



Facultad 5

*Trabajo de diploma para optar por el título
de Ingeniero en Ciencias Informáticas*

*Título: Capa de acceso a datos a través del Mapeo Relacional de
Objetos Difusos en la herramienta para la evaluación de competencias*

ANAELIX

Autora: Aylín Rodríguez Pérez

Tutor: Ing. Luis Manuel Teijón Acosta

Co-Tutores: MsC. Manuel Villanueva Betancourt

Ing. Lorena Pita Pérez

“Año 55 de la Revolución”

La Habana, Junio de 2014

“(…) Quien pretenda en las aulas universitarias vivir solo con el oxígeno que en ellas
se respira, morirá a causa de una asfixia cultural (…)”

J. F. Bulté

Capa de acceso a datos a través del Mapeo Relacional de Objetos Difusos en la herramienta para la evaluación de competencias ANAELIX

DECLARACIÓN DE AUTORÍA

Declaro ser autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los __ días del mes de _____ del año _____.

Firma de la Autora

Aylín Rodríguez Pérez

Firma del Tutor

Ing. Luis M. Teijón Acosta.

Firma del Co-Tutor

Msc. Manuel Villanueva Betancourt

Firma de la Co-Tutora

Ing. Lorena Pita Pérez

Capa de acceso a datos a través del Mapeo Relacional de Objetos Difusos en la herramienta para la evaluación de competencias ANAELIX

Autor: Aylín Rodríguez Pérez

Correo: arodriguezp@estudiantes.uci.cu

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Tutor: Ing. Luis Manuel Teijón Acosta

Correo: lmteijon@uci.cu

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Co-Tutor: MsC. Manuel Villanueva Betancourt

Correo: manuelvb@uci.cu

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Co-Tutora: Ing. Lorena Pita Pérez

Correo: lpita@uci.cu

Universidad de las Ciencias Informáticas, La Habana, Cuba.

A nuestra invicta Revolución, por darme la oportunidad de estudiar en esta escuela y formarme como Ingeniera en Ciencias Informáticas.

A mis padres por ser lo más grande que tengo en este mundo, que han sabido guiarme por el camino correcto, estando siempre a mi lado apoyándome tanto en los momentos de alegría como en los de tristeza para lograr mis objetivos.

A mi hermana que la quiero muchísimo.

A mi abuela porque ha sabido aconsejarme y ayudarme en todos los momentos y me siento orgullosa de contar con ella para hacerle este regalo.

A mi tía Betty y mis tíos Juanmami y Pepe que me han ayudado de forma incondicional como si fueran mis padres.

A una persona muy especial para mí, Dairon por recibir todo su apoyo y disfrutar de su compañía en todos los momentos, por darme fuerzas para seguir adelante.

A mis primos Yosep, Arianna, Andy y Jenny por llenar de felicidad mis días.

A todos los miembros de mi familia que siempre han confiado en mí y me han deseado lo mejor.

A mi amiga Yure de tantos años, por apoyarme siempre en los momentos más difíciles y por enseñarme a luchar por las cosas que se quieren en la vida y por sus consejos en cada momento que los necesité.

A mis amigas Elizabeth, Anayanci y Ailen que durante 5 años hemos compartido juntas todos los momentos de vida universitaria y han sido incondicionales conmigo.

A mi amigo Reydel que desde la secundaria venimos juntos y nunca me ha olvidado.

A Ernesto, Frank y David por ser los mis compañeros consejeros de estos 5 años, y a todos los demás del grupo 5505, el más unido y divertido de la UCI y a mis compañeras de apartamento que hemos convivido juntas también estos 5 años, en fin a todos los amigos y amigas que he conocido en esta universidad, a todos gracias por haber compartido una parte de sus vida conmigo, agradezco a todos su aprecio y confianza.

A mis tutores por permitirme llegar hasta aquí, especialmente al profe Teijón por dedicarme pequeños espacios de su codiciado tiempo.

Agradezco también a Oscar que a pesar de no ser mi tutor me ayudó muchísimo, a Leitniz por sus sugerencias y por ser el mejor diseñador de invitaciones de la UCI y a todos los del apartamento, a Prevot que me ha brindado siempre su ayuda.

A mis suegros y otros miembros de la familia por acogerme como uno más de ellos.

A todos los miembros del tribunal por todos sus consejos, logrando de esta forma una mayor calidad del trabajo de diploma.

A todos aquellos que hicieron posible que se realizara mi sueño muchas gracias.

A toda mi familia por apoyarme y por haber confiado en mí siempre.

En especial a mis padres, mi hermana y mi abuela, porque son ellos los que han estado a mi lado, desde que soy muy pequeña añorando cumplir este gran sueño y deseándome siempre lo mejor.

Resumen

El mantenimiento de software es una actividad muy amplia que incluye la corrección de errores, mejoras de las capacidades, eliminación de funciones obsoletas y optimización. Debido a que el cambio es inevitable, es importante desarrollar mecanismos que faciliten la evolución del software. El presente trabajo se centra en el desarrollo de uno de estos mecanismos para facilitar el acceso a una base de datos difusa existente, la cual forma parte de una herramienta, para la selección de recursos humanos a través del procesamiento difuso de los datos, llamado ANAELIX. El mantenimiento de la herramienta se ve actualmente limitado debido a la complejidad del lenguaje de consultas difusas utilizado, el cual es una extensión del lenguaje estándar de consultas. Este detalle obliga a cualquier equipo de mantenimiento a dominar este complejo lenguaje de consultas para poder brindar mantenimiento correctivo y perfectivo al sistema. El propósito de la investigación es crear una capa de acceso, con el objetivo de abstraer a los desarrolladores y al equipo de mantenimiento, del acceso directo a la base de datos difusa, a través del mapeo relacional de objetos difusos con el objetivo de facilitar el proceso de desarrollo actual y futuro de la herramienta ANAELIX.

Palabras Clave: base de datos difusa, capa de acceso a datos, mapeo relacional de objetos difusos.

ÍNDICE

INTRODUCCIÓN.....	12
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	16
1.1 Conceptos asociados al dominio del problema.....	16
1.1.1 Base de datos Difusa	16
1.1.2 Interfaz Difusa	17
1.1.3 Atributo Difuso (<i>Fuzzy</i>).....	18
1.1.4 Capa de Acceso a datos	20
1.1.5 <i>Framework</i>	20
1.2 Estado del arte de los sistemas evaluadores de competencias.....	21
1.2.1 Talent Test v3.0.	21
1.2.2 RHPlan	22
1.2.3 Teamsoft.....	22
1.3 Lenguajes, herramientas, metodología y tecnologías utilizadas.....	23
1.3.1 Metodologías de desarrollo de software.....	23
1.3.2 Lenguaje de Modelado.....	25
1.3.3 Herramientas CASE. Visual Paradigm.....	25
1.3.4 Lenguajes de programación. Java.....	26
1.3.5 Entorno de desarrollo integrado.....	26
1.3.6 <i>Frameworks</i> para el mapeo de objetos en el lenguaje Java.....	28
1.3.7 Gestor de base de datos. Oracle 10g.....	30
1.4 Consideraciones del capítulo	30
CAPÍTULO 2: SOLUCIÓN PROPUESTA.....	32
2.1 Descripción de la solución.....	32
2.2 Fase de Planificación.....	33
2.2.1 Lista de reserva del producto.....	34
2.2.2 Historias de Usuario.....	36
2.2.3 Estimación de esfuerzos por historias de usuario.....	40

