilidades que poseen. De manera similar a como se asocia un empleado a un determinado rol y habilidades, el caso de uso de gestionar relaciones corresponde a la funcionalidad de asociar empleados entre ellos, describiendo la razón de dicha asociación como puede ser la conformación de una pareja, o una relación de enemistad, extendiendo la funcionalidad de modificar los datos básicos del empleado. Para los empleados del sistema también debe haber una funcionalidad que permite cargar una fotografía representativa de ellos.

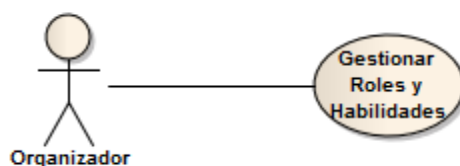


Figura 3.5: Caso de uso Gestionar Roles y Habilidades.

Gestionar Roles y Habilidades: este caso de uso abarca la funcionalidad que consiste en crear o modificar todos los roles posibles que pueden desempeñar los empleados y asociarlos con habilidades que son propias de cada rol.

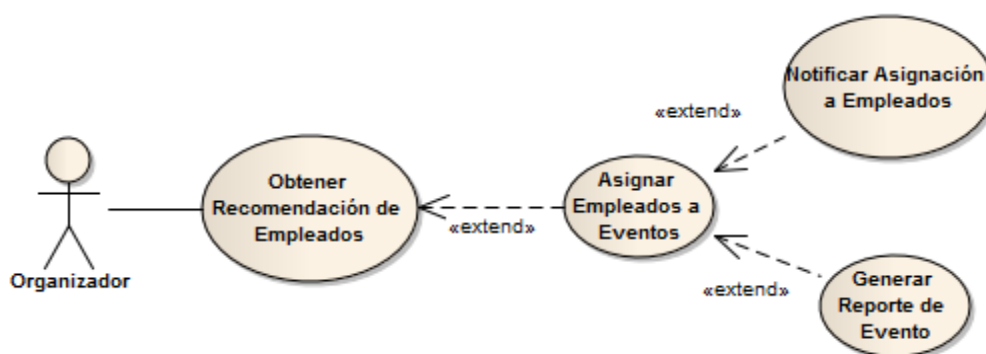


Figura 3.6: Caso de uso Obtener Recomendación de Empleados.

Obtener Recomendación de Empleados: esta funcionalidad es la que se corresponde con el proceso de asignación de empleados a eventos. A partir de un evento próximo seleccionado, se

obtendrá una lista de empleados recomendados ordenados de mejor a peor para desempeñar un rol seleccionado. A partir de la recomendación obtenida, el organizador puede realizar la asignación efectiva de los empleados al evento indicando un rango de horario que puede o no durar por la totalidad del evento. Una vez confirmada la asignación para el evento se puede notificar a los empleados que quedaron asignados vía e-mail inmediatamente y que estos puedan ponerse en contacto en caso de no poder asistir. En dicho caso debe ser posible alguna modificación en la asignación. Una vez realizada la asignación de empleados debe ser posible generar un reporte de los empleados que participarán del evento ya que suelen utilizarse dichas planillas durante el evento para llevar un registro de los empleados que están trabajando en ese momento.



Figura 3.7: Caso de uso Ver Información de Eventos Pasados.

Ver Información de Eventos Pasados: durante esta funcionalidad se puede observar la información relacionada a eventos ya ocurridos. Se podrá ver los empleados que trabajaron durante el evento, en que etapa y la actividad que realizaron. El organizador tendrá la opción de calificar el desempeño de los empleados por su trabajo.

Además durante esta funcionalidad se debe permitir la generación de un reporte en forma de planilla que detalle los empleados que trabajaron del evento indicando el rango horario y el rol que desempeñaron durante este.

CAPÍTULO 4: DISEÑO Y ARQUITECTURA

4.1) ARQUITECTURA

Para el desarrollo del módulo de recursos humano, decidimos llevarlo a cabo en forma de una aplicación web. A partir de la necesidad del cliente de necesitar acceder a la aplicación en forma remota, nos pareció oportuno esta alternativa ya que la aplicación estará al alcance de cualquier usuario siempre y cuando cuente con un dispositivo que tenga un navegador web, el cual puede hallarse en cualquier computadora ya sea de escritorio o portátil así como también en dispositivos móviles como celulares, tablets, etc. De hecho, Las aplicaciones web son populares hoy en día dada la ubicuidad de los navegadores web. El dispositivo deberá contar con una conexión a internet, que puede ser inalámbrica haciendo uso de tecnologías de telecomunicaciones usadas actualmente como Wi-Fi, 3G, 4G, etc.

Dada su infraestructura, las aplicaciones web aportan las siguientes ventajas:

- **Ahorra tiempo:** Se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa.
- **No hay problemas de compatibilidad:** Basta tener un navegador actualizado para poder utilizarlas.
- **No ocupan espacio** en disco duro.
- **Actualizaciones inmediatas:** Como el software lo gestiona el propio desarrollador, cuando nos conectamos estamos usando siempre la última versión que haya lanzado.
- **Consumo de recursos bajo:** Dado que toda la aplicación (en nuestro caso) no se encuentra en el ordenador desde el cual se accede, muchas de las tareas que realiza el software no consumen recursos de dicho cliente porque se realizan desde otro ordenador (servidor).
- **Multiplataforma:** Se pueden usar desde cualquier sistema operativo porque sólo es necesario tener un navegador.
- **Portables:** Es independiente del ordenador donde se utilice (una PC de escritorio, portátil, etc) porque se accede a través de una página web (sólo es necesario disponer de acceso a Internet).

- **La disponibilidad suele ser alta** porque el servicio puede ofrecerse desde múltiples localizaciones para asegurar la continuidad del mismo.
- **Los virus no dañan** los datos porque éstos están guardados en el servidor de la aplicación. El servidor debe contar con la seguridad adecuada.
- **Colaboración:** Gracias a que el acceso al servicio se realiza desde una única ubicación es sencillo el acceso y compartición de datos por parte de varios usuarios.

El uso de aplicaciones web también trae una serie de inconvenientes que se deben tener en cuenta que son:

- Habitualmente ofrecen **menos funcionalidades que las aplicaciones de escritorio**. Se debe a que las funcionalidades que se pueden realizar desde un navegador son más limitadas que las que se pueden realizar desde el sistema operativo. Para este caso tuvimos en cuenta que para el alcance del módulo que desarrollamos para este proyecto nos bastaba con las funcionalidades que pueden ofrecer las aplicaciones web hoy en día.
- **La disponibilidad depende de un tercero**, el proveedor de la conexión a internet o el que provee el enlace entre el servidor de la aplicación y el cliente. Así que la disponibilidad del servicio está supeditada al proveedor.

Para el desarrollo de este sistema se optó por utilizar una arquitectura de división por capas. En la ingeniería de software, la arquitectura de división por capas es una arquitectura cliente-servidor en donde la presentación, el procesamiento, y el manejo de datos se encuentran lógicamente separados. El modelo más utilizado, sobre todo en aplicaciones web, es el de tres capas, que es un patrón de arquitectura de software que se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento. Esta arquitectura nos provee un modelo en el que se puede desarrollar una aplicación flexible y reutilizable. Al separar la aplicación en capas, existe la posibilidad de modificar capas existentes o agregar nuevas capas sin necesidad de modificar toda la aplicación. Una arquitectura de tres capas está compuesta tradicionalmente por la capa de presentación, la capa de lógica de negocio, y la capa de almacenamiento de datos.

Particularmente para nuestra aplicación se ha utilizado una extensión del modelo de tres capas, incorporando una nueva capa llamada capa de razonamiento, donde se encuentra la ontología y donde por ende se realiza el procesamiento de inferencia. Esta capa es alternativa a la capa de almacenamiento de datos, ya que se encuentran a la misma altura interactuando ambas con la lógica de negocio. Finalmente para nuestra arquitectura de cuatro capas quedan definidas:

Capa de Presentación: Presenta la información y la lógica de negocio como salida, en un formato adecuado para interactuar con el usuario. Para construir las interfaces de usuario se optó por utilizar Vaadin. Vaadin es un framework para el desarrollo de aplicaciones web Open Source modernas o también llamadas RIA (Rich Internet Applications) que posee una librería con numerosos componentes de usuario con los que se puede llegar a construir aplicaciones web profesionales similares a las interfaces de una aplicación de escritorio. Este framework se ejecuta del lado del servidor lo que significa que la mayoría de la lógica y por tanto la mayor carga del trabajo recae en el servidor.

Vaadin utiliza tecnología Ajax del lado del cliente cuya base es GWT (Google Web Toolkit) para renderizar la interfaz de usuario en el navegador. Las aplicaciones Google Web Toolkit se escriben en Java y son automáticamente traducidas a JavaScript y HTML. En la figura 4.1 se muestra la separación de componentes básicos entre el cliente y servidor.



Figura 4.1 Separación de lados Cliente-Servidor.

Capa de Lógica de Negocio: En esta capa es donde residen la aplicación que se ejecuta, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, con la capa de datos, para solicitar al gestor de base de datos o al almacenar o recuperar datos de él, y con la capa de razonamiento para recuperar resultados de una inferencia realizada por el razonador. Para este sistema la capa de lógica se refleja por medio de clases del modelo del dominio que representan las entidades principales y clases de servicio que se invocaran desde las interfaces mediante eventos que pueden ser disparados desde determinados componentes de la interfaz gráfica (por ejemplo presionar un botón, seleccionar una fila de una tabla, etc), que indican las acciones del usuario. En la figura 4.2 se muestra la ubicación de los componentes que incluye la capa de lógica de negocio.



Figura 4.2 Componentes de la Lógica de Negocio.

Capa de Almacenamiento de Datos: Aquí se encuentra toda la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, mediante mecanismos de persistencia. Para separar esta capa de la lógica de negocios y que aun así puedan interactuar utilizamos un ORM (object-relational mapping) para dar soporte, principalmente a las funcionalidades de inserción, eliminación y actualizaciones de datos del sistema (empleados, eventos, etc.) que no establezca dependencias con los mecanismos de almacenamiento de datos. Elegimos el framework Hibernate, que provee mecanismos de mapeo de un modelo de clases a un modelo relacional de tablas mediante la incorporación de metadatos que indiquen cómo llevarlo a cabo. Además provee herramientas de consulta orientadas a objetos que nos permiten consultar una base de datos en forma transparente como si sólo estuviéramos trabajando con objetos sin tener en cuenta la estructura del modelo relacional. En la figura 4.3 se muestra como hibernate se ubica como intermediario entre una aplicación y los datos.

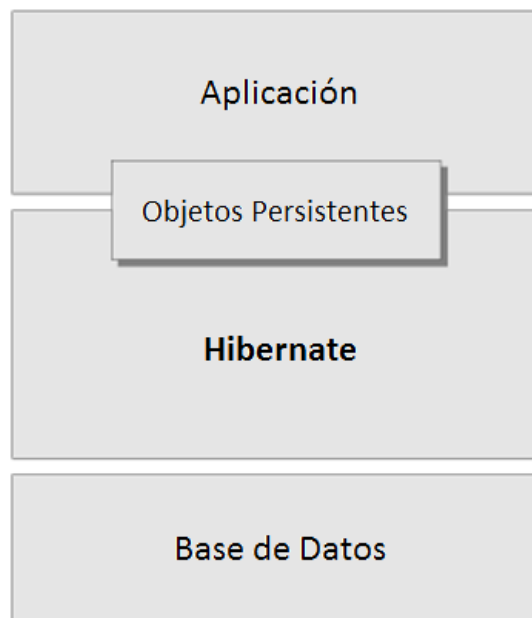


Figura 4.3 Componentes de interacción entre la aplicación y datos mediante un ORM.

Capa de Razonamiento: en esta capa se ubicará una ontología para dar soporte a funcionalidades que requieren inferencias lógicas a partir de datos conocidos. Para interactuar con la ontología se usará el framework Jena. Jena es un framework de web semántica de código abierto para java. Provee una API para extraer e introducir datos en grafos RDF que se encuentra representados como un modelo abstracto. La fuente de datos de un modelo puede provenir de archivos, bases de datos, URLs o una combinación de estos. Un modelo puede ser consultado mediante SPARQL y actualizado mediante SPARUL.

Jena da soporte para OWL. Además posee varios razonadores internos. Particularmente el razonador Pellet (un razonador OWL-DL de java de código abierto), utilizado en nuestra aplicación, puede incorporarse para usarse con Jena. En la figura 4.4 se muestra como Jena actúa de intermediario entre la aplicación y la ontología utilizando un razonador.

Para la construcción de la ontología con la que interactúa la aplicación se utilizó la metodología EDON introducida en el capítulo 2. En el capítulo 5 se explican los pasos de la metodología que se fueron siguiendo para el desarrollo de la ontología.

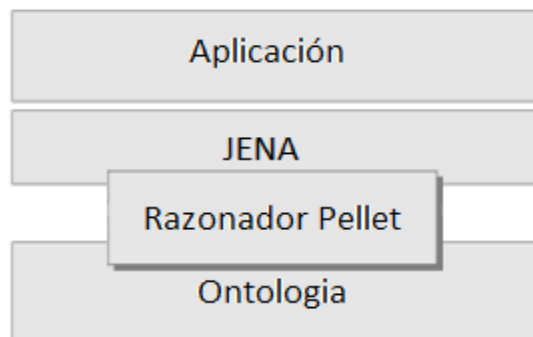


Figura 4.4 Componentes de interacción entre la aplicación y ontología mediante un razonador.