Capa de acceso a datos a través del Mapeo Relacional de Objetos Difusos en la herramienta para la evaluación de competencias ANAELIX

2.2.4 Plan de duración de las iteraciones.....	41
2.3 Atributos Difusos existentes en la base de datos.	43
2.4 Fase de Diseño.....	46
2.4.1 Tarjetas Contenido, Responsabilidad, Colaboración.....	46
2.4.2 Modelo de Datos.....	47
2.4.3 Patrón de Diseño.	48
2.4.4 Diagrama de los paquetes de la capa de acceso a datos.....	49
2.5 Fase de Implementación.....	50
2.5.1 Tareas de Programación.....	51
2.6 Consulta difusa.	54
2.7 Consideraciones del capítulo	55
CAPÍTULO 3: VALIDACIÓN Y PRUEBAS DEL DISEÑO.....	57
3. Fase de Prueba.....	57
3.1 Validación teórica del diseño.....	57
3.1.1 Integridad de los datos.....	57
3.2 Pruebas de aceptación.....	59
3.2.1 Resultados esperados.....	70
3.3 Consideraciones del capítulo	71
CONCLUSIONES GENERALES	72
RECOMENDACIONES	73
BIBLIOGRAFÍA.....	74
GLOSARIO DE TÉRMINOS	77
ANEXOS.....	78

ÍNDICE DE FIGURAS

Figura 1 Solución Propuesta.....	33
Figura 2 Diagrama de paquetes de la capa de acceso a datos.....	50
Figura 3 Resultado de la ejecución de una consulta difusa.....	55

ÍNDICE DE TABLAS

Tabla 1 Requisitos funcionales.	34
Tabla 2 Requisitos no funcionales	36
Tabla 3 HU Traducir consulta FSQL a SQL	37
Tabla 4 HU Insertar Competencia.....	38
Tabla 5 HU Modificar Competencia	38
Tabla 6 HU Eliminar Competencia.....	39
Tabla 7 Estimación de esfuerzos por historias de usuario	40
Tabla 8 Plan de duración de las iteraciones.....	42
Tabla 9 Representación interna de atributos difusos Tipo 2.....	44
Tabla 10 Atributos difusos de la tabla “evaluator”	44
Tabla 11 Atributos difusos de la tabla “requirement”	45
Tabla 12 Atributos difusos de la tabla “similarity_degree”	45
Tabla 13 Atributos difusos de la tabla “evaluation”.....	45
Tabla 14 Tarjeta CRC GenericDAO	46
Tabla 15 Tarjeta CRC JdbcQueryManager	47
Tabla 16 Tareas de Programación.....	51
Tabla 17 Análisis de integridad de entidad.....	58
Tabla 18 Caso de Prueba para la tabla <i>Dimension</i>	60
Tabla 19 Caso de Prueba para la tabla <i>Dimension</i>	61
Tabla 20 Caso de Prueba para la Tabla <i>Competence</i>	62
Tabla 21 Caso de Prueba para la Tabla <i>Competence</i>	63
Tabla 22 Caso de Prueba para la Tabla <i>Evaluator</i>	65
Tabla 23 Caso de Prueba para la Tabla <i>Evaluator</i>	66
Tabla 24 Caso de Prueba para la Ejecución de una consulta difusa.....	68
Tabla 25 Caso de Prueba para la ejecución de una consulta difusa.	69

INTRODUCCIÓN

La evaluación de competencias es un proceso de retroalimentación, determinación de idoneidad y certificación de los aprendizajes, mediante el análisis del desempeño de las personas en tareas y problemas pertinentes. Esto implica importantes cambios en la evaluación tradicional, pues en este nuevo enfoque de evaluación los evaluados deben tener mucha claridad del para qué, para quién, por qué y cómo es la evaluación, de lo contrario esta no va a tener la significación necesaria para contribuir a formar profesionales idóneos. Es así como la evaluación debe plantearse mediante tareas y problemas lo más reales posibles que impliquen curiosidad y reto. (1)

El desarrollo del talento en las instituciones docentes cubanas es un reto que asume la educación contemporánea y constituye una necesidad para enfrentar y aumentar el desarrollo de la ciencia y la técnica en la sociedad actual. Como una de las respuestas a esta necesidad se crea en la Universidad de las Ciencias Informáticas el proyecto investigativo de innovación pedagógica Talenmático, que trabaja en un modelo de atención educativa a los estudiantes potencialmente talentosos, y para lo cual la identificación de potencialidades constituye un aspecto decisivo.

En el año 2012 se comenzó a desarrollar un software para facilitar el proceso de gestión de las competencias y habilidades de los estudiantes con el fin de descubrir las fortalezas y debilidades de cada uno. Este sistema serviría de apoyo durante el proceso de selección de personal para los proyectos de desarrollo de software de la propia universidad y poder así lograr un mejor aprovechamiento de cada uno de ellos. (2) Como complemento a este software se desarrolló una aplicación llamada ANAELIX, que implementa un modelo matemático para la selección de recursos humanos en proyectos de software, y que incorporó un grupo de nuevas funcionalidades.

La aplicación utiliza una Base de Datos Difusa (BDD) (3) para el manejo de la incertidumbre asociada a los procesos de evaluación y selección del personal del proyecto. La incertidumbre aparece principalmente como resultado de la evaluación de competencias cualitativas, cuya evaluación posee cierto grado de subjetividad o imprecisión.

Actualmente el acceso a la base de datos difusa se realiza utilizando el Lenguaje de Consultas Difusas (FSQL, por sus siglas en inglés), el cual es una extensión del Lenguaje Estructurado de Consultas (SQL, por sus siglas en inglés), con la finalidad de flexibilizar el proceso de selección a través del uso de operadores difusos. Para acceder a la base de datos difusa se requiere de un servidor difuso, que interpreta las sentencias difusas y las traduce al SQL. Cada sentencia FSQL puede ser escrita usando solamente SQL, pues muchas de las extensiones “difusas” de este nuevo lenguaje son funciones implementadas en SQL, pero escribir las consultas difusas en SQL implica conocimientos profundos de FSQL que permitan realizar la traducción. Como resultado se tiene una aplicación altamente dependiente a la base de datos que utiliza.

El uso del lenguaje FSQL ocasiona serios problemas de mantenimiento del software, toda vez que el personal encargado de dar el mantenimiento debe poseer conocimientos de FSQL para poder realizar mantenimiento correctivo o perfectivo que impliquen acceso a datos. Estos problemas se evidenciaron durante el desarrollo de una primera versión de la herramienta, en la cual los cambios realizados en el acceso a datos requerían cambios en el sistema implicando una pérdida de tiempo significativa. Por lo que se crean riesgos de gran impacto, ya que el software podría no evolucionar y quedar obsoleto.

Por tanto, la **situación problemática** es la siguiente:

- Dependencia directa entre la base de datos difusa y la herramienta ANAELIX.
- Elevada complejidad en la sintaxis de FSQL para el desarrollo y mantenimiento del sistema.
- Problemas de mantenimiento del software.
- Riesgos de gran impacto, ya que el software podría no evolucionar y quedar obsoleto en poco tiempo.

Luego, el **problema de investigación** se formula de la siguiente forma: ¿Cómo eliminar la dependencia directa entre el sistema ANAELIX y la base de datos difusa que este utiliza, con el fin de facilitar un mantenimiento continuo a la herramienta?

La presente investigación tiene como **objeto de estudio**: Capas de acceso a las bases de datos.

Se define como **objetivo de la investigación**: Desarrollar una capa de acceso a datos, a través del mapeo relacional de objetos difusos para la herramienta ANAELIX, con el fin de eliminar la dependencia directa entre la base de datos y la herramienta, facilitando un mantenimiento futuro a la misma.

A partir de todo lo anterior el **campo de acción** se enmarca: Capas de acceso a datos a las bases de datos difusas.

Para cumplir el objetivo de la investigación y darle solución al problema se realizan las siguientes **tareas de la investigación**:

- Elaboración del marco teórico de la investigación, a partir del análisis del estado del arte del tema.
- Realización de estudios sobre Herramientas para el Mapeo Relacional de Objetos (ORM, por sus siglas en inglés).
- Selección de metodologías y herramientas a utilizar para la elaboración de la capa de acceso a datos.
- Realización del levantamiento de requisitos de la capa de acceso a datos.
- Diseño de la capa de acceso a datos con una herramienta ORM.
- Pruebas del funcionamiento de la capa de acceso a datos.

Entre los **métodos teóricos** utilizados en esta investigación se encuentran:

- **Histórico-lógico**, se utiliza para el estudio de los distintos sistemas encargados de la evaluación de competencias y las herramientas para las capas de acceso a datos, desde su desarrollo cronológico, dando la posibilidad de analizar la trayectoria histórica de las mismas, así como la evolución y desarrollo de sus funcionalidades.
- **Análisis-Síntesis**: se requiere para el estudio de la información sobre BDD y el acceso a datos, que permita su desglose progresivo y el resumen para arribar a ideas generales sobre la capa de acceso para la base de datos difusa del sistema de evaluación por competencias.

Como **métodos empíricos** se utilizaron:

Capa de acceso a datos a través del Mapeo Relacional de Objetos Difusos en la herramienta para la evaluación de competencias ANAELIX

- **Observación:** utilizado como medio para la adquisición de conocimientos. En los intercambios realizados con el personal del CDAE (Centro de Consultoría y Desarrollo de Arquitecturas Empresariales), mediante este método se aprendió una forma más fácil de utilizar el *framework* seleccionado, tomando de éstos algunas buenas prácticas para el posterior desempeño de la capa de acceso a datos.
- **Pruebas de validación:** se emplearon para validar las funcionalidades de la capa de acceso a datos.

El presente trabajo se estructuró en los siguientes capítulos:

Capítulo 1: Fundamentación teórica.

En este capítulo se realiza un estudio del estado del arte de los sistemas de evaluación de competencias y el uso de capas de acceso a datos. Además, se tratan todos los términos que dan lugar a la investigación, así como las metodologías, tecnologías y herramientas que serán utilizadas.

Capítulo 2: Descripción de la solución Propuesta.

En el capítulo se identifican las características de la capa de acceso a datos del sistema ANAELIX. Se describe la solución que se propone al problema planteado, para ello se definen los requisitos funcionales y no funcionales, se muestra el modelo de datos, el patrón de acceso a datos utilizado, así como las historias de usuarios y el plan de iteraciones.

Capítulo 3: Validación y pruebas del diseño.

En este último capítulo se tratan las pruebas a aplicar para verificar el correcto funcionamiento de la capa de acceso a datos, para ello se realiza una validación funcional de la solución propuesta. Se exponen aspectos como la integridad de dominio, referencial y de entidad. Además, se realizan también las pruebas de aceptación para hacer corresponder las funcionalidades de la capa con los requerimientos definidos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

Introducción

En este capítulo se abordarán los principales conceptos que contribuirán a una mejor comprensión de los términos que se emplearán durante la investigación. Se lleva a cabo una caracterización del objeto de estudio, realizando un bosquejo acerca de las tendencias actuales de las capas de acceso a base de datos difusas. Además, se describen las herramientas, tecnologías y la metodología utilizada para dar solución al problema, con el fin de justificar su empleo en el desarrollo de la capa de acceso a datos.

1.1 Conceptos asociados al dominio del problema.

Con el objetivo de lograr una mayor comprensión de los términos abordados en el documento, se dan a conocer un conjunto de conceptos tratados durante la investigación, relacionados con el tema de capa de acceso a datos.

1.1.1 Base de datos Difusa

El término "Bases de Datos Difusas" designa al área de investigación que trata sobre la aplicación de los Conjuntos Difusos y la Lógica Difusa en Bases de Datos Relacionales. Esta área busca compensar la deficiencia de los Sistemas de Bases de Datos clásicos en representar y manipular información imprecisa y consultas flexibles o graduales. A este problema se le conoce como el problema de rigidez de las Bases de Datos convencionales, así que los Sistemas de Bases de Datos que tratan de resolver este problema se califican como "Flexibles". (4)

La base de datos difusa permite la representación de dos tipos fundamentales de datos según (5): **datos precisos**, que son datos libres de incertidumbre, como valores numéricos, fechas, horas, entre otros.; y **datos imprecisos**, los cuales contienen información vaga, difusa o borrosa. Los datos imprecisos, según su naturaleza, pueden ser representados como una función matemática o distribución de posibilidad trapezoidal. Estas funciones pueden asociarse a etiquetas lingüísticas para facilitar su uso. También pueden usarse otro tipo de funciones, tales como, valores aproximados, intervalos de posibilidad, o valores clásicos o precisos. (3)

Capa de acceso a datos a través del Mapeo Relacional de Objetos Difusos en la herramienta para la evaluación de competencias ANAELIX

La base de datos maneja una serie de conceptos que deben ser explicados con el objetivo de poder entender más adelante el tratamiento que recibe cada uno de ellos durante el mapeo relacional difuso. A continuación se explican algunos que se consideraron importantes.

1.1.2 Interfaz Difusa

El Dr. Medina en 1994 expuso un módulo para permitir extender la capacidad de un SGBD (Sistema Gestor de Base de Datos) clásico para que pueda representar y manipular información “imprecisa”. Este módulo, llamado Interfaz Difusa para Sistemas Relacionales (FIRST, por sus siglas en inglés), utiliza un modelo Generalizado de BD Relacionales Difusas (GEFRED, por sus siglas en inglés) como modelo teórico. (6)

Fundamentos de FIRST

- **Base de Metaconocimiento Difuso (FMB, por sus siglas en inglés):** La FMB permite recoger aquella información necesaria relacionada con la naturaleza “imprecisa” de la información a procesar. La FMB se puede describir como el catálogo del sistema del cual se servirá la base de datos difusa para procesar información imprecisa. Esta guarda la información asociada a cada atributo difuso, etiqueta lingüística, trapecio probabilístico y metadatos para los operadores difusos. (7). Algunas de las tablas que se mencionan a continuación forman parte también de la FMB: **FUZZY_COL_LIST:** Contiene la descripción de los atributos difusos, **FUZZY_LABEL_DEF:** Contiene la definición de etiquetas trapezoidales, **FUZZY_APPROX_MUCH:** Contiene los valores para el margen (MARGEN) y el valor mínimo o distancia mínima **M** para considerar dos valores muy separados (MUCH) para atributos difusos Tipo 1 ó 2, **FUZZY_OBJECT_LIST:** Contiene la descripción de los objetos difusos definidos para cada atributo (OBJ#,COL#).
- **Servidor FSQL:** Su objetivo es captar las sentencias difusas y traducirlas a un lenguaje que entienda el SGBD, el lenguaje SQL. Para efectuar esta traducción utilizará la información almacenada en la FMB (8) .
- **Cliente FSQL:** Se trata de un programa que hace de interfaz entre el hombre (u otro programa) y el Servidor FSQL. Este programa puede ser muy simple, pues el

trabajo principal de una operación FSQL será efectuado por el Servidor FSQL. El programa Cliente FSQL puede ser programado para cualquier Sistema Operativo y en cualquier lenguaje de programación (8).