El paradigma utilizado para el desarrollo fue el de programación orientada a objetos, por lo que la aplicación en si estará compuesta por un conjunto de clases. Por un lado debemos contar con las clases que representen el modelo de dominio del sistema (Empleado, Evento, Recomendación, etc.) para instanciar los objetos que se manipulen y reconozcan solo por dentro de la aplicación, por otro las clases que constituyen cada una de las capas que se definieron en la arquitectura (Presentación, Servicios, Acceso a Datos).

4.2) ESTRUCTURA: CLASES DEL MODELO DE DOMINIO

Para describir parte de la estructura estática del sistema, confeccionamos un diagrama de clases UML. En la figura 4.5 se muestran las clases del sistema, sus atributos y las relaciones entre ellas. Este diagrama de clases abarca los elementos del modelo de dominio representado dentro de la propia aplicación y que solo se manejan internamente. El modelo de datos también está relacionado al modelo de objetos para este proyecto ya que se utilizó, como explicamos anteriormente, un ORM que construye el modelo de datos a partir de las clases de nuestra aplicación. El modelo de la ontología se construye en base a términos y procedimientos que se explican en el capítulo 5, en la aplicación de la metodología EDON para desarrollar ontologías.

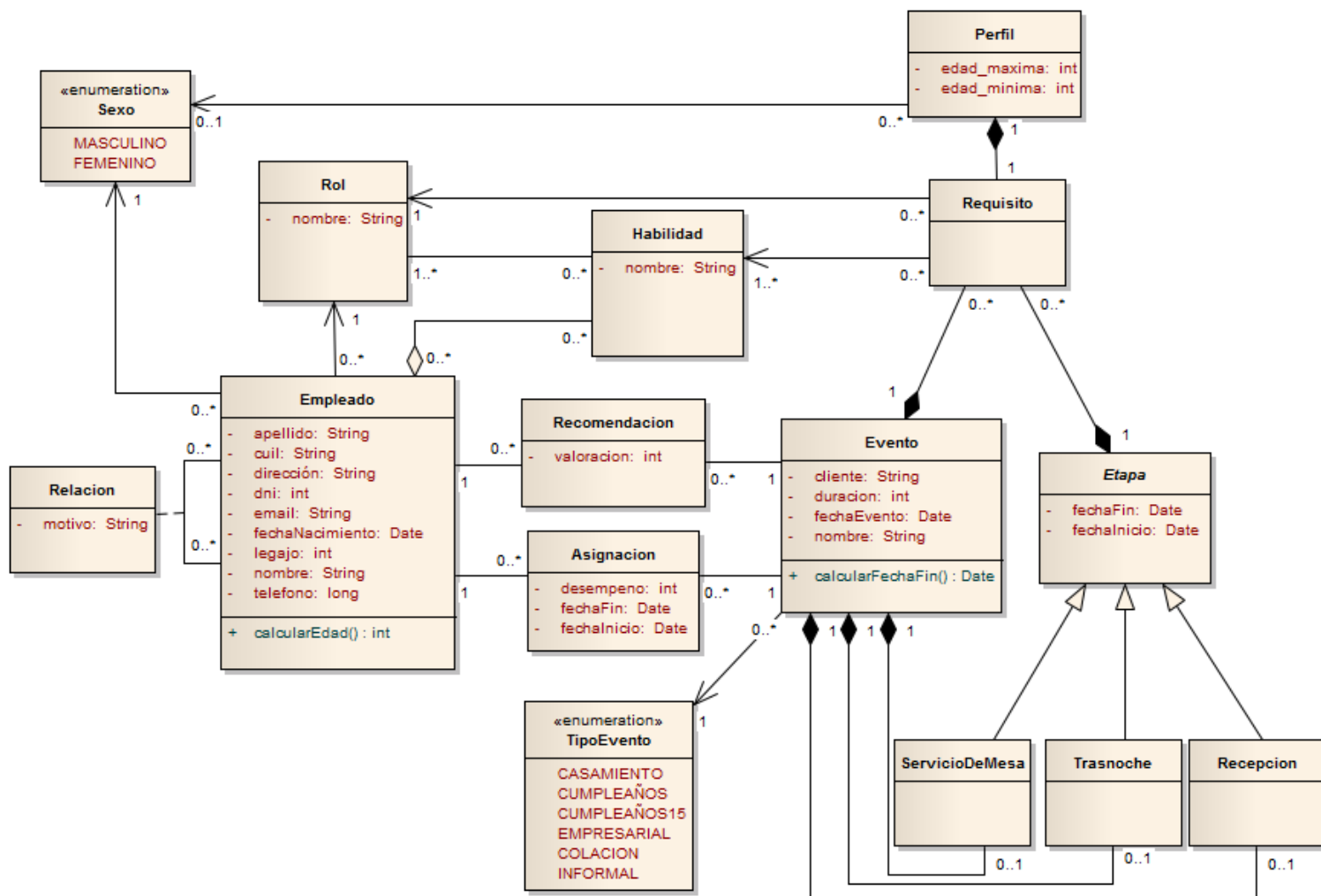


Figura 4.5: Diagrama de Clases.

En primer lugar tenemos una clase **Empleado**, cuyos atributos están determinados principalmente a partir de la información que conoce la empresa y de la cual guarda registros (apellido, CUIL, legajo, nombre, etc.). Los empleados tienen un **Rol** que desempeñan en su trabajo, y tienen asociado un conjunto de **Habilidades**.

Los Roles y las Habilidades están representadas por su nombre.

La clase **Relación** asocia dos instancias de Empleado y representa la existencia de una relación conflictiva entre los mismos con fines de evitar la convivencia de ambos en un mismo evento a la hora de llevar a cabo las recomendaciones de empleados a eventos. Tienen como

atributo el motivo de tal relación, para describir la magnitud del conflicto y que el usuario decida si es un factor determinante a la hora de llevar a cabo la asignación.

La clase **Evento** tiene como atributos un nombre que lo representa, el tipo de evento (Casamiento, cumpleaños, empresarial, etc.), una fecha, la duración aproximada en horas y el nombre del cliente. El Evento puede estar asociado a una **Etapa de Servicio de Mesa**, a una Etapa de **Trasnoche** y/o a una Etapa de **Recepción**. Cada Evento puede tener asociado un conjunto de **Requisitos** que son los que se tendrán en cuenta a la hora de realizar una Recomendación de empleados que deberían participar del mismo.

Las instancias de la clase **Recomendación** no se persisten. Es decir, no tienen registros asociados a ellas en la base de datos ya que son objetos que representan las inferencias de la ontología y se generan al momento en que se accede a la funcionalidad de asignar empleados a eventos. Las Recomendaciones asocian un Empleado, un Evento y tienen un atributo valoración que es calculado por inferencia de la ontología para determinar qué tan apto es dicho Empleado para participar del Evento.

Las **Asignaciones** se asemejan a las Recomendaciones en que también asocian un Empleado a un Evento. Sin embargo, a diferencia de estas últimas, las instancias de Asignación si son persistidas en la base de datos ya que representan el hecho de asociar definitivamente el Empleado al Evento y debe quedar registrado para usos que van más allá de la funcionalidad de la recomendación y asignación (por ejemplo, para calificar el desempeño del trabajo del empleado en el evento al que fue asignado una vez que el mismo haya sido concretado). Las Asignaciones tienen como atributo además la fecha inicial y final de la misma para determinar la cantidad de horas de trabajo de un empleado en un Evento.

La clase **Requisito** se utiliza para instanciar características deseables de los empleados que participen de un evento. Asocian para un Rol, un conjunto de Habilidades deseadas y también un Perfil que consiste de un rango de edades y un **Sexo**. Luego la ontología se encargará de determinar correspondencias entre las características del empleado y características requeridas por el requisito y determinar una valoración de la recomendación resultante.

4.3) ESTRUCTURA: CLASES DE COMPONENTES DE CAPAS DE LA APLICACIÓN

En la figura 4.6 se muestra como se asocian las clases de interfaces gráficas con las clases de servicios y estas clases de servicios con las clases de objetos de acceso a datos (DAO) delimitando las tres capas impuestas por la arquitectura utilizada:

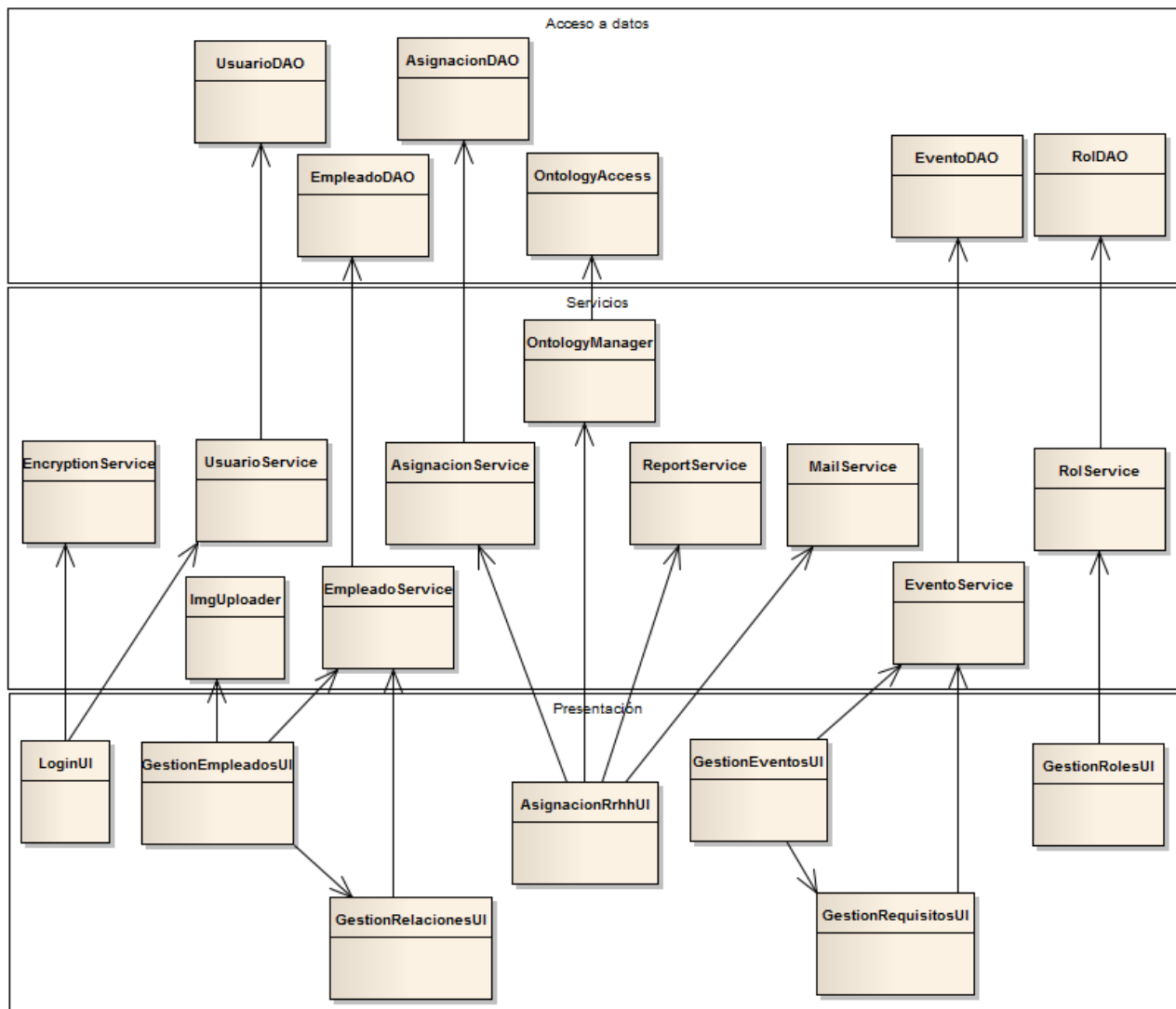


Figura 4.6 Diagrama de clases de servicios por capas.

Clases de Presentación:

- **LoginUI:** clase donde se codifica la Interfaz gráfica de usuario inicial de la aplicación donde se encuentran los campos para ingresar los datos correspondientes al nombre de usuario y contraseña. Dicha interfaz gráfica tiene referencias a `UsuarioService` y a `EncryptionService` que son, respectivamente, para corroborar que el nombre de usuario es válido y que la contraseña ingresada es correcta luego de pasar por el proceso de encriptación y coincida con la que se encuentre almacenada en la base de datos.
- **GestionEmpleadosUI:** clase donde se codifica la Interfaz gráfica de la gestión de empleados. Por un lado, tiene referencia a `EmpleadoService` para realizar operaciones con las instancias de empleados. También posee una referencia al servicio `ImgUploader` que tiene los métodos necesarios para la carga de imágenes y establecer una fotografía de perfil para los empleados. Finalmente se tiene una referencia a la clase de interfaz gráfica `GestionRelacionesUI` para poder gestionar las relaciones de un determinado empleado.
- **GestionRelacionesUI:** clase donde se codifica la Interfaz gráfica de la gestión de relaciones. Esta interfaz tiene como parámetro el empleado de quien se gestionan las relaciones y posee una referencia a `EmpleadoService` para realizar operaciones con instancias de `Empleado` y `Relación`.
- **GestionEventosUI:** clase donde se codifica la Interfaz gráfica de la gestión de eventos. Por un lado, tiene referencia a `EventoService` para realizar operaciones con las instancias de eventos. En esta clase se tiene una referencia a la clase de interfaz gráfica `GestionRequisitosUI` para poder gestionar los requisitos de un determinado evento.
- **GestionRequisitosUI:** clase donde se codifica la Interfaz gráfica de la gestión de requisitos. Esta interfaz tiene como parámetro el evento del cual se gestionan los requisitos y posee una referencia a `EventoService` para realizar operaciones con instancias de `Evento` y `Requisito`.
- **GestionRolesUI:** clase donde se codifica la Interfaz gráfica de la gestión de roles y habilidades. Tiene referencia a `RolService` para realizar operaciones con las instancias de roles y habilidades.
- **AsignacionRrHhUI:** clase donde se codifica la Interfaz gráfica de la asignación de recursos humanos a eventos. En esta clase hay referencias a diversos servicios. En primer lugar, hay

una referencia a `AsignacionService` para realizar operaciones con las instancias de `Asignacion`. Además, hay una referencia a `OntologyManager` para poder realizar operaciones de inferencia y operar sobre instancias de `Recomendacion`. Finalmente se referencian las clases `MailService` y `ReportService`. El servicio `Mail` permite realizar el envío de correo electrónico para llevar a cabo la funcionalidad de notificar a los empleados asignados. `ReportService` permite llevar a cabo la creación de un archivo descargable que consiste en la planilla de datos de los empleados asignados a un evento.