1.1.3 Atributo Difuso (*Fuzzy*)

Los atributos difusos se consideran datos con dominio de referencial ordenado y no ordenado. También son considerados en ambos tipos de dominio los valores *Unknown* (valor desconocido), *Undefined* (valor no definido o sin sentido) y *Null* (ignorancia total, no se sabe si es *Unknown* o *Undefined*). La clasificación adoptada se basa en criterios de representación y de tratamiento de los datos “imprecisos”: se clasifica según el tipo de dominio que les subyace y por si permiten representar la información imprecisa o sólo permiten el tratamiento impreciso de datos sin imprecisión (6).

Los atributos difusos pueden ser de 3 tipos (8):

- **Tipo 1 (*crisp*):** Estos son atributos con “datos precisos”, clásicos o *crisp* (tradicionales, sin imprecisión), que pueden tener etiquetas lingüísticas definidas sobre sus dominios. Estos tipos de atributos reciben una representación igual que los datos precisos, pero admiten que se puedan utilizar en consultas difusas, utilizando o no las etiquetas definidas en su dominio. (8)
- **Tipo 2 (*probabilísticos*):** Admiten datos tradicionales y difusos como distribuciones de probabilidad sobre un dominio ordenado. Los valores de este tipo de atributo difuso pueden ser las propias etiquetas lingüísticas, distribuciones de posibilidad trapezoidales, entre otras. (8). Para concluir se puede decir que son atributos similares a los de Tipo 1, pero que además permiten el almacenamiento de información difusa o borrosa.

Para un atributo difuso Tipo 2 llamado F, la representación usa 5 atributos: FT para almacenar el código de tipo que le corresponde a cada valor y los atributos F1, F2, F3 y F4 para almacenar los parámetros de cada dato. Esta representación implica que la forma de crear, actualizar o eliminar un atributo difuso requiera un conjunto de pasos adicionales que se tuvieron en cuenta para el mapeo relacional difuso, y que será explicado más adelante. Los valores NULL que aparecen en los atributos tienen el significado de valor

“no-aplicable” en el Sistema de Gestión de Base de Datos Relacional (SGBDR, por sus siglas en inglés) anfitrión. (8)

- **FT:** Almacena el tipo de valor que corresponde al dato que se quiere almacenar, indicando su representación. Según lo visto, puede ser: *UNKNOWN* (0), *UNDEFINED* (1), *NULL* (2), *CRISP* (3), *LABEL* (4), *INTERVALO* (5), *APROXIMADAMENTE* (6) o *TRAPEZOIDAL* (7).
- **F1, F2, F3 y F4:** Los atributos cuyo nombre se forma añadiendo los números 1, 2, 3 y 4 al nombre del atributo almacenan la descripción de los parámetros que definen el dato y que depende del tipo de valor (FT) al que pertenezca:
 - ***UNKNOWN, UNDEFINED, NULL:*** Estos 3 valores no necesitan ningún parámetro, por lo que todos ellos permanecen a *NULL* (entendiendo este valor como el *NULL* del SGBD anfitrión y no como el *NULL* del valor difuso, que no deben confundirse).
 - ***CRISP:*** Un valor de tipo *crisp*, necesita tan solo un parámetro, F1, en el cual se almacenará el valor *crisp* en cuestión.
 - ***LABEL:*** Igualmente, un valor de tipo etiqueta solo necesita un parámetro para almacenar el identificador asociado a dicha etiqueta (*FUZZY_ID*). Ese indicador es útil para poder acceder a la FMB y obtener la descripción asociada a esta etiqueta.
 - ***INTERVALO:*** Necesita los dos valores extremos del intervalo $[n, m]$, que son almacenados en F1 y F4 respectivamente.
 - ***APROXIMADAMENTE:*** Este valor solo necesita un valor que se almacena en F1 y que es el valor central de la distribución de posibilidad triangular, d . Sin embargo, para reducir operaciones (tanto matemáticas como de acceso a datos), se aprovechan los atributos F2, F3 y F4 para almacenar los valores siguientes: d -margen, d +margen y margen, respectivamente. El valor margen es un valor almacenado en la FMB para cada atributo difuso, y su valor depende del significado de dicho atributo. Esto permite la posibilidad de almacenar valores aproximados sin indicar el margen, usando el margen por

defecto almacenado en la FMB, o también almacenar valores aproximados con un margen particular para ellos, distinto al de la FMB.

- **TRAPECIO:** Necesita almacenar forzosamente los 4 valores que identifican a un trapecio: $[\alpha, \beta, \gamma, \delta]$. En F2 y F3 se almacenan unas operaciones que simplifican las ecuaciones cuando se opera con este tipo de dato. (8)
- **Tipo 3 (escalares):** Datos definidos sobre un dominio discreto no ordenado con analogía. En estos atributos se definen algunas etiquetas (“rubio”, “moreno”, “pelirrojo”, entre otros.) que son escalares con una relación de similitud (proximidad, μ) definida sobre ellas, de forma que esta relación indique en qué medida se parecen entre sí cada par de etiquetas, ejemplo: μ (rubio, pelirrojo)=0.8, μ (rubio, moreno)=0. (9)

1.1.4 Capa de Acceso a datos

Una capa de acceso a datos es una capa de abstracción que permite ocultar los detalles de implementación de ciertas funcionalidades. La capa de acceso a datos incluye el código que crea una conexión a la base de datos y que emite los comandos *Select*, *Insert*, *Update* y *Delete*. (10)

1.1.5 Framework

Un *framework* o marco de trabajo es una colección organizada de clases que constituyen un diseño reutilizable para un dominio específico de software. Contiene un conjunto de bibliotecas, componentes de software y directrices arquitectónicas que ofrece al desarrollador herramientas para construir una aplicación de principio a fin, siempre teniendo en cuenta que es necesario adaptarlo a cada aplicación en particular.