Clases de Servicio:

- **EncryptionService:** clase de servicio que posee métodos para encriptar contraseñas que deben ser guardadas en la base de datos y chequear que las contraseñas ingresadas coincidan con la almacenada. Los métodos principales son `encrypt(String passwordNativa)` que retorna la password encriptada y `check(String passwordIngresada, String passwordEncriptada)` que retorna verdadero o falso dependiendo de si coincide o no el parámetro `passwordIngresada` con `passwordEncriptada`. Las contraseñas se almacenan en forma encriptada mediante una función de derivación de claves PBKDF2 (Password-Based Key Derivation Function 2). Esta función tiene como entradas por un lado la contraseña y por otro lo que se denomina una *sal* (*salt*) que comprende bits aleatorios. Luego aplica a estas entradas una función hash para generar una contraseña encriptada.
- **UsuarioService:** es el servicio que provee métodos para operar con instancias de `Usuario`. Se utiliza principalmente para recuperar la instancia de usuario correspondiente al nombre de usuario que se ingresa en la funcionalidad de autenticación con el fin de corroborar que la contraseña almacenada coincide con la ingresada. Tiene una referencia a `UsuarioDAO` que es donde se encuentran los métodos que interactúan con los datos.
- **EmpleadoService:** este servicio posee métodos para recuperar instancias de los empleados registrados en el sistema así como también los métodos para guardar empleados nuevos o eliminarlos de la base de datos. Posee métodos para trabajar tanto con instancias de `Empleado` como con instancias de `Relación`. Tiene una referencia a `EmpleadoDAO` que es donde se encuentran los métodos que interactúan con los datos.

- **ImgUploader:** este servicio provee el método `upload(String filePath)` que permitirá la carga de una imagen al servidor a partir de un parámetro donde se indique la ruta del sistema de archivos donde se ubica dicho archivo de imagen.
- **EventoService:** este servicio posee métodos para recuperar instancias de los eventos registrados en el sistema así como también los métodos para guardar nuevos eventos o eliminarlos de la base de datos. Posee métodos para trabajar tanto con instancias de Evento como con instancias de Requisito. Tiene una referencia a `EventoDAO` que es donde se encuentran los métodos que interactúan con los datos.
- **RolService:** este servicio posee métodos para recuperar instancias de los roles registrados en el sistema así como también los métodos para guardar nuevos roles o eliminarlos. Posee métodos para trabajar tanto con instancias de Rol como con instancias de Habilidad. Tiene una referencia a `RolDAO` que es donde se encuentran los métodos que interactúan con los datos.
- **AsignacionService:** este servicio posee métodos para recuperar instancias de las asignaciones registradas en el sistema así como también los métodos para guardar nuevas asignaciones o eliminarlas. Tiene una referencia a `AsignacionDAO` que es donde se encuentran los métodos que interactúan con los datos.
- **MailService:** este servicio provee principalmente el método `send(String remitente, String receptor, String asunto, String mensaje)` donde se implemente el envío de un correo electrónico donde figure como remitente, receptor, asunto y mensaje aquellos que sean pasados por parámetro.
- **ReportService:** este servicio provee principalmente un método para poder descargar una planilla donde se visualicen los empleados que trabajaron de un evento en forma de un archivo con extensión `.xls`.
- **OntologyManager:** este servicio provee los métodos para interactuar con la ontología y operar sobre instancias de Recomendación. Tiene una referencia a `Ontology Access` que es donde se encuentran los métodos que interactúan directamente con la ontología.

Clases de Acceso a Datos:

- **UsuarioDAO:** clase que interactúa con los datos de Usuario y a partir de los mismos crea las instancias de Usuario para ser utilizadas en la aplicación.

- **EmpleadoDAO:** clase que interactúa con los datos de Empleado y Relación y a partir de estos mismos crea las instancias de Empleado y Relación para ser utilizadas en la aplicación.
- **EventoDAO:** clase que interactúa con los datos de Evento y Requisito y a partir de estos mismos crea las instancias de Evento y Requisito para ser utilizadas en la aplicación.
- **RolDAO:** clase que interactúa con los datos de Rol y Habilidad y a partir de estos mismos crea las instancias de Rol y Habilidad para ser utilizadas en la aplicación.
- **AsignacionDAO:** clase que interactúa con los datos de Asignación y a partir de estos mismos crea las instancias de Asignación para ser utilizadas en la aplicación.

La interacción entre las clases DAO mencionadas y los datos se realiza mediante consultas HQL y la API Criteria que son mecanismos que provee el framework Hibernate.

- **OntologyAccess:** clase que interactúa con la ontología. La implementación utilizada para la misma hace uso del framework JENA que provee métodos para cargar la ontología, incorporando el razonador Pellet para realizar inferencias y luego a partir de la ontología resultante conformar las instancias de Recomendación que serán utilizadas por la aplicación.

4.4) COMPORTAMIENTO

Para mostrar el comportamiento principal del sistema confeccionamos por un lado el diagrama de secuencia correspondiente a la funcionalidad de recomendación y asignación de empleados a eventos (mostrados en las figuras 4.7 y 4.8 respectivamente) para describir detalles de implementación de este escenario, incluyendo los objetos y clases que se usan para implementarlo y mensajes intercambiados entre los objetos. Elegimos este requerimiento ya que es donde entra en juego la interacción con la ontología y es el requerimiento en base a la que se construirá la misma como explicaremos en el capítulo 5.

Diagrama de Secuencia para Obtener Recomendación de Empleados:

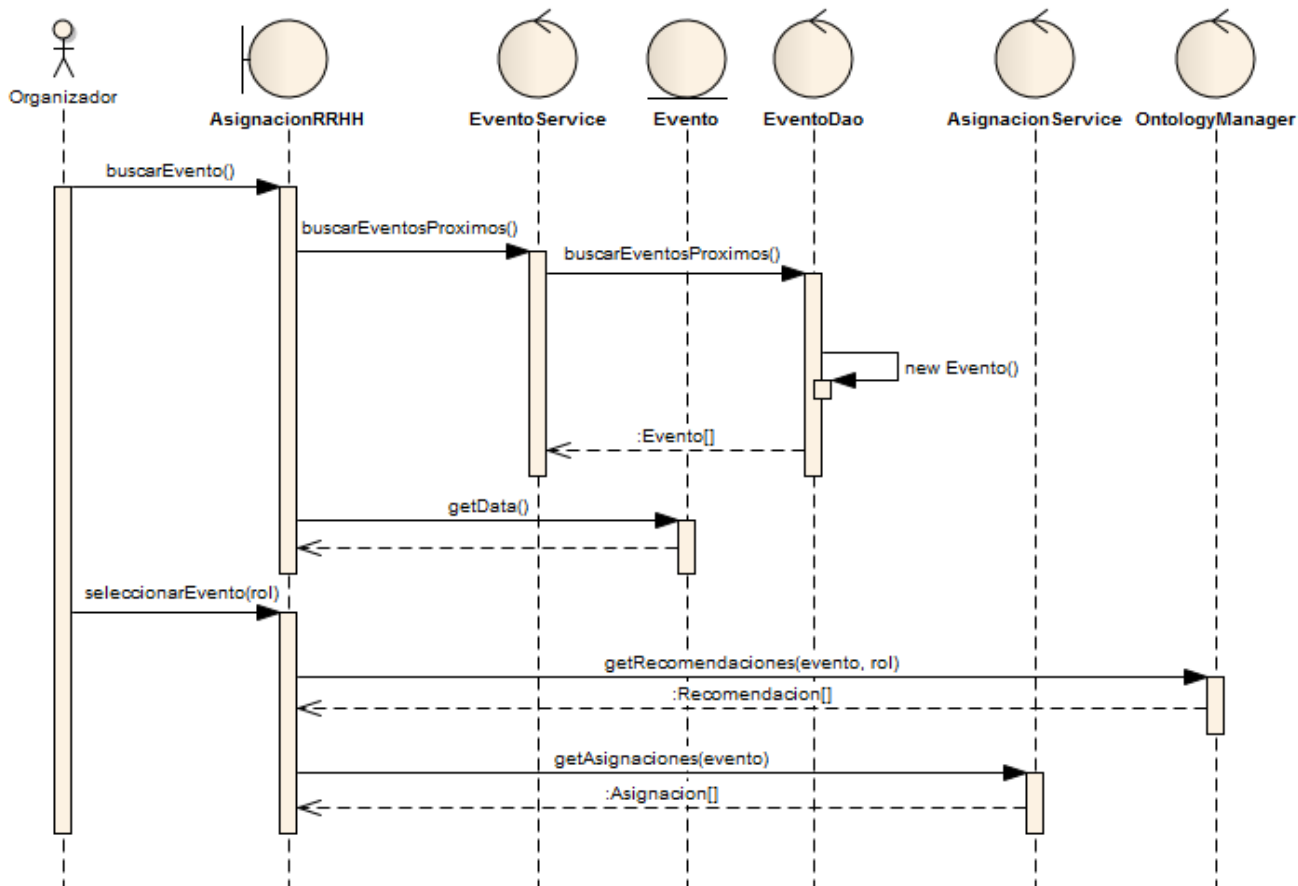


Figura 4.7: Diagrama de Secuencia para el caso de uso Obtener Recomendación de Empleados.

Diagrama de Secuencia para Asignación de Empleados a Eventos:

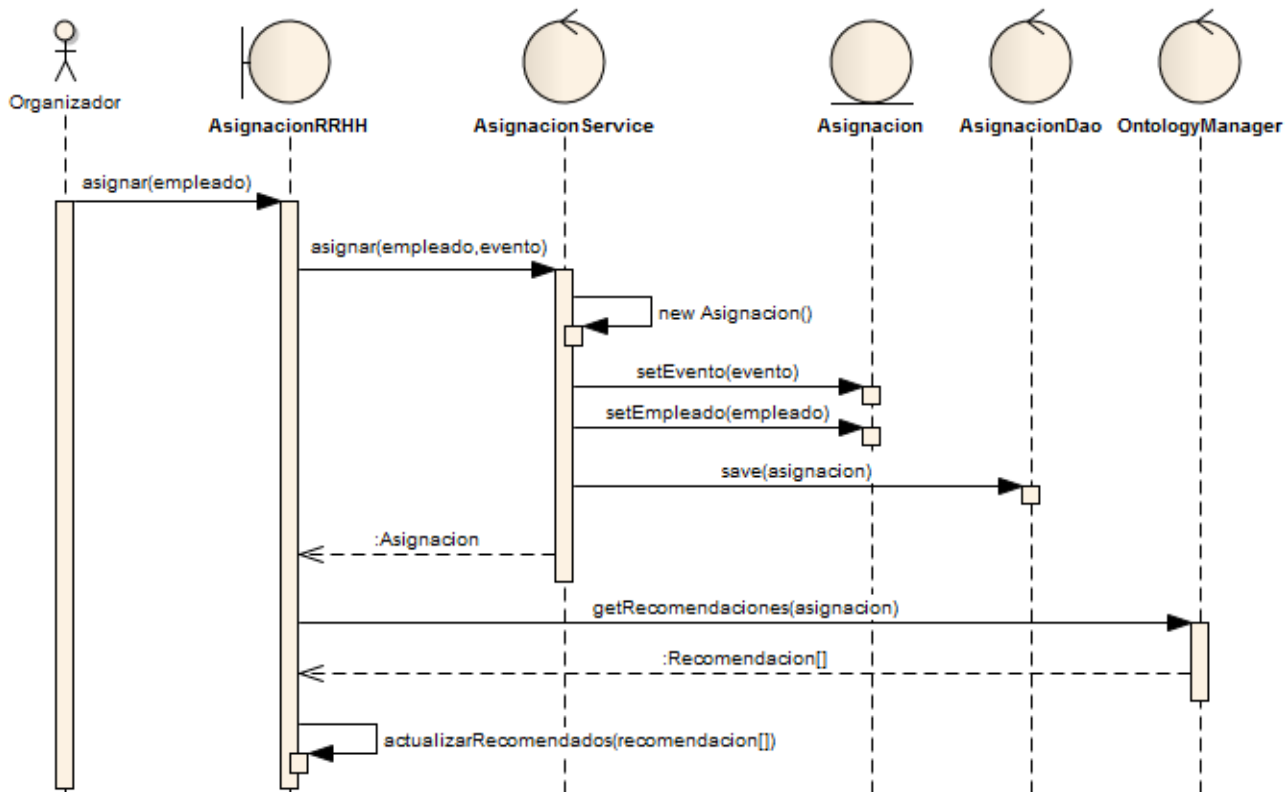


Figura 4.8: Diagrama de Secuencia para el caso de uso Asignación de Empleados a Eventos.

Por otro lado, para modelar parte del comportamiento confeccionamos diagramas de estado de las entidades principales del sistema:

Estados de un Evento:

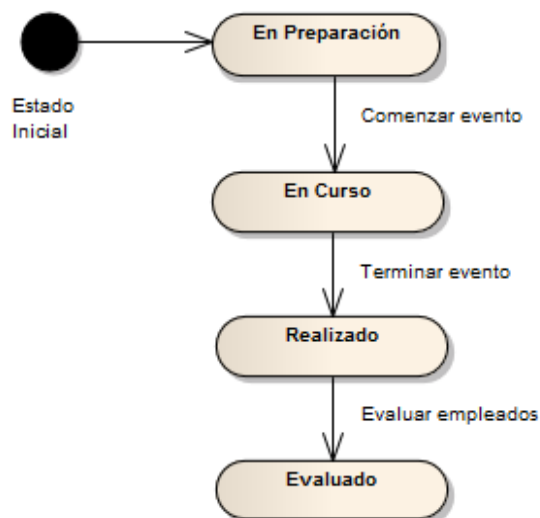


Figura 4.9 Máquina de estados de Evento.

En la figura 4.9 se ilustran los estados de un evento. Los estados del evento son secuenciales. Todo evento comienza en estado de *preparación* ya que se crea previo a su fecha de realización. Mientras el evento se encuentra en este estado es cuando se puede realizar la asignación de los empleados que trabajarán en el mismo. Una vez que llega la fecha del evento éste pasa a estar *en curso*, es decir el evento se está llevando a cabo en ese momento y ya no se podrán realizar modificaciones a la asignación de los empleados. Una vez finalizado el evento éste pasa a estar *realizado* y mientras el evento esté en este estado se podrá calificar a los empleados que trabajaron en el mismo. Una vez calificados todos los empleados que participaron el evento se alcanza el estado final de *evaluado* donde se puede realizar modificaciones a evaluaciones de empleados realizadas previamente durante el estado anterior.

Estados de un Empleado respecto al Evento que se asigna:

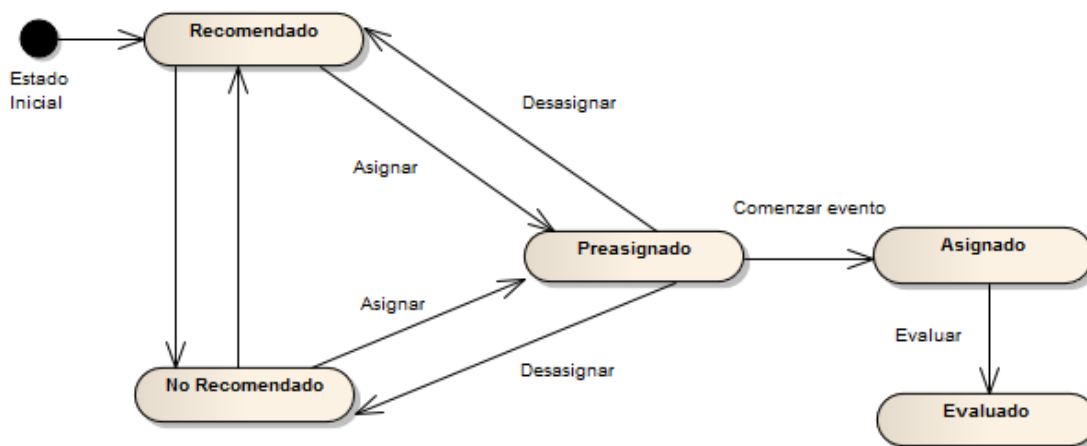


Figura 4.10 Máquina de estados de Empleado.

En la figura 4.10 se ilustran los estados de un empleado. Los empleados comienzan en estado *recomendado* por defecto. Los empleados pueden pasar a ser *no recomendados* en caso de que se cumpla alguna condición de porque no sería conveniente asignar al empleado al evento. Además, los empleados pueden pasar a ser *recomendados* en caso de que dichas condiciones se dejen de cumplir. Estas condiciones por las que un empleado se considera no recomendado son:

- El empleado obtuvo una baja calificación de desempeño.
- Fue asignado al evento un empleado con quien existe una relación conflictiva.
- El empleado no ha participado todavía de ningún evento anteriormente y por lo tanto se considera inexperto.
- El empleado ha sido pre-asignado a otro evento que se realiza en un horario que se superpone con el evento al que se quiere asignar.

Tanto empleados en estado *recomendado* como *no recomendado*, pueden ser pre-asignados al evento para dar la libertad al organizador de tomar la decisión de si quiere dejar trabajar a un empleado no recomendado con el fin de darle la oportunidad de mejorar su desempeño o tener más experiencia, dependiendo de las características y la formalidad del evento al cual se quiere asignar. Una vez llegada la fecha del evento, donde este pasa a estar en curso,

todos los empleados *preasignados* pasan a ser *asignados* definitivamente de manera que ya no pueden disociarse del evento. Los empleados *asignados* pueden recibir una evaluación de su desempeño por trabajar en el evento. Una vez recibida esta evaluación pasan al estado final de *evaluado*.

CAPÍTULO 5: CONSTRUCCIÓN DE LA ONTOLOGÍA

A continuación iremos entrando en detalle en cada uno de los pasos, explicando cómo se fue aplicando la metodología EDON introducida en el capítulo 2 para la construcción de la ontología que utiliza nuestra aplicación.

5.1) PRIMER PASO: PROCESO DE SELECCIÓN DE REQUERIMIENTOS

El primer paso es el proceso de selección de requerimientos que se lleva adelante como muestra la figura 5.1.

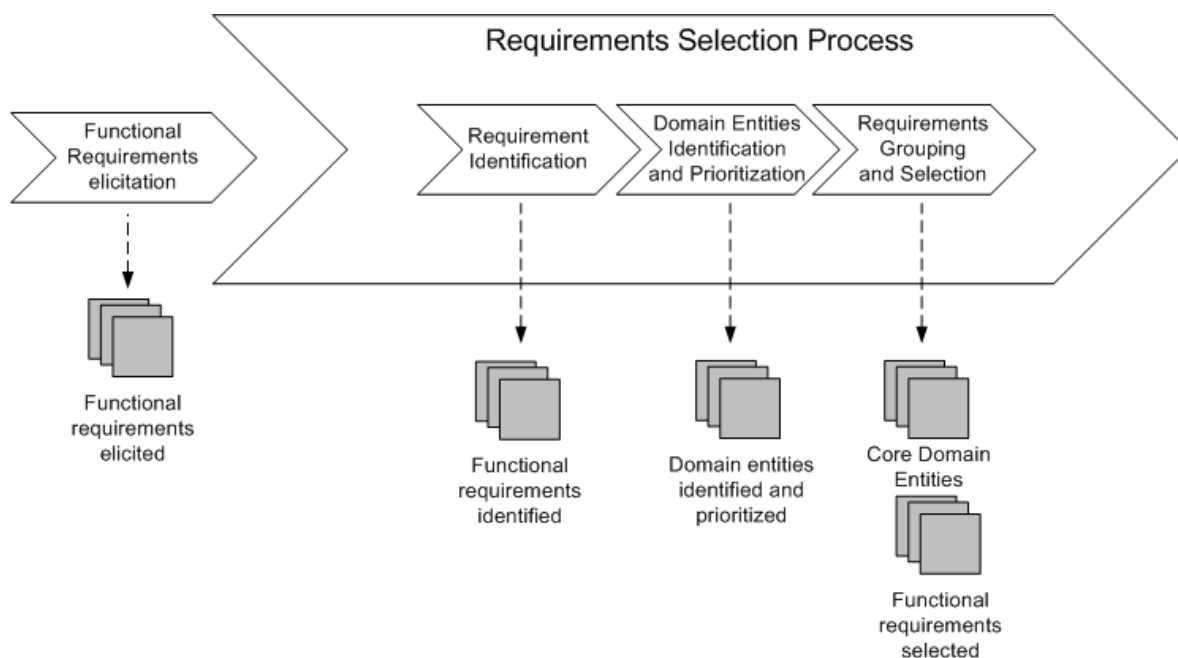


Figura 5.1 Composición del Proceso de Selección de Requerimientos.

Identificación de los requerimientos: para dar inicio al proceso de selección de requerimientos, se comienza por educir un conjunto de requerimientos funcionales, en base de reglas de negocio, que luego deberán ser soportados por la ontología. Para nuestro proyecto definimos que *la funcionalidad a ser soportada por la ontología será la correspondiente a la asignación de empleados a eventos, dando soporte a la fase de obtener un conjunto de empleados recomendados para ser asignados*. La ontología se encargará de inferir las recomendaciones

pertinentes con el fin de guiar al organizador a la hora de determinar los empleados que serán convocados para trabajar en un evento. La idea es implementar un agente Recomendador cuyo conocimiento para dar sus recomendaciones se encuentre en la ontología. A la hora de elegir un evento para realizar la asignación de recursos humanos, la ontología debe cargarse con los datos de los empleados y el evento en cuestión, y a partir de estos realizar las inferencias necesarias para que sea posible obtener listas ordenadas de los empleados recomendados para asignar.

Identificación y Priorización de las Entidades: se identifican un conjunto de entidades, que estén involucradas en los requerimientos funcionales obtenidos en la Identificación de los Requerimientos. Estas entidades, forman parte del núcleo del dominio. Se formulan con preguntas de competencia, que son preguntas informales que la ontología debe ser capaz de responder. Es muy común generar éstas preguntas en lenguaje natural. A su vez, las preguntas de competencia permiten la identificación del conjunto inicial de términos base de la ontología. Dichos términos deben estar dentro del alcance del dominio de la ontología. Estos representan los principales objetos del dominio. Para nuestro caso, el alcance está delimitado por la funcionalidad de recomendar empleados a ser asignados, por lo tanto, todos los términos que estén involucrados de alguna manera en la elaboración de dicha recomendación deben estar presentes en la ontología, debidamente relacionados.

Preguntas de Competencia Definidas:

¿Qué requisitos tiene un evento?

¿Para qué rol se deben tener en cuenta los requisitos del evento?

¿El evento tiene requisitos de habilidades?

¿El evento tiene requisitos de sexo?

¿El evento tiene requisitos de edad?

¿Qué habilidades son necesarias para llevar a cabo un evento?

¿Hay alguna preferencia de edad o sexo de los empleados que deberían trabajar en el evento?

¿El empleado tiene una edad que se encuentre dentro del rango de edad requerido para trabajar en el evento?

¿El empleado cumple con algún requisito del evento?

¿Qué habilidades tiene un empleado?

¿Qué rol desempeña un empleado?

¿Cuáles son los empleados asignados al evento?

¿Qué relación hay entre dos empleados?

¿Pueden participar dos empleados determinados en un mismo evento?

¿Un determinado empleado posee las habilidades necesarias para participar de un evento?

¿El empleado ya ha participado en eventos anteriores?

¿El empleado tiene un buen desempeño en los eventos que ha trabajado anteriormente?

¿Está disponible un determinado empleado temporario para trabajar en un evento que se desarrolla en una fecha?

¿El empleado está asignado a otro evento que se realice en la misma fecha que el evento para el cual se quiere recomendar?

¿Qué requisitos tienen los roles que se deben desempeñar en el transcurso del evento?

¿Cuáles son los empleados que pueden desempeñar un rol requerido en el evento?

¿Es recomendable asignar un determinado empleado a un evento?

¿Es no recomendable la asignación de un empleado a un evento?

¿Por qué motivo no es aconsejable asignar a un empleado al evento?

Términos Base de la Ontología Definidos:

Evento: Los eventos donde se asignan empleados, para los que se lleva a cabo una recomendación. Todo el razonamiento se realizará en base a cada evento.

Requisito: las características deseadas para que un empleado participe de un evento. Pueden ser muy específicos o generales de acuerdo a cuantas características se definen en cada uno.

Habilidad: Una característica que puede pedir un requisito que puede ser cumplida por los empleados.

Sexo: Una característica que puede pedir un requisito que puede ser cumplida por los empleados.

Edad: Una característica numérica que puede pedir un requisito que puede ser cumplida por los empleados.

Rol: Clasifica a los empleados y requisitos, para recomendar empleados que desempeñan un determinado rol.

Asignación: Asociación definitiva de un empleado y un evento.

Empleado: el recurso que puede ser asignado o no al evento.

Empleado Recomendado: recurso que se considera conveniente recomendar al evento.

Empleado No Recomendado: recurso que no se considera conveniente recomendar al evento.

Valoración: puntaje numérico de un empleado que define el grado de recomendación a un evento particular. Mientras mayor sea, más recomendado es a ser asignado.

Motivo: razón de porque un empleado no está recomendado a un evento.

Relación: asociación entre dos empleados indicando un conflicto entre ambos. Se tiene en cuenta a la hora de clasificar como recomendado o no a un empleado cuyo empleado relacionado se encuentre asignado.

Desempeño: puntaje numérico de un empleado de acuerdo a su rendimiento a lo largo de los eventos en que ha participado.

Agrupamiento y Selección de los Requisitos: según las prioridades definidas en la Identificación y Priorización de las Entidades y las entidades halladas, se reúnen y agrupan los requerimientos. El objetivo de este paso es obtener un grupo de requerimientos que incluyan a las entidades básicas de la ontología.