El objetivo de los marcos de trabajo es proporcionar una estructura común, de modo que los desarrolladores no tienen que hacer el código de cero, ya que pueden volver a utilizarlo. (11)

Algunos de los *frameworks* más utilizados en la actualidad son:

Hibernate es una herramienta de mapeo relacional, de código abierto, creado por un grupo de desarrolladores dirigido por Gavin King a finales de 2001. Este *framework* emplea atributos declarativos mediante el Lenguaje de Marcas Extensibles (XML, por sus siglas en inglés) o anotaciones. Permite al desarrollador utilizar objetos persistentes en lugar de manipular directamente los datos de la BD. (12)

Doctrine es un proyecto independiente, con la especialidad de que posee su propio lenguaje para el acceso a datos, llamado *Doctrine Query Language*. Está por encima de la capa de abstracción a la base de datos, una de sus características es la posibilidad de escribir consultas a la base de datos a partir del tratamiento con objetos. (12)

iBATIS basa su funcionamiento en el mapeo de sentencias SQL que se incluyen en ficheros XML. Tiene herramientas para generar consultas dinámicas. (12)

1.2 Estado del arte de los sistemas evaluadores de competencias.

En la actualidad son muchas las empresas que se apoyan en sistemas de evaluación por competencias para asignar puestos de trabajos, con el propósito de evaluar y acreditar las destrezas que poseen los trabajadores. Ello permite que todos puedan realizar con eficiencia y calidad, las funciones laborales asignadas por la empresa y por tanto a nivel de país.

A continuación se muestran algunos sistemas evaluadores de competencias que existen en el mundo:

1.2.1 Talent Test v3.0.

Evalúa de forma integral a cualquier candidato, independientemente de su nivel jerárquico, ya que es capaz de evaluar desde niveles directivos, niveles gerenciales, jefaturas, puestos especializados, puestos semiespecializados, oficios, puestos administrativos y puestos operativos, esto permite tener una evaluación consistente y justa en todos los niveles.

La plataforma *desktop* consiste en adquirir el software para ser instalado en una PC, de esta forma, el candidato es evaluado de manera local, es decir, el candidato físicamente

contestará las pruebas, una vez que el evaluado termine de contestar se contará con los resultados. El sistema instala y utiliza la base de datos My SQL. (13)

1.2.2 RHPlan

RHPlan es una aplicación de escritorio, de Gestión de Proyectos, que se especializa fundamentalmente en el área de Gestión de Recursos Humanos, permitiendo gestionar proyectos, tareas y personal de un proyecto. Permite la gestión por competencias del personal así como la asignación de personal a proyectos. Garantiza la comunicación entre los miembros del proyecto a través del correo electrónico. (14)

Esta herramienta permite la toma de decisiones en la empresa y ayuda a la asignación de recursos humanos a un proyecto. Realiza además un seguimiento del proyecto, asignando tareas y recursos humanos a los mismos. Sin embargo, aísla la gestión de recursos humanos del resto de las áreas del conocimiento de la Gestión de Proyectos. (15)

1.2.3 Teamsoft

Es un sistema cliente servidor. Está compuesto por una aplicación de escritorio para el registro del desempeño personal de cada uno de los miembros del equipo y una aplicación *web* para la gestión del trabajo en equipo. (16)

La herramienta se inserta en el grupo de herramientas de Gestión de proyectos que implementa los procesos más importantes del área de Gestión de Recursos Humanos en su versión dos.

El uso de Java propició la utilización de un conjunto de tecnologías que surgieron a raíz del desarrollo de este como lenguaje de programación orientado a objetos de propósito general. Entre estas tecnologías se encuentran Hibernate, herramienta que garantiza el mapeo objeto/relacional en el entorno Java. (15)

Después del análisis realizado se ha concluido que de Teamsoft se pueden tomar experiencias para el mapeo de objeto relacional, pues es el único sistema, de los estudiados, que implementa una capa de abstracción entre la base de datos y la aplicación, utilizando para ello el *framework* Hibernate.

Capa de acceso a datos a través del Mapeo Relacional de Objetos Difusos en la herramienta para la evaluación de competencias ANAELIX

De los sistemas mencionados, no se encontró ninguno que tuviera implementada una capa de acceso a datos difusos, pues ninguno trabaja con bases de datos difusas. Sin embargo, se estudió un *framework* que sí permite realizar consultas difusas. Este es el caso de FJDBC, que es una ampliación de JDBC (*Java Database Connectivity* por sus siglas en inglés) para permitir realizar consultas difusas desde cualquier aplicación Java. El *framework* se puede utilizar principalmente para evaluar migraciones a otros modelos de implementación de bases de datos difusas. (17)

1.3 Lenguajes, herramientas, metodología y tecnologías utilizadas.

Existen herramientas que facilitan el manejo de bases de datos, diagramas que permiten visualizar la estructura de un producto, metodologías que definen la guía para el desarrollo de un software y tecnologías que constituyen la base para la creación de un sistema.

1.3.1 Metodologías de desarrollo de software.

Las metodologías de desarrollo se pueden enmarcar en dos grandes grupos, las llamadas metodologías tradicionales y las metodologías ágiles. Las tradicionales se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, las herramientas y notaciones que se usarán; son recomendadas para los proyectos de grandes dimensiones y con grandes equipos de desarrollo. En tanto, las metodologías ágiles dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas, respondiendo además a los cambios que puedan surgir a lo largo del desarrollo del proyecto. Su selección depende de qué producto se desee desarrollar, de las dimensiones que tendrá el mismo, del tiempo que se disponga y del equipo de trabajo, entre otros factores. (18)

Proceso Unificado de Desarrollo.

El Proceso Unificado de Desarrollo (RUP, por sus siglas en inglés) es una metodología de desarrollo para proyectos grandes, aunque también es usada en proyectos pequeños. RUP se divide en cuatro fases de desarrollo (Inicio o Concepción, Elaboración, Construcción, Transición).

Los procesos de RUP estiman tareas y horarios del plan midiendo la velocidad de iteraciones concerniente a sus estimaciones originales. Las iteraciones tempranas de proyectos conducidas por RUP se enfocan fuertemente sobre la arquitectura del software; la puesta en práctica rápida de características se retrasa hasta que se ha identificado y se ha probado una arquitectura firme. (19)

RUP permite asignar tareas y responsabilidades de forma disciplinada en una empresa de desarrollo (quién hace qué, cuándo y cómo). Un cambio en las etapas de vida del sistema incrementaría notablemente el costo. RUP es un marco del proyecto que describe una clase de los procesos que son iterativos e incrementales. (19)

Programación Extrema.

La metodología Programación Extrema (XP, por sus siglas en inglés). Nace en busca de simplificar el desarrollo del software y reducir el costo del proyecto. Método ligero: No produce demasiado tiempo sobre las actividades de desarrollo, y no impide el avance de los proyectos. Reduce el costo del cambio en las etapas de vida del sistema. Se requiere un grupo pequeño de programadores para trabajar con esta metodología entre 2 y 15 personas, éstas irán aumentando conforme sea necesario. (19)

Sus programadores pueden ser ordinarios. Combina las que han demostrado ser las mejores prácticas de desarrollo de software, y las lleva al extremo. Se rediseñará todo el tiempo (*refactoring*), dejando el código siempre en el estado más simple posible. Se harán pruebas todo el tiempo, no sólo de cada nueva clase (pruebas unitarias), sino que también los clientes comprobarán que el proyecto va satisfaciendo los requisitos (pruebas funcionales). Las pruebas de integración se efectuarán siempre, antes de añadir cualquier nueva clase al proyecto, o después de modificar cualquiera existente (integración continua), utilizando marcos de trabajo de *testing*, como el xUnit. (19)

Selección de la metodología de desarrollo a utilizar.

Para el desarrollo de la capa de acceso a datos propuesta se selecciona como metodología de desarrollo XP, por ser una metodología ágil, es decir de corta duración, diseñada para equipos de trabajos pequeños, centrada en vincular al cliente en el ciclo de desarrollo. Además XP presenta propiedades flexibles que aumentan la productividad y la

resistencia al cambio de requerimientos. Adaptándolo a las necesidades reales del desarrollo en proceso, pues se toma como el rol de cliente al mismo equipo de desarrollo.

1.3.2 Lenguaje de Modelado.

El Lenguaje de Modelado Unificado (UML, por sus siglas en inglés) es la sucesión de una serie de métodos de análisis y diseño orientadas a objetos que aparecen a fines de los 80's y principios de los 90s. **(20)**

UML incrementa la capacidad de lo que se puede hacer con otros métodos de análisis y diseño orientados a objetos. Los autores de UML apuntaron también al modelado de sistemas distribuidos y concurrentes para asegurar que el lenguaje maneje adecuadamente estos dominios. **(20)**

El lenguaje de modelado es la notación (principalmente gráfica) que usan los métodos para expresar un diseño. El proceso indica los pasos que se deben seguir para llegar a un diseño. La estandarización de un lenguaje de modelado es invaluable, ya que es la parte principal del proceso de comunicación que requieren todos los agentes involucrados en un proyecto informático. Si se quiere discutir un diseño con alguien más, ambos deben conocer el lenguaje de modelado y no así el proceso que se siguió para obtenerlo. **(20)**

1.3.3 Herramientas CASE. Visual Paradigm.

Las herramientas CASE facilitan los procesos de desarrollo de software y automatizan las actividades de gestión de proyectos, y colaboran en la etapa de análisis, diseño y codificación. **(21)**

Como herramienta de modelado se escogió el Visual Paradigm pues le permite al programador trabajar de forma organizada y ágil. Con la utilización de esta herramienta se logra optimizar el tiempo de trabajo, pues se genera el código y las relaciones entre las clases. Además actualmente la UCI cuenta con una licencia que permite el uso de esta herramienta.

Visual Paradigm provee características de modelación extensiva para casos de uso incluyendo funciones completas para diagramas de casos de uso, editor de eventos de flujo y generación de diagrama de actividades. Los desarrolladores pueden diseñar

Capa de acceso a datos a través del Mapeo Relacional de Objetos Difusos en la herramienta para la evaluación de competencias ANAELIX

documentación del sistema a través de plantillas. El análisis del sistema puede estimar las consecuencias a cambios con diagramas de análisis de impacto, como matrices y diagramas de análisis.

1.3.4 Lenguajes de programación. Java

Un lenguaje de programación está compuesto por un conjunto de reglas que definen el significado de cada uno de los elementos que lo componen. A continuación se explican algunas características del lenguaje escogido para el desarrollo de la capa de acceso a datos, aunque en la actualidad existen muchos:

Java es un lenguaje orientado a objeto, de una plataforma independiente. Fue desarrollado por la compañía *Sun Microsystems*, con la idea original de usarlo para la creación de páginas *web*.

Esta programación Java tiene muchas similitudes con el lenguaje C y C++, así que si se tiene conocimiento de este lenguaje, el aprendizaje de la programación Java será de fácil comprensión por un programador que haya realizado programas en estos lenguajes.

Otra de las utilidades de la programación en Java es el desarrollo de aplicaciones, que son programas que se ejecutan en forma independiente, es decir con la programación Java, se pueden realizar aplicaciones como un procesador de palabras, una hoja que sirva para cálculos, una aplicación gráfica. En resumen cualquier tipo de aplicación se puede realizar con ella.

La programación en Java, permite el desarrollo de aplicaciones bajo el esquema de cliente servidor, y de aplicaciones distribuidas, lo que lo hace capaz de conectar dos o más computadoras u ordenadores, ejecutando tareas simultáneamente, y de esta forma logra distribuir el trabajo a realizar. **(22)**

Se empleará el lenguaje Java. La principal característica que conlleva a su selección es su fácil conexión con el SGBD Oracle que es donde se encuentra la base de datos difusa que se empleará en la implementación. Además el sistema ANAELIX está implementado en Java.

1.3.5 Entorno de desarrollo integrado.

Capa de acceso a datos a través del Mapeo Relacional de Objetos Difusos en la herramienta para la evaluación de competencias ANAELIX

Un Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés), es un programa compuesto por un conjunto de herramientas útiles para un desarrollador de software. Además cuenta con en un editor de código, un compilador/intérprete y un depurador. (23)

A continuación se describen algunos IDEs para aplicaciones en Java:

Netbeans.

NetBeans es una aplicación de código abierto ("*Open Source*") diseñada para el desarrollo de aplicaciones fácilmente portables entre las distintas plataformas. Dispone de soporte para crear interfaces gráficas de forma visual, desarrollo de aplicaciones *web*, control de versiones, colaboración entre varias personas, creación de aplicaciones compatibles con teléfonos móviles y resaltado de sintaxis. (24)

Eclipse.

Es una potente y completa plataforma de programación, desarrollo y compilación de elementos tan variados como sitios *web*, programas en C++, o aplicaciones Java. Es un IDE en el que se encuentran herramientas y funciones necesarias para el desarrollo de aplicaciones, recogidas además en una atractiva interfaz que lo hace fácil y agradable de usar. En otras palabras, es un entorno de desarrollo integrado, de código abierto y multiplataforma. (25)

Selección del IDE a utilizar.

Como IDE se escogió Netbeans por las prestaciones que brinda a los programadores pues:

- Contiene gran cantidad de bibliotecas locales que facilitan el trabajo con el *framework* Hibernate, que Eclipse no cuenta con ellas, por lo que se necesita importarlas.
- Es un producto libre y gratuito sin restricciones de uso.
- Existe en la actualidad mucha documentación sobre este entorno de desarrollo, pues son muchos los usuarios que lo utilizan.
- Se tienen experiencias de su uso, pues en la universidad se trabaja con él.

1.3.6 Frameworks para el mapeo de objetos en el lenguaje Java.

El trabajar con software orientado a objetos y con una base de datos relacional puede llegar a ser engorroso y puede llegar a tomar bastante tiempo en los entornos empresariales de hoy en día, es por eso que han surgido nuevas herramientas llamadas *frameworks* que agilizan el proceso de desarrollo de la aplicación. (26)

A continuación se realiza una breve descripción de algunos de los *frameworks* más utilizados en la actualidad para el lenguaje de programación Java, aunque existen más:

iBATIS

Basa su funcionamiento en el mapeo de sentencias SQL que se incluyen en ficheros XML. Eso significa que requiere conocimiento de SQL por parte del programador. Además, permite la optimización de las consultas, ya sea con lenguaje estándar o con SQL propietario del motor de base de datos utilizado. Con iBATIS, siempre se sabe lo que se está ejecutando en la base de datos. Cuando el modelo de datos es muy cambiante o es preexistente al desarrollo de la aplicación (y compartido con otras), iBATIS es un claro caso de uso. También lo es cuando las relaciones entre las entidades del modelo son muy complicadas, porque con algo de trabajo se puede conseguir que el número de consultas que se pasan a la base de datos no sea excesivo, sobre todo en los listados descriptivos. (12)

iBATIS requiere escribir su propio código del SQL, por lo tanto, la portabilidad de una solicitud de iBATIS depende de qué SQL se ha implementado. Si las consultas se escriben utilizando SQL portátil, iBATIS también es portable a través de diferentes bases de datos relacionales. (12)