5.2) SEGUNDO PASO: DESARROLLO DE LA ONTOLOGÍA

El segundo paso es el proceso de desarrollo de la ontología que se lleva adelante como muestra la figura 5.3.

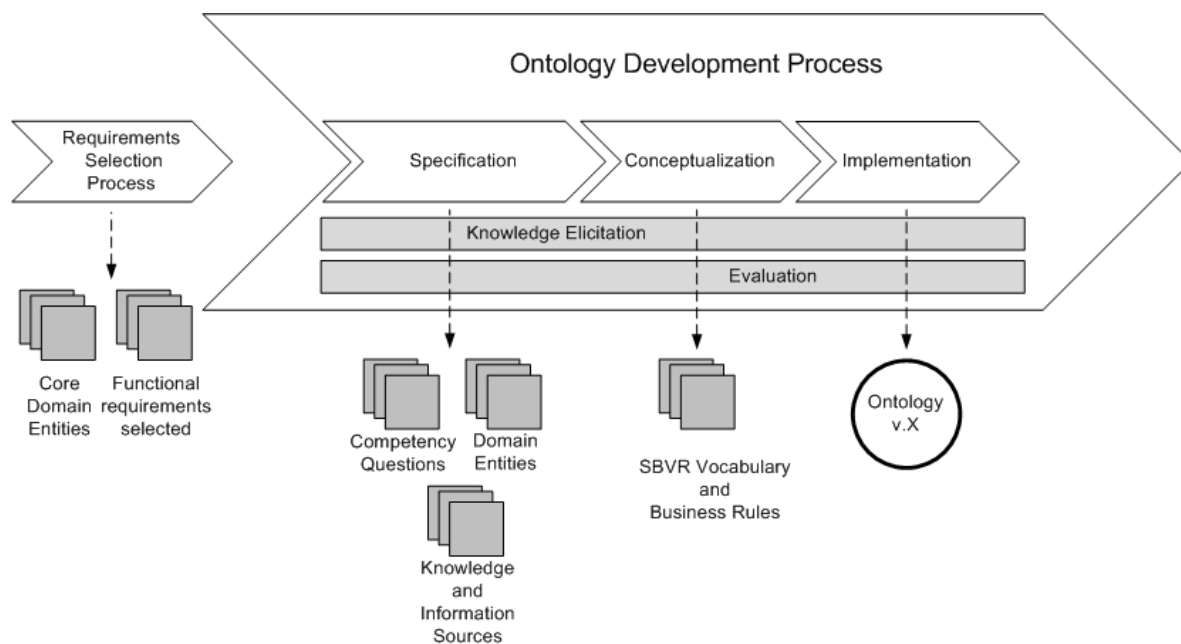


Figura 5.3 Composición del Proceso de Desarrollo de la Ontología.

En este paso se procede al desarrollo de la ontología en sí. Por medio de actividades de desarrollo como especificación, conceptualización, e implementación, sumando además actividades de apoyo como educación del conocimiento y evaluación.

Edución de Conocimiento: En esta actividad se obtienen los conocimientos necesarios para realizar cada una de las Actividades de Desarrollo.

Evaluación: Esta actividad consiste en la verificación, validación que permiten determinar si el proceso es correcto. La Evaluación de la Ontología, comprende verificación de la ontología, tratando de asegurar que sus definiciones implementen los requisitos de manera correcta, validar la ontología, asegurarse que el significado de las definiciones representa verdaderamente el modelo real, para el cual se creó la ontología.

Especificación de Ontología: El objetivo global y las preguntas de competencia se verifican con los requisitos funcionales. Los términos base son verificados de acuerdo a las preguntas de competencia.

Conceptualización de la Ontología: Los conocimientos adquiridos deben estar alineados al objetivo global, los requerimientos funcionales y las condiciones de base. El objetivo es organizar el conocimiento mediante una representación independiente del paradigma de modelado y el lenguaje de implementación. La representación propuesta se basa en SBVR (Semántica de Vocabulario y Reglas de Negocio). SBVR es un estándar adoptado para ser la base formal y detallada de una entidad compleja. SBVR pretende formalizar reglas complejas, como así también las de seguridad y estándares.

Haciendo uso de SBVR se puede definir el vocabulario y las reglas para documentar la semántica del lenguaje de negocios, hechos, y reglas de negocio. Puede aplicarse en dominios relacionados a todo tipo de actividades de negocio y a todo tipo de organizaciones. SBVR es independiente del diseño del sistema de información:

- Consiste en descripciones en lenguaje natural de conceptos, hechos y reglas.
- Las reglas se construyen en base a los hechos, y los hechos se construyen en base a los conceptos. Los términos expresan conceptos del negocio; los hechos son aserciones sobre estos conceptos; Las reglas imponen restricciones sobre estos hechos.

Formalización de la Ontología: La ontología debe ser verificada con el objetivo global, los requerimientos funcionales y los términos SBVR.

El objetivo es generar una ontología base, independiente del lenguaje de implementación. Los términos descritos en la actividad de Conceptualización de Ontología son el núcleo de la ontología que es generada por medio de un proceso sistemático. El proceso se basa en SBVR y permite el desarrollo de una ontología con tres elementos: relaciones, conceptos y reglas. Para la formalización planteamos en primer lugar los conceptos que surgen a partir de los términos obtenidos en la especificación que derivaron de las preguntas de competencia definidas.

Refinamiento de la Ontología: La meta es refinar la ontología base generada por la actividad Formalización de la Ontología, centrándose en la formulación de las restricciones, los axiomas e individuos. Este refinamiento de nuevo puede dar lugar a cambios de tipo de una serie de elementos.

Esta actividad genera una ontología madura, que es un modelo estricto de la realidad y cumple con los requisitos funcionales y de calidad.

EDON, considera el rendimiento de la actividad de refinamiento de la ontología con la expansión de la misma, centrándose en la formulación declarativa de las reglas de negocio. La realización de actividades de desarrollo permite evolucionar desde un modelo abstracto a la ontología. A lo largo del proceso de desarrollo se llevan a cabo tareas de apoyo, que dependen de la naturaleza de la actividad que se realiza.

Aquí definimos los hechos y reglas de SBVR:

- HECHOS:

Empleado **desempeña** Rol

Empleado **tiene relación con** Empleado

Empleado **posee** Habilidad

Empleado **tiene** Desempeño Promedio

Empleado **tiene** Edad

Empleado **tiene** Sexo

Empleado **tiene** Valoración

Empleado **cumple** Requisito

Empleado **es recomendado en** Evento

Empleado **no es recomendado en** Evento

Empleado Recomendado **is a** Empleado

Empleado No Recomendado **is a** Empleado

Empleado No Recomendado **tiene** Motivo

Empleado **está asignado** a Evento

Casamiento **is a** Evento

Cumpleaños de 15 **is a** Evento

Cumpleaños **is a** Evento

Empresarial **is a** Evento

Colación de Grado **is a** Evento

Informal **is a** Evento

Evento **tiene** Etapa

Evento **tiene** Fecha

Recepción **is a** Etapa

Servicio de Mesa **is a** Etapa

Trasnoche **is a** Etapa

Evento **tiene asignado** Empleado

Evento **tiene** Requisito

Requisito **es para** Rol

Requisito **pide** Habilidad

Requisito **pide** Sexo

Requisito **pide** Edad Mínima

Requisito **pide** Edad Máxima

- REGLAS para Empleado:

Cada empleado cumple un rol:

each Empleado desempeña **exactly one** Rol

Cada empleado puede tener relación con uno o más empleados empleado:

some Empleado tiene relación con **some** Empleado

Existe al menos un motivo por el cual un empleado es no recomendado:

each Empleado No Recomendado tiene **at least one** Motivo

Los empleados pueden tener un desempeño promedio si han participado anteriormente en eventos:

some Empleado tiene **exactly one** Desempeño Promedio

Los empleados tienen una valoración, un sexo, una edad:

each Empleado tiene **exactly one** Sexo

tiene **exactly one** Edad

tiene **exactly one** Valoración

Los empleados pueden tener habilidades:

each Empleado posee **some** Habilidad

Los empleados pueden cumplir con los requisitos:

each Empleado cumple **some** Requisito

Un empleado cumple con un requisito cuando tiene la edad, habilidad, y/o sexo que dicho requisito pide:

if empleado tiene sexo1 **and** empleado tiene edad **and** empleado tiene habilidad1 **and** requisito pide sexo2 **and** sexo1=sexo2 **and** requisito pide edadMinima **and** requisito pide

$edadMaxima \text{ and } edad > edadMinima \text{ and } edad > edadMaxima \text{ and } requisito \text{ pide } habilidad2 \text{ and } habilidad1 = habilidad2 \text{ then empleado cumple requisito}$

- REGLAS para Eventos:

Los eventos se pueden tener hasta tres etapas (recepción, servicio de mesa, trasnoche):

some Evento tiene **at most 3** Etapa

tiene **at most 1** Recepción

tiene **at most 1** Servicio de Mesa

tiene **at most 1** Trasnoche

Los eventos tienen una fecha:

each Evento tiene **exactly one** Fecha

Los eventos pueden tener asignados empleados:

each Evento tiene asignado a **some** Empleado:

Los eventos pueden tener requisitos:

some Evento tiene **some** Requisito

Los requisitos se definen para un rol:

each Requisito es para **exactly one** Rol

Los requisitos pueden o no pedir un sexo, una habilidad, una edad mínima y/o una edad máxima

some requisito Requisito pide **at most one** Sexo

pide **at most one** Habilidad

pide **at most one** Edad Mínima

pide **at most one** Edad Máxima

- REGLAS para Recomendación:

Un empleado es recomendado siempre que no existan razones para no serlo:

empleado is a Empleado Recomendado **or** empleado is a Empleado No Recomendado **but not both**

Si un empleado A está relacionado a otro empleado B y empleado B está asignado al evento, el empleado A será no recomendado a ser asignado a dicho evento:

if evento tiene asignado a empleadoA **and** empleadoA tiene relación con empleadoB **then** empleadoB is a Empleado No Recomendado **and** empleadoB tiene motivo = "Hay conflicto con empleado ya asignado"

Si un empleado está asignado a un evento A y el evento A se realiza en la misma fecha que el evento B, el empleado en cuestión no será recomendado para el evento B:

if eventoA tiene fecha **and** eventoB tiene fecha **and** eventoA tiene asignado a empleado **then** empleado is a Empleado No Recomendado **and** empleado tiene motivo = "El empleado ya se encuentra asignado a otro evento esa fecha"

Un empleado no será recomendado cuando su desempeño promedio sea bajo (siendo el máximo desempeño = 5, se considerará no recomendado a aquellos empleados con un valor menor a la mitad):

If empleado tiene desempeño **and** desempeño < 2.5 **then** empleado is a Empleado No Recomendado **and** empleado tiene motivo = "El desempeño promedio del empleado es bajo"

Un empleado que no ha participado nunca de un evento será considerado no recomendado. Un empleado que no participó nunca de un evento no tiene desempeño promedio:

If is not the case that empleado tiene desempeño **then** empleado is a Empleado No Recomendado **and** empleado tiene motivo = “No ha participado de ningún evento aun”

Se puede observar como las reglas para llevar a cabo la recomendación consiste en determinar cuando un empleado NO está recomendado a participar del evento. Todos los empleados para los que no exista una regla que los clasifique como no recomendados se considerarán recomendados. Definimos una consulta SPARQL para calcular la valoración de los empleados y el puntaje de la recomendación. La clasificación entre empleado recomendado o no, no está directamente relacionada al puntaje de la recomendación, ya que la NO recomendación se da por motivos particulares más allá el puntaje. Por ejemplo, un empleado puede tener una muy buena valoración, pero si previamente se asignó al evento un empleado relacionado a él se considerara no recomendado. De la misma manera un empleado que cumpla con todos los requisitos de un evento pero su desempeño ha sido muy bajo a lo largo de sus asignaciones se considerará no recomendado con un alto nivel de valoración. Esto sirve para guiar al usuario en que si bien un empleado puede tener un alto puntaje de recomendación, hay cuestiones que debe tener en cuenta para realizar la decisión final de asignar de dicho empleado al evento.

Implementación de la Ontología: Las tareas de evaluación son continuas a lo largo de las actividades de desarrollo.

El objetivo es implementar la ontología madura obtenida en la actividad de Refinamiento de la Ontología.

La actividad genera una ontología computable, implementada en un lenguaje de máquina. Esta ontología reúne todos los requisitos funcionales llevados a cabo hasta la iteración actual. El lenguaje de implementación se debe seleccionar antes de desarrollar la primera versión de la ontología, teniendo en cuenta el entorno de ejecución del sistema y los requisitos funcionales para satisfacer. En este proyecto utilizamos OWL (Ontology Web Language) y SWRL (Semantic Web Rule Language) para la implementación, lenguajes que fueron presentados en nuestro marco teórico. Para la implementación se realizó un mapeo desde el desarrollo de la ontología genérica planteada en SBVR a términos OWL/SWRL.

Para llevar a cabo la implementación de la ontología se utilizó el software **Protégé**. **Protégé** es un editor de ontologías gratuito y libre y un sistema de gestión de conocimiento. Provee una interfaz gráfica de usuario para definir las ontologías. Además incluye clasificadores deductivo para validar modelos consistentes e inferir nueva información a partir del análisis de una ontología.