Hibernate

Hibernate es una herramienta de mapeo objeto/relacional para entornos Java. El término de mapeo objeto/relacional se refiere a la técnica de mapear una representación de datos desde un modelo de objeto a un modelo de datos relacionales con un esquema basado en SQL. (26)

Hibernate no solamente se ocupa del mapeo desde las clases Java a las tablas de las bases de datos (y desde los tipos de datos de Java a los tipos de datos de SQL), sino que *Capa de acceso a datos a través del Mapeo Relacional de Objetos Difusos en la herramienta para la evaluación de competencias ANAELIX*

también facilita la consulta y recuperación de datos. Esto puede reducir de manera importante el tiempo de desarrollo que se tomaría con el manejo de datos de forma manual en SQL y JDBC. (26)

La meta de Hibernate es aliviar el trabajo del desarrollador en 95% de la persistencia de datos comunes, relacionados con tareas de programación. (26)

Según Gavin King (26), algunas de las **características técnicas** de Hibernate son:

- Mapeado flexible gracias a las asociaciones bidireccionales, ya sea con archivos XML o anotaciones.
- Facilidades en consultas, debido en parte a que se realizan en un potente lenguaje de consultas orientado a objetos.
- Facilidades en metadatos, pues soporta el formato del mapeado XML, diseñado para ser editado a mano y el mapeado basado en anotaciones. Además de validación basada en anotaciones.

Hibernate en su funcionamiento 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 todas las bases de datos con un ligero incremento en el tiempo de ejecución. A la hora de realizar una optimización en las consultas, el hecho de generar el código en caliente hace que sea más fácil modificarlo y ajustarlo, cambiando un parámetro de *true* a *false*, pasamos de ejecutar todas las sentencia con y sin uniones de tablas, compárese esto con la otra posibilidad, que es cambiar todo a mano. (27)

La arquitectura de Hibernate.

Internamente, Hibernate utiliza JDBC, que proporciona una capa de abstracción de la base de datos. La información de conexión que Hibernate necesita para interactuar con la base de datos es proporcionada por la agrupación de conexiones JDBC, que tiene que ser configurada. (28)

La arquitectura de Hibernate se compone principalmente de dos interfaces (*Session* y *Transaction*) junto con la interfaz *Query*, que se encuentra en la capa de persistencia de la aplicación. Las clases que se definen en la capa de negocio de la solicitud, interactúan a

través de metadatos independientes de la capa de persistencia, que a su vez mantiene relación con la capa de datos. Además, Hibernate usa otras interfaces para la configuración, sobre todo la clase *Configuration*. Hibernate también hace uso de las interfaces de devolución de llamada y algunas interfaces opcionales para extender la funcionalidad de asignación. (28)

Selección del *framework* para el mapeo de objetos.

En el presente trabajo se utilizará como *framework* **Hibernate** haciendo uso de las **anotaciones**, pues es uno de los más usados en la actualidad para el lenguaje Java y además permite una solución completa como ORM, ya que iBATIS no proporciona una solución ORM completa y no proporciona ninguna asignación directa de los objetos y modelos relacionales. Hibernate posee gran portabilidad a través de diferentes bases de datos relacionales, pues una vez que se desee cambiar de SGBD solo se debe sustituir en el archivo de configuración el tipo de dialecto. También en la actualidad existe gran diversidad de documentación referente a Hibernate, y tiene una amplia comunidad activa que proporciona soporte para los nuevos usuarios.

1.3.7 Gestor de base de datos. Oracle 10g.

Se empleará como gestor de base de datos Oracle 10g porque es el que utiliza el sistema ANAELIX, pues es donde se encuentra el servidor difuso. Aunque este gestor es propietario, la universidad cuenta con una licencia para su uso. Además se logró probar la base de datos difusa usando PostgreSQL y se pudo comprobar que el servidor difuso para PostgreSQL está incompleto a la vez que presenta varios errores que imposibilitan su uso.

1.4 Consideraciones del capítulo

- Se elaboró el marco teórico de la investigación, mediante un análisis sobre el estado actual de los sistemas evaluadores de competencias y sobre cómo estos realizan el acceso a los datos. De esta manera resultó más fácil seleccionar las herramientas, metodologías, técnicas y tecnologías a utilizar.
- Se realizó un estudio sobre las principales herramientas de mapeo existentes seleccionando a Hibernate como el ORM a utilizar.

- Se seleccionó a XP como metodología de desarrollo, Netbeans como IDE de desarrollo, Java como lenguaje de programación y Oracle 10g como gestor de base de datos, en cada epígrafe se justificó debidamente el uso de cada tecnología a emplear, considerando que son las más idóneas para la implementación de la capa de acceso.

CAPÍTULO 2: SOLUCIÓN PROPUESTA.

Introducción

En el presente capítulo se describen de forma general las características principales de la capa de acceso a datos. Se especifican además los requisitos funcionales, no funcionales y se presenta una breve descripción de algunas historias de usuarios, para las cuales se toma como usuario al desarrollador.

2.1 Descripción de la solución.

La solución propuesta se enfoca en la creación de una **capa de acceso a datos**, que permite **dos tipos de entradas** desde el sistema ANAELIX:

- Entradas necesarias para realizar las operaciones de *Insert*, *Delete* y *Update* respectivamente, sobre las tablas existentes en la base de datos, referentes a los conceptos de competencias, evaluaciones, puestos de trabajos, instrumentos evaluadores, así como toda la información difusa relacionada con el sistema. De esta forma se logrará el mapeo relacional de objetos difusos, utilizando el *framework* Hibernate, con todas las funciones que brinda para la realización de consultas.
- Parámetros necesarios para realizar consultas a la BDD, esta última permite el almacenamiento y tratamiento de información imprecisa a través del uso de atributos difusos, operadores difusos, etiquetas lingüísticas, trapecios probabilísticos y valores aproximados, para la representación y el tratamiento de la información imprecisa presente en los fenómenos de la realidad. Ante la dificultad del *framework* Hibernate, de trabajar con consultas difusas, se presenta como alternativa la traducción de las sentencias difusas al lenguaje SQL, haciendo uso para ello de la función **FSQL2SQL**, perteneciente al paquete del servidor FSQL y a partir de ahí entonces trabajar naturalmente con las funciones de Hibernate para lograr la ejecución de consultas difusas en la BD, y obtener los resultados esperados.

La siguiente figura muestra la solución que se propone para lograr el acceso desde el sistema ANAELIX a la BDD que se hace referencia (9).

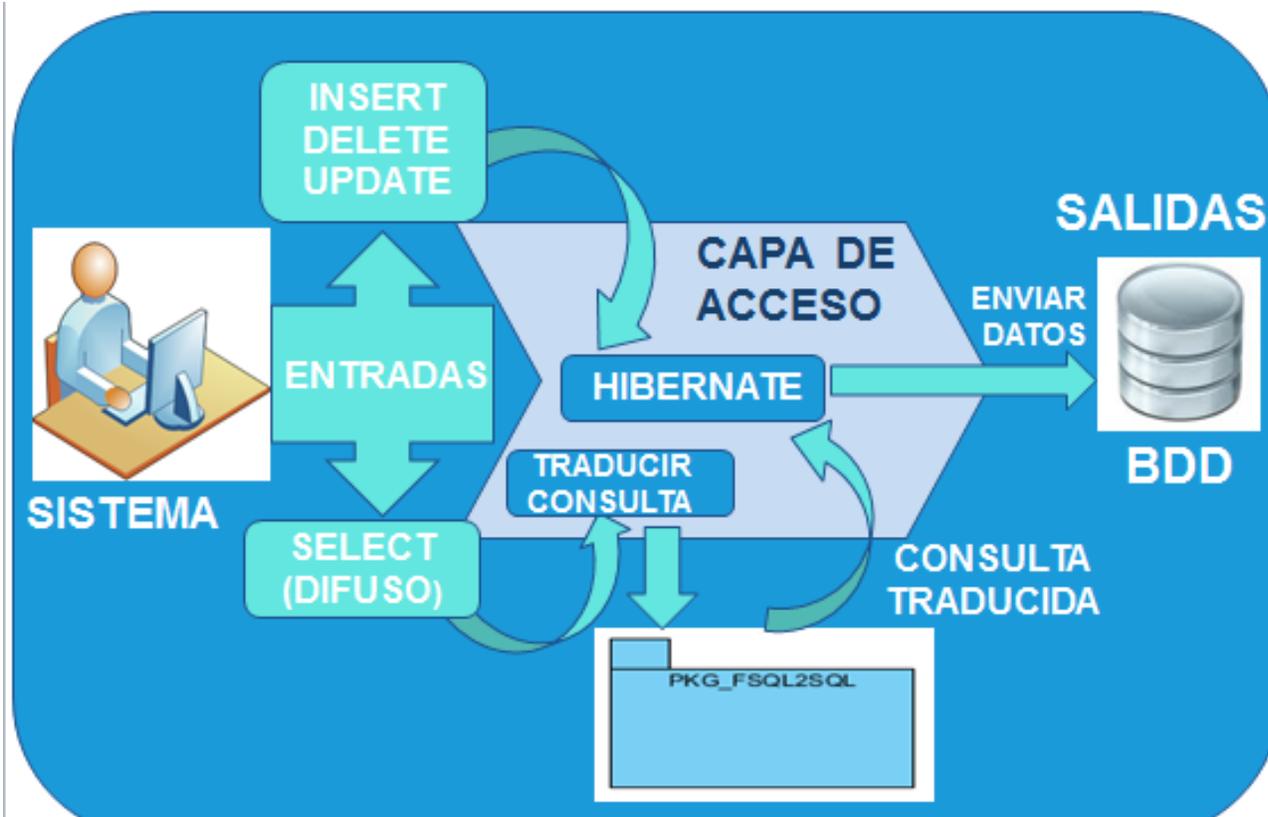


Figura 1 Solución Propuesta

Entre las ventajas de la creación de la capa de acceso a datos se encuentran:

- Se puede mapear una jerarquía de clases a una sola tabla o crear una tabla por cada clase concreta.
- Soporta el mapeo de todo tipo de relaciones que pueden existir entre los objetos de dominio, como asociaciones de uno a muchos, uno a uno y muchos a muchos.
- Se disminuye las veces que se accede a la BD, ya que se guardan en memoria los objetos que son accedidos varias veces.
- No es necesario tener conocimientos profundos de FSQ para realizar peticiones difusas a la base de datos.

2.2 Fase de Planificación.

Para el desarrollo de la capa de acceso a datos se divide el trabajo en cuatro fases como propone la metodología XP, las mismas son: Fase de Planificación, Fase de Diseño, Fase

de Implementación y Fase de Prueba según (18).

En la fase de planificación se lleva a cabo el proceso enmarcado dentro de la ingeniería de requerimientos; se realiza la lista de reserva del producto con los requisitos funcionales y no funcionales, se considera que una descripción detallada de estos podría, junto a las historias de usuario, facilitar el desarrollo del sistema. Se crean, además, las historias de usuario para obtener una descripción de las funcionalidades con que debe cumplir el sistema y finalmente se planifica el tiempo que durará el desarrollo mediante un plan de iteración.

2.2.1 Lista de reserva del producto.

A través de la lista de reserva del producto se definen y priorizan las funcionalidades que tendrá la capa de acceso a datos, además se describen los requisitos no funcionales que tendrá el software.

Requisitos Funcionales.

Los requisitos funcionales definen las funciones que la capa de acceso a datos será capaz de realizar. Describen también las transformaciones que la capa de acceso a datos realizará sobre las entradas para producir salidas. (29)

A continuación se muestra una tabla donde se reflejan los requisitos funcionales de la capa de acceso a datos.

Tabla 1 Requisitos funcionales.

Código	Nombre del Requisito	Prioridad
	Gestionar Competencia.	Media
RF#1	Insertar Competencia.	Media
RF#2	Modificar Competencia.	Media
RF#3	Eliminar Competencia.	Media
	Gestionar Dimensión.	Media
RF#4	Insertar Dimensión.	Media
RF#5	Modificar Dimensión.	Media
RF#6	Eliminar Dimensión.	Media

Capa de acceso a datos a través del Mapeo Relacional de Objetos Difusos en la herramienta para la evaluación de competencias ANAELIX

	Gestionar Ocupación.	Media
RF#7	Insertar Ocupación.	Media
RF#8	Modificar Ocupación.	Media
RF#9	Eliminar Ocupación.	Media
	Gestionar Requerimiento.	Alta
RF#10	Insertar Requerimiento.	Alta
RF#11	Modificar Requerimiento.	Alta
RF#12	Eliminar Requerimiento.	Alta
	Gestionar Pregunta.	Media
RF#13	Insertar Pregunta.	Media
RF#14	Modificar Pregunta.	Media
RF#15	Eliminar Pregunta.	Media
	Gestionar Evaluación.	Alta
RF#16	Insertar Evaluación.	Alta
RF#17	Modificar Evaluación.	Alta
RF#18	Eliminar Evaluación.	Alta
	Gestionar Evaluador.	Alta
RF#19	Insertar Evaluador.	Alta
RF#20	Modificar Evaluador.	Alta
RF#21	Eliminar Evaluador.	Alta
	Gestionar Persona.	Media
RF#22	Insertar Persona.	Media
RF#23	Modificar Persona.	Media
RF#24	Eliminar Persona.	Media
	Gestionar Grado de Similitud.	Alta
RF#25	Insertar Grado de Similitud.	Alta
RF#26	Modificar Grado de Similitud.	Alta
RF#27	Eliminar Grado de Similitud.	Alta
	Gestionar Tipo de Pregunta.	Media
RF#28	Insertar Tipo de Pregunta.	Media
RF#29	Modificar Tipo de Pregunta.	Media
RF#30	Eliminar Tipo de Pregunta.	Media

Capa de acceso a datos a través del Mapeo Relacional de Objetos Difusos en la herramienta para la evaluación de competencias ANAELIX

RF#31	Traducir consulta FSQL a SQL.	Muy Alta
-------	-------------------------------	----------

Requisitos no Funcionales.

Los requisitos no funcionales están relacionados con las características que de una u otra forma puedan limitar el funcionamiento de la capa de acceso a datos. Describen una restricción sobre el sistema que limita nuestra elección en la construcción de una solución. (29)

En la siguiente tabla se muestran los requisitos no funcionales de la capa de acceso a datos.