La determinación de la Valoración de cada Recomendación se implementa mediante una consulta de SPARQL. La misma consiste en contabilizar la cantidad de requisitos que tiene el evento que cumple cada empleado y de ahí otorgar una valoración más alta a aquellos que cumplan más requisitos. También se suma a la valoración un valor que dependerá del desempeño promedio de cada empleado. La consulta se encarga de devolver el listado de empleados ordenados por la valoración resultante.

La consulta resultante es la siguiente:

```
SELECT      ?empleado      ?desempeno      (count(?requisito)      as
?numRequisitos)
WHERE {
    ?empleado a :Empleado .
    ?empleado :desempeno_promedio ?desempeno .
    OPTIONAL {?empleado :cumple_requisito ?requisito }
}
GROUP BY ?empleado ?desempeno
ORDER BY DESC(?numRequisitos) DESC(?desempeno)
```

CAPÍTULO 6: PRUEBA DE FUNCIONALIDADES

En este capítulo mostramos como las funcionalidades, explicadas a lo largo del informe, se ven con sus respectivas interfaces gráficas en la aplicación finalmente desarrollada y como el usuario deberá interactuar con las mismas.

6.1) AUTENTICAR USUARIO

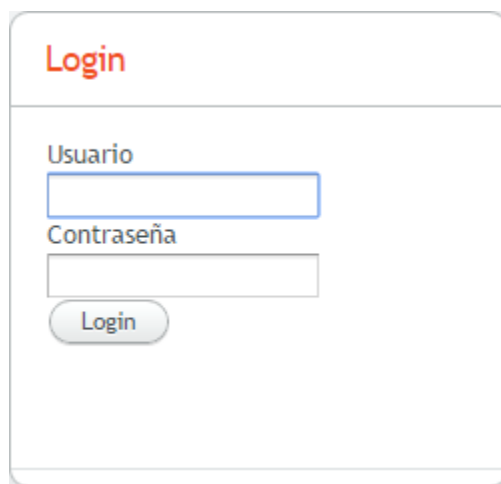


Figura 6.1 Pantalla de Autenticación de Usuario.

La primera pantalla que se le presenta al usuario del sistema es la pantalla de autenticación mostrada en la figura 6.1. La autenticación es el proceso de verificar la identidad digital de un usuario en su petición para conectarse. Se presentan en pantalla un primer campo para ingresar el nombre del usuario y un segundo campo para ingresar la contraseña el cual se encuentra enmascarado y su contenido se visualiza como círculos (●) ya que es confidencial. Una vez completados los campos se presiona el botón “LOGIN” y se verificará que el nombre de usuario y la contraseña ingresados sean válidos. Los usuarios con sus contraseñas se encuentran guardados en la base de datos. Una vez que el usuario se autenticó con éxito se le presentará en pantalla la interfaz gráfica de la aplicación que le permitirá acceder a las funcionalidades correspondientes.

En caso de que falle la autenticación se mostrará al usuario que lo ingresado no corresponde a un usuario y/o contraseña válidos y se le permitirá ingresar los valores nuevamente.

6.2) GESTION DE ROLES Y HABILIDADES

A continuación explicaremos las funcionalidades a las que puede acceder un usuario ya autenticado como organizador en la aplicación, comenzando por la gestión de roles y habilidades.

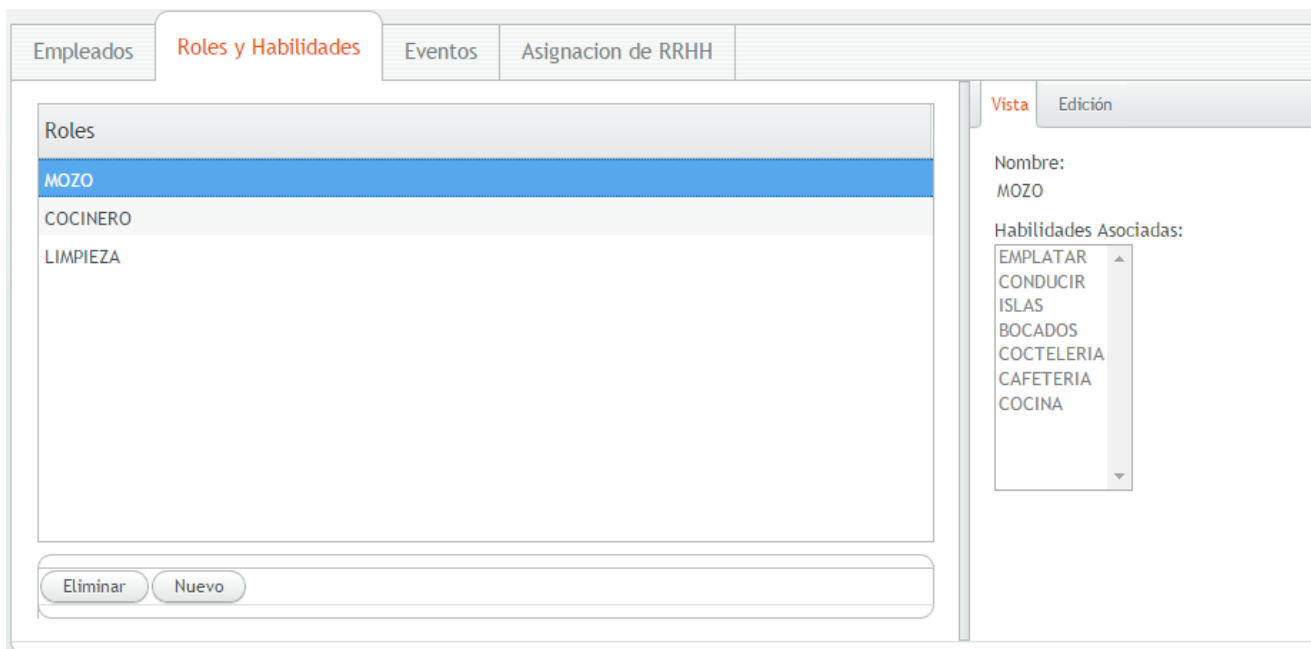


Figura 6.2 Pantalla de Gestión de Roles y Habilidades.

En la pantalla ilustrada en la figura 6.2 se muestran por un lado, en la parte izquierda de la pantalla, una tabla donde se enumeran los roles que se encuentran registrados en el sistema. En el título de la columna puede hacerse clic para reordenar los roles por orden alfabético, ascendente o descendente alternativamente mientras se siga presionando sobre el título. Al seleccionar un rol de la tabla se mostrarán en la parte derecha de la interfaz unas pestañas que son:

- Vista: en esta pestaña se muestran los atributos correspondientes al rol seleccionado en forma de solo lectura. Se puede observar un campo de texto con el nombre del rol y una lista con las habilidades que están asociadas a ese rol. Esta pestaña es la activada

inicialmente cuando se selecciona un rol de la tabla para permitir ver rápidamente los datos del mismo.

- Edición: en esta pestaña, en forma similar a la vista, se muestran los atributos correspondientes al rol seleccionado pero de manera tal que se pueden editar los valores. El nombre del rol puede ser editado desde el campo de texto correspondiente el cual es obligatorio como marca el asterisco (*) a su lado. En el caso de las habilidades se agrega un campo de texto donde se puede ingresar el nombre de una nueva habilidad y luego al presionar el botón “+” quedará agregada a la lista de habilidades asociadas. Para guardar los cambios y queden registrados se debe presionar el botón “Guardar” en la parte inferior derecha de la pantalla.

En la parte inferior izquierda de la pantalla hay dos botones. El botón “Eliminar” sirve para eliminar el rol seleccionado de la tabla permanentemente. El botón “Nuevo” permite la incorporación de un nuevo rol. Al hacer clic en botón nuevo se activará a la derecha una ventana de edición de rol (similar a cuando se selecciona la pestaña luego de seleccionar un rol de la tabla, pero en este caso con campos vacíos para ser completados) para ingresar el nombre del nuevo rol y asociar habilidades correspondientes. Una vez ingresados estos datos se puede persistir definitivamente el rol creado haciendo clic en el botón “Guardar”.

6.3) GESTIÓN DE EMPLEADOS

Para llevar a cabo la funcionalidad de gestión de empleados implementamos la interfaz gráfica mostrada en la figura 6.3:

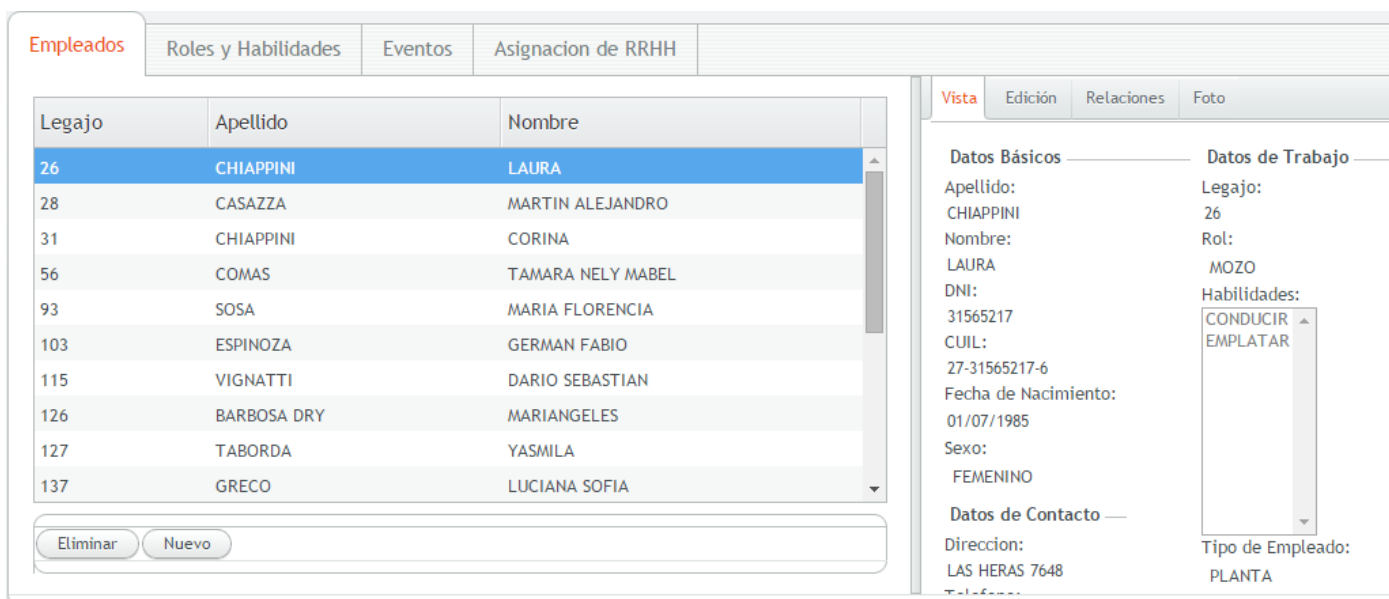


Figura 6.3 Pantalla de Gestión de Empleados.

De manera similar a la gestión de roles, esta pantalla cuenta por un lado con un sector izquierdo donde se encuentran enumerados todos los empleados en una tabla donde se ve el legajo, el apellido y el nombre para identificarlos. Esta tabla puede ser reordenada ascendente o descendentemente por cualquiera de sus columnas al hacer clic sobre el título de la misma. Al seleccionar un empleado se habilita la parte derecha de la interfaz gráfica que consiste en un grupo de pestañas donde se encuentran las siguientes pantallas:

- **Vista:** en esta pestaña se muestran los atributos correspondientes al empleado seleccionado en forma de solo lectura. Se puede observar un campo de texto para los atributos textuales como nombre, apellido, DNI, etc. y se muestra una lista donde se ven los nombres de las habilidades que el empleado posee.. Esta pestaña es la activada inicialmente cuando se selecciona un empleado de la tabla para permitir ver rápidamente los datos del mismo.
- **Edición:** en esta pestaña, en forma similar a la vista, se muestran los atributos correspondientes al empleado seleccionado pero de manera tal que se pueden editar los valores. Los atributos apellido, nombre, DNI, legajo, CUIL, dirección, teléfono, y e-mail son mostrados en campos de texto donde se puede editar el contenido. La fecha de

nacimiento es un campo de fecha que a su lado tiene un icono que al presionarlo se abre un calendario que facilite la búsqueda y selección de una fecha en particular. El sexo, rol y tipo de empleado se muestran como listas desplegables ya que sus valores deben ser seleccionados a partir de un conjunto de opciones disponibles. Las habilidades aparecen en forma de casillas seleccionables de acuerdo al rol que se seleccione ya que cada rol está asociado a un conjunto concreto de habilidades. Para guardar los cambios y queden registrados se debe presionar el botón “Guardar” en la parte inferior derecha de la pantalla. Para poder guardar los cambios deben estar completos mínimamente todos los campos donde se muestre el asterisco (*) ya que indica que son obligatorios.

- Relaciones: esta pestaña se visualiza como se muestra en la figura 6.4:

The screenshot shows a web application interface for employee management. The main menu includes 'Empleados', 'Roles y Habilidades', 'Eventos', and 'Asignación de RRHH'. The 'Empleados' section contains a table with columns 'Legajo', 'Apellido', and 'Nombre'. Below this table are 'Eliminar' and 'Nuevo' buttons. The 'Relaciones' tab is active, showing a table with columns 'Legajo', 'Apellido', 'Nombre', and 'Motivo'. The relationship table contains two entries: one for legajo 28 (CASAZZA, MARTIN ALEJANDRO) with motive 'Enemistad', and one for legajo 56 (COMAS, TAMARA NELLY MABEL) with motive 'Expareja'. Below the relationship table are 'Eliminar' and 'Nueva Relacion' buttons.

Figura 6.4 Pantalla de Gestión de Empleados - Relaciones.

Consiste de una tabla donde se enumeran los empleados con los que el empleado seleccionado de la izquierda mantiene una relación. Esta tabla muestra el legajo, el apellido y el nombre para identificar al empleado, y además hay una columna donde se muestra el motivo de dicha relación. La tabla puede ser ordenada ascendente o descendientemente por cada una de sus columnas al hacer clic en el título de las mismas.

Una relación de la tabla puede ser seleccionada para luego poder ser eliminada presionando el botón “Eliminar”. En caso de querer agregar una nueva relación para el empleado

seleccionado en la tabla izquierda se debe presionar el botón “Nueva Relación” que abre una ventana emergente para poder crearla. La figura 6.5 muestra como se compone la ventana:

Legajo	Apellido	Nombre	DNI
26	CHIAPPINI	LAURA	31565217
28	CASAZZA	MARTIN ALEJANDRO	28882161
31	CHIAPPINI	CORINA	30489480
56	COMAS	TAMARA NELY MABEL	33686861
93	SOSA	MARIA FLORENCIA	35448788

Motivo

Figura 6.5 Pantalla de Gestión de Empleados – Ventana Emergente para crear una relación.

En esta ventana se muestra una tabla con todos los empleados registrados la cual puede ordenarse por cualquiera de sus columnas. Una vez seleccionado un empleado de la tabla e ingresada una descripción del motivo en el campo de texto se puede presionar el botón Guardar para que dicha relación quede registrada definitivamente. Cabe mencionar que la relación figurará tanto para el empleado que se encontraba seleccionado en primer lugar como para el empleado que se seleccionó en esta última ventana, es decir que se crea en forma bidireccional.

- Foto: en esta pestaña se implementa un componente gráfico, mostrado en la figura 6.6, que permite la carga de una imagen para guardar una fotografía que represente al empleado. Este componente nos permite buscar una imagen en el sistema de archivo del dispositivo del cual se está accediendo a la aplicación que será guardada en el servidor asociada a la entidad de empleado correspondiente.

Empleado

Roles y Habilidades Eventos Asignación de RRHH

Legajo	Apellido	Nombre
26	CHIAPPINI	LAURA
28	CASAZZA	MARTIN ALEJANDRO
31	CHIAPPINI	CORINA
56	COMAS	TAMARA NELLY MABEL
93	SOSA	MARIA FLORENCIA
103	ESPINOZA	GERMAN FABIO
115	VIGNATTI	DARIO SEBASTIAN
126	BARBOSA DRY	MARIANGELES
127	TABORDA	YASMILA
137	GRECO	LUCIANA SOFIA

Eliminar Nuevo

Vista Edición Relaciones Foto

Cargar la foto aquí

Seleccionar archivo Ningún archivo seleccionado Cargar

Foto cargada:



Figura 6.6 Pantalla de Gestión de Empleados – Solapa de carga de foto.

6.4) GESTIÓN DE EVENTOS

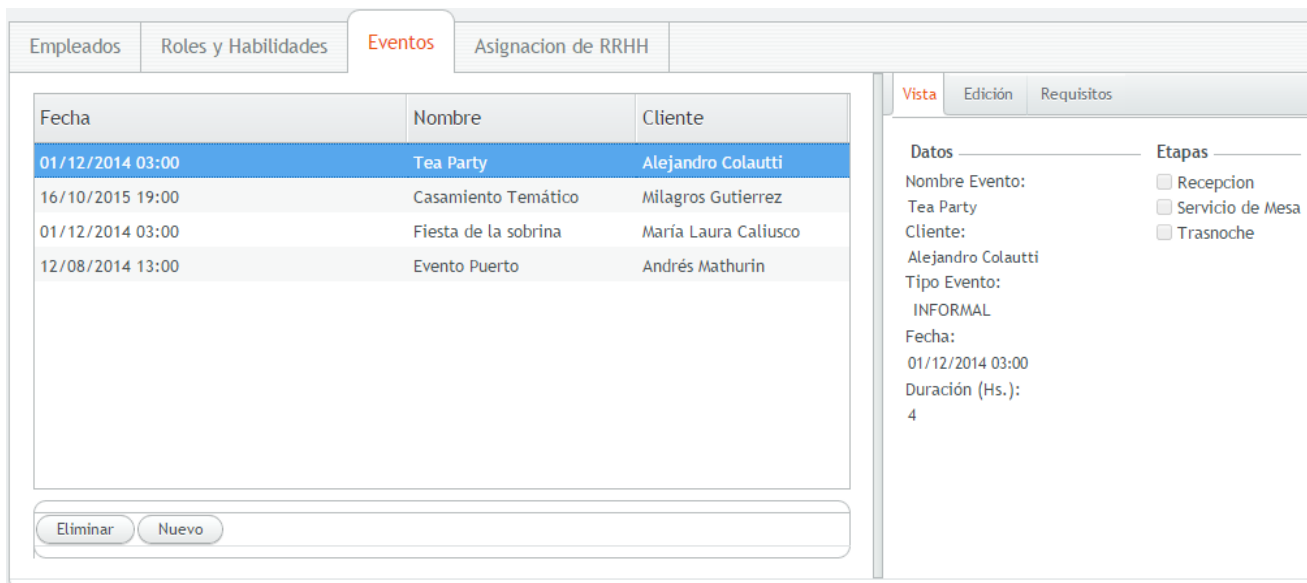


Figura 6.7 Pantalla de Gestión de Eventos.

Siguiendo la línea de las pantallas de gestión mostradas anteriormente, esta pantalla, como muestra la figura 6.7, cuenta por un lado con un sector izquierdo donde se encuentran enumerados todos los eventos en una tabla donde se ve la fecha y hora, un nombre o descripción y el nombre del cliente para identificarlos. Esta tabla puede ser reordenada ascendente o descendientemente por cualquiera de sus columnas al hacer clic sobre el título de la misma. Al seleccionar un evento se habilita la parte derecha de la interfaz gráfica que consiste en un grupo de pestañas donde se encuentran las siguientes pantallas:

- **Vista:** en esta pestaña se muestran los atributos correspondientes al evento seleccionado en forma de solo lectura. Se puede observar un campo de texto para los atributos textuales como el nombre del cliente y el nombre del evento. Las etapas se muestran como unas casillas seleccionables que están marcadas en caso de que el evento cuente con cada etapa en particular. Esta pestaña es la activada inicialmente cuando se selecciona un empleado de la tabla para permitir ver rápidamente los datos del mismo.
- **Edición:** en esta pestaña, en forma similar a la vista, se muestran los atributos correspondientes al evento seleccionado pero de manera tal que se pueden editar los

valores. Los atributos nombre de evento y nombre de texto son mostrados en campos de texto donde se puede editar el contenido. El campo de duración es un campo de texto donde se debe ingresar un número indicando la cantidad de horas aproximadas que durará el evento desde el inicio hasta que se concluya. La fecha de evento es un campo de fecha que a su lado tiene un icono que al presionarlo se abre un calendario que facilite la búsqueda y selección de una fecha en particular y el horario. El tipo de evento se muestra como listas desplegable ya que sus valores deben ser seleccionados a partir de un conjunto de opciones disponibles (Casamiento, Cumpleaños, Empresarial, Informal, Colación). Las etapas aparecen en forma de casillas de manera que puede seleccionarse la que corresponda si es que el evento se divide en etapas y si es que consiste de todas o sólo alguna/s de ellas. Para guardar los cambios y queden registrados se debe presionar el botón “Guardar” en la parte inferior derecha de la pantalla. Para poder guardar los cambios deben estar completos mínimamente todos los campos donde se muestre el asterisco (*) ya que indica que son obligatorios.

- Requisitos: esta pestaña se muestra en la figura 6.8:

The screenshot shows a web application interface with a main menu at the top containing 'Empleados', 'Roles y Habilidades', 'Eventos', and 'Asignacion de RRHH'. The 'Eventos' tab is active, displaying a table with the following data:

Fecha	Nombre	Cliente
01/12/2014 03:00	Tea Party	Alejandro Colautti
16/10/2015 19:00	Casamiento Temático	Milagros Gutiérrez
01/12/2014 03:00	Fiesta de la sobrina	María Laura Caliusco
12/08/2014 13:00	Evento Puerto	Andrés Mathurin

Below the table are 'Eliminar' and 'Nuevo' buttons. To the right, the 'Requisitos' sub-tab is active, showing a table with the following data:

Rol	Sexo	Edad Minima	Edad Maxima	Habilidad
MOZO	MASCULINO	18	28	COCTELERIA

Below this table are 'Eliminar' and 'Nuevo Requisito' buttons.

Figura 6.8 Pantalla de Gestión de Eventos – Solapa de carga de Requisitos.

Consiste de una tabla donde se enumeran los requisitos que se tendrán en cuenta a la hora de que el sistema deba recomendar empleados para ser asignados al evento seleccionado en la tabla izquierda. Esta tabla muestra el rol, el sexo, la edad mínima y máxima y una habilidad para definir un requisito. Los requisitos pueden ser muy específicos y tener valores en todas las

columnas o pueden ser más generales especificando sólo algunas de estas características y dejando las demás sin especificar. La tabla puede ser ordenada ascendente o descendentemente por cada una de sus columnas al hacer clic en el título de las mismas.

Un requisito de la tabla puede ser seleccionado para luego poder ser eliminado presionando el botón “Eliminar”. En caso de querer agregar una nuevo requisito para el evento seleccionado en la tabla izquierda se debe presionar el botón “Nuevo Requisito” que abre la siguiente ventana para poder crearla:

6.5) ASIGNAR EMPLEADOS A EVENTOS

The screenshot shows a web application interface for 'Asignación de RRHH'. It features a navigation bar with tabs for 'Empleados', 'Roles y Habilidades', 'Eventos', and 'Asignación de RRHH'. Below the navigation, there are sub-tabs for 'Asignación de Empleados' and 'Evaluación de Empleados'. A search button 'Buscar Evento Próximo' is present, along with event details: 'Evento Seleccionado: MOZO para Casamiento Temático (CASAMIENTO) Fecha: 16/10/2015 19:00 Cliente: Milagros'. The interface is divided into two main sections: 'Recomendados' and 'Asignados'. The 'Recomendados' section contains a table with columns 'Legajo', 'Apellido', 'Nombre', and 'Valoración'. The 'Asignados' section contains a table with columns 'Legajo', 'Apellido', 'Nombre', 'Rol', 'Inicio', and 'Fin'. Below the 'Recomendados' table are buttons for 'Asignar >' and '< Desasignar'. At the bottom right, there are buttons for 'Notificar Asignados', 'Descargar Planilla', and 'Guardar Horarios'. On the bottom left, there are observation fields for 'Edad: 31' and 'Habilidades:'.

Legajo	Apellido	Nombre	Valoración
219	ROMERO	PABLO	1
221	SOSA	LEANDRO	1
103	ESPINOZA	GERMAN FABIO	0,8
145	MONZALVO	GUADALUPE	0,6
222	COMAS	MARIA DEL ROSARIO	0,6

Legajo	Apellido	Nombre	Rol	Inicio	Fin
219	ROMERO	PABLO	MOZO	19:00	23:00
221	SOSA	LEANDRO	MOZO	19:00	23:00

Figura 6.9 Pantalla de asignación de RRHH.

En esta interfaz gráfica se distribuyen dos tablas con información de empleados como se muestra en la figura 6.9. Al principio las tablas se encontraran vacías hasta que el usuario presione el botón de “Buscar Evento Próximo” en donde se mostrará al usuario una ventana conteniendo una tabla con todos los eventos con fecha próxima en los cuales se puede realizar la asignación de empleados (es decir eventos en estado “En Preparación” teniendo en cuenta la máquina de estados expuesta en el capítulo 4 para las instancias de Evento). El usuario deberá seleccionar, además de un evento de la tabla, el rol para el que quiere obtener las recomendaciones. Una vez

ingresado estos datos, la ventana se cierra y se cargaran las tablas. Por un lado la tabla de la izquierda contiene los empleados recomendados y no recomendados (separados en dos pestañas). Esta tabla está inicialmente ordenada según la valoración, que es mayor de acuerdo a que tan recomendable es un empleado para ser asignado (teniendo en cuenta como sus atributos y características se adapten a los requisitos del evento). La tabla de la derecha muestra los empleados asignados hasta el momento en el evento indicando una hora de inicio y hora de fin del trabajo que por defecto se inicializa con la hora de inicio del evento y el final teniendo en cuenta su duración total.

Una vez que se selecciona un empleado de la tabla de recomendaciones de la izquierda, este puede ser asignado si se presiona el botón "Asignar", lo cual puede producir una modificación en las recomendaciones de la tabla si es que el empleado que se asignó mantuviera alguna relación con alguno que no ha sido asignado aún. De la misma manera se pueden desasignar empleados seleccionados en la tabla de asignaciones de la derecha y presionando el botón "Desasignar", revirtiendo los cambios que pudo haber causado la asignación de dicho empleado en cuanto a la tabla de recomendaciones. Para guardar las modificaciones de horarios en las asignaciones en caso de que no se desee que el empleado trabaje durante todo el evento se debe presionar el botón guardar horarios para que queden registrados los cambios.

Otras funcionalidades a las que se puede acceder desde esta pantalla son la notificación de empleados, donde presionando el botón "Notificar Empleados" se enviará un correo electrónico a los empleados que se encontrarán en la tabla de asignaciones en ese momento indicándoles de su asignación y el horario que deberán cumplir.

Al presionar el botón "Descargar Planilla" se abre un cuadro de diálogo para descargar el archivo .xls que tiene la información de las asignaciones de los empleados al evento que se está gestionando en ese momento.

6.6) ASIGNAR DESEMPEÑO A EMPLEADOS

Evento Seleccionado:
Evento Puerto (COLACION) Fecha: 12/08/2014 13:00 Cliente: Andrés Mathurin

Legajo	DNI	CUIL	Apellido	Nombre	Rol	Inicio	Fin	Desempeño
170	34147592	23-34147592-9	COLOMBINI	FRANCISCO	MOZO	13:00	14:00	4 - Bueno
219	36127910	20-36127910-8	ROMERO	PABLO	MOZO	13:00	15:00	5 - Muy Bueno
221	39631651	20-39631651-0	SOSA	LEANDRO	MOZO	13:00	16:00	5 - Muy Bueno
103	27762151	20-27762151-3	ESPINOZA	GERMAN FABIO	MOZO	13:00	17:00	5 - Muy Bueno

Descargar Plantilla

Figura 6.10 Pantalla evaluación de empleados.

Esta interfaz está compuesta principalmente de una única tabla como se muestra en la figura 6.10. Al igual que en la asignación de empleados a eventos, en esta pantalla se busca un evento, pero en este caso se busca un evento pasado, es decir que ya haya pasado su fecha y esté terminado. Una vez seleccionado un evento se verán en la tabla los empleados que trabajaron en éste, permitiendo asignarle a cada uno un desempeño con escala del 1 al 5. En esta pantalla puede descargarse una plantilla de la misma manera que en la asignación de empleados haciendo clic sobre el botón correspondiente.

CAPÍTULO 7: CONCLUSIÓN

Finalmente llegamos a las conclusiones una vez terminado el proyecto final de carrera. La idea principal de este proyecto fue desarrollar una aplicación a partir de necesidades de un cliente mientras hacer uso de una herramienta novedosa para su implementación. Explicaremos a continuación como se instaló la aplicación para que el cliente pueda realizar las pruebas necesarias y luego se exponen las conclusiones finales del proyecto.

7.1) INSTALACIÓN DE APLICACIÓN PARA REALIZAR PRUEBAS

Para que el sistema pueda ejecutarse por usuarios y que se realicen pruebas de sus funcionalidades, instalamos en una PC del cliente el software XAMPP (Ver Anexo I). Este software permite gestionar un servidor local (Apache Tomcat) y una base de datos MySQL, que se utilizaron para desplegar la aplicación. Para poder realizar pruebas, la aplicación se levanta en un entorno local, y mientras este en ejecución se puede acceder a la misma desde cualquier dispositivo a través de un navegador introduciendo como dirección la dirección IP del equipo en que se está ejecutando.

7.2) CONCLUSIONES FINALES

En este proyecto determinamos que era oportuno el uso de ontologías por las características de la aplicación en sí ya que la misma consistió de un módulo de asignación de recursos humanos con recomendación. La necesidad de llevar a cabo una recomendación de empleados, ordenándolos de mejor a peor en base a determinadas características nos pareció una buena oportunidad para experimentar y desarrollar una ontología subyacente a la aplicación y dejarle la responsabilidad de dar soporte a dicha funcionalidad. De esta manera evitamos llevar a cabo un razonamiento engorroso en forma de código java, lo cual se vuelve poco mantenible. Mantener una ontología externa al código de la aplicación nos da la ventaja de evitar realizar nuevamente un despliegue de la aplicación en caso de requerir realizar cambios en la lógica del razonamiento ya que no se debería modificar nada del código de la aplicación.

Para este proyecto investigamos los mecanismos usados en los sistemas recomendadores de filtrado basado en contenido y colaborativo y los pusimos en práctica en la aplicación que desarrollamos. Cabe destacar que consideramos que el resultado final fue positivo ya que fue posible lograr lo que se pretendía y llegar a implementar una ontología que cumpla con lo que se esperaba. Además consideramos que si bien este es un proyecto de asignación de empleados de una empresa de servicios de Catering a los eventos que esta organiza, el módulo desarrollado podría parametrizarse y así poder emplearse en cualquier empresa que requiera llevar a cabo una asignación, de cualquier tipo de recurso ya sea humano o no a cualquier tipo de actividad donde especifiquen mínimamente las características deseadas que deben ser tenidas en cuenta.

El hecho de incorporar una ontología en un tipo de sistemas en el cual no se suelen usar con frecuencia, nos trajo problemas a la hora de adoptar una metodología tradicional, por lo cual decidimos tomar los pasos que nos parecieron adecuados sacándolos de procesos de desarrollo que sean para sistemas tradicionales tanto como para un desarrollo de un sistema completamente basado en ontologías. Sin embargo, si bien fue una dificultad, de esta manera pudimos poner en práctica todo tipo de conocimientos adquiridos durante el transcurso de nuestra carrera, por un lado proveniente de materias troncales que nos dan las herramientas para poder desarrollar un sistema y tener en cuenta aspectos generales como son la seguridad, la robustez, el manejo de excepciones, la eficiencia, entre otras cuestiones. Por otro lado la aplicación de conocimientos obtenidos de materias electivas, como el uso de herramientas particulares, como por ejemplo las estructuras de datos, las arquitecturas de software y las ontologías. Podemos afirmar entonces que este proyecto nos sirvió para aplicar en menor o mayor medida los conceptos que nos aportaron las materias de nuestra carrera.

BIBLIOGRAFÍA

- Gómez-Pérez, A., Fernandez-Lopez, M., Corcho, O. "Ontological Engineering". Editorial Springer, Heidelberg (2004).
- Weigand, H. Multilingual Ontology-Based Lexicon for News Filtering - The TREVI Project. 15th International Joint Conference on Artificial Intelligence (IJCAI). Agosto 23-29. (1997).
- Gruber, T. R. A Translation Approach to Portable Ontology Specifications. Journal Knowledge Acquisition, Volume 5 (Issue 2): 199-220 (1993).
- Maedche A. "Ontology Learning for the Semantic Web", Capítulo 2: "Ontology - Definition & Overview". Editorial Springer (2002).
- O'Connor, M. and Das, A. "SQWRL: a query language for OWL". Proc. 6th workshop OWL: experiences and directions (2009).
- Mochol, M.; Oldakowski, R. y Heese, R. "Ontology based Recruitment Process". GI Conference; Berlín, Alemania (2004).
- Popescu, M.; Popescu, E. "A Human Resource Ontology for Recruitment". University of Pitesti Romania, Faculty of Economic Sciences, RepublicII Avenue, No. 71, Pitesti, Arges (2010).
- Dorn, J.; Naz, T. y Pichhlmair, M. "Ontology Development for Human Resource Management". 4th International Conference on Knowledge Management, Institute of Information Systems 184/2, Technical University Vienna, Favoritenstraße 9-11, Vienna, A-1040, Austria (2007).
- Tim Berners-Lee, Dieter Fensel, Wolfgang Wahlster, Henry Lieberman, James Hendler. "Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential", MIT Press (2002).
- Malgorzata Mochol, Elena Paslaru y Bontas Simperl. "Practical Guidelines for Building Semantic eRecruitment Applications". Free University of Berlin Takustr. 9, 14195 Berlin, Alemania (2006).
- Middleton, Roure, Shadbolt. "Methods and Applications for Ontology-Based Recommender Systems" Handbook. Springer, Berlin, pág. 779-796 (2009).

- Hosein Jafarkarimi; A.T.H. Sim and R. Saadatdoost. “A Naïve Recommendation Model for Large Databases”, International Journal of Information and Education Technology vol. 2, no. 3, pp. 216-219 (2012).
- Francesco Ricci, Lior Rokach, Bracha Shapira. “Introduction to Recommender Systems Handbook” Capítulo 1. Editorial Springer (2011).
- John S. Breese, David Heckerman, Carl Kadie. “Empirical analysis of predictive algorithms for collaborative filtering”, 14th Conference on Uncertainty in Artificial Intelligence (1998).
- Mavetera, N. Towards a Comprehensive Ontology-Driven Software Development Approach. Innovation and Knowledge Management: A Global Competitive Advantage, 1112-1121, 16th IBIMA International conference (2011).
- Emiliano Reynares, Ma. Laura Caliusco, y Ma. Rosa Galli. EDON: A Method for Building an Ontology as Software Artefact (2012).

ANEXO I: HERRAMIENTAS UTILIZADAS

Exponemos en esta sección las tecnologías y herramientas elegidas para el desarrollo de la aplicación, describiendo brevemente las características principales de cada una de ellas.

Eclipse IDE for Java EE Developers:



Es un entorno de desarrollo integrado de código abierto para desarrollar aplicaciones JAVA que se ejecuten en un servidor web.

Apache Tomcat:

Apache Tomcat



Es un contenedor web con soporte de servlets y JSPs. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache. Tomcat puede funcionar como servidor web por sí mismo. Es usado hoy en día como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

MySQL Workbench:



Es una herramienta visual de diseño de bases de datos que integra desarrollo de software, administración de bases de datos, diseño de bases de datos, creación y mantenimiento para el sistema de base de datos MySQL.

MySQL es un sistema de gestión de bases de datos relacional, multi thread y multiusuario muy rápida en la lectura. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.

Protégé:



Es un editor de ontologías libre, de código abierto y un framework basado en conocimientos. Las Ontologías realizadas en Protégé pueden ser exportadas a una variedad de formatos, incluyendo RDF (S), OWL y XML Schema.

Protégé se basa en Java, es extensible, y proporciona un entorno de plug-and-play que lo hace una base flexible para creación rápida de prototipos y desarrollo de aplicaciones. La aplicación usa Swing para crear su compleja interfaz gráfica.

Jena:



Apache Jena es un framework de web semántica de código abierto para Java. Consiste de una API que permite extraer datos de gráficos RDF y escribir sobre estos. Estos gráficos son representados mediante un modelo abstracto. Este modelo puede obtenerse a partir de datos de un archivo, bases de datos, URLs o una combinación de estos. Sobre el modelo pueden realizarse consultas utilizando SPARQL.

Spring Framework:



Es un framework para el desarrollo de aplicaciones y contenedor de inversión de control, de código abierto para la plataforma Java. Si bien Spring comprende diversos módulos que proveen un rango de servicios, para este proyecto utilizamos principalmente el Contenedor de inversión de control que permite la configuración de los componentes de aplicación y la administración del ciclo de vida de los objetos Java, llevadas a cabo principalmente a través de la inyección de dependencias.

El contenedor de inversión de control (IoC) proporciona una forma consistente de configuración y administración de objetos Java usando la Reflexión. El Contenedor se encarga de gestionar los ciclos de vida de objetos de los objetos específicos: la creación de estos objetos, llamando a sus métodos de inicialización, y configurando estos objetos asociándolos.

Los objetos creados por el Contenedor también se denominan objetos gestionados o beans. El Contenedor se puede configurar mediante la carga de archivos XML o la detección de anotaciones Java específicas sobre la configuración de las clases. Estas fuentes de datos contienen las definiciones que proporcionan la información necesaria para la creación de las beans.

Los objetos pueden ser obtenidos por cualquiera de los medios de dependencia de búsqueda o dependencia de inyección. *Dependencia de búsqueda* es un modelo donde se pide al objeto contenedor un objeto con un nombre específico o de un tipo específico. *Dependencia de inyección* es un modelo en el que el contenedor pasa objetos por nombre a otros objetos, ya sea a través de métodos constructores, propiedades, o métodos de la fábrica.

En muchos casos cuando se utilizan otras partes del Spring Framework no necesita utilizar el Contenedor, aunque probablemente su uso permite hacer una aplicación más fácil de configurar y personalizar. El Contenedor de Spring proporciona un mecanismo consistente para configurar las aplicaciones, y se integra con casi todos los entornos Java, desde aplicaciones de pequeñas a grandes aplicaciones empresariales.

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 beans de las entidades que permiten establecer estas relaciones.

Hibernate es software libre, distribuido bajo los términos de la licencia GNU LGPL y busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional). Para lograr esto permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. Con esta información Hibernate le

permite a la aplicación manipular los datos en la base de datos operando sobre objetos, con todas las características de la programación orientada a objetos. Hibernate convertirá los datos entre los tipos utilizados por Java y los definidos por SQL. 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.

Hibernate está diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente. También tiene la funcionalidad de crear la base de datos a partir de la información disponible.

Hibernate ofrece también un lenguaje de consulta de datos llamado HQL (Hibernate Query Language), al mismo tiempo que una API para construir las consultas programáticamente (conocida como "criteria").

Hibernate para Java puede ser utilizado en aplicaciones Java independientes o en aplicaciones Java EE, mediante el componente Hibernate Annotations que implementa el estándar JPA, que es parte de esta plataforma.

Vaadin:



Es un framework de presentación de aplicaciones web de código abierto. A diferencia de las librerías JavaScript y complementos de navegadores web, posee una arquitectura de lado de servidor, es decir que la mayor parte de la lógica se ejecuta en los servidores. Del lado del navegador web se utiliza tecnología Ajax para lograr una experiencia interactiva con el usuario. Del lado del cliente, Vaadin está construido sobre, y puede ser extendido con, Google Web Toolkit.

Vaadin utiliza Java como lenguaje de programación para crear contenido web. Incorpora programación orientada a eventos y widgets, que permiten desarrollar un modelo de

programación que se asemeja más al desarrollo de interfaces gráficas de usuario que al desarrollo web tradicional que utiliza HTML y JavaScript.

XAMPP:



XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl.

El programa está liberado bajo la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas.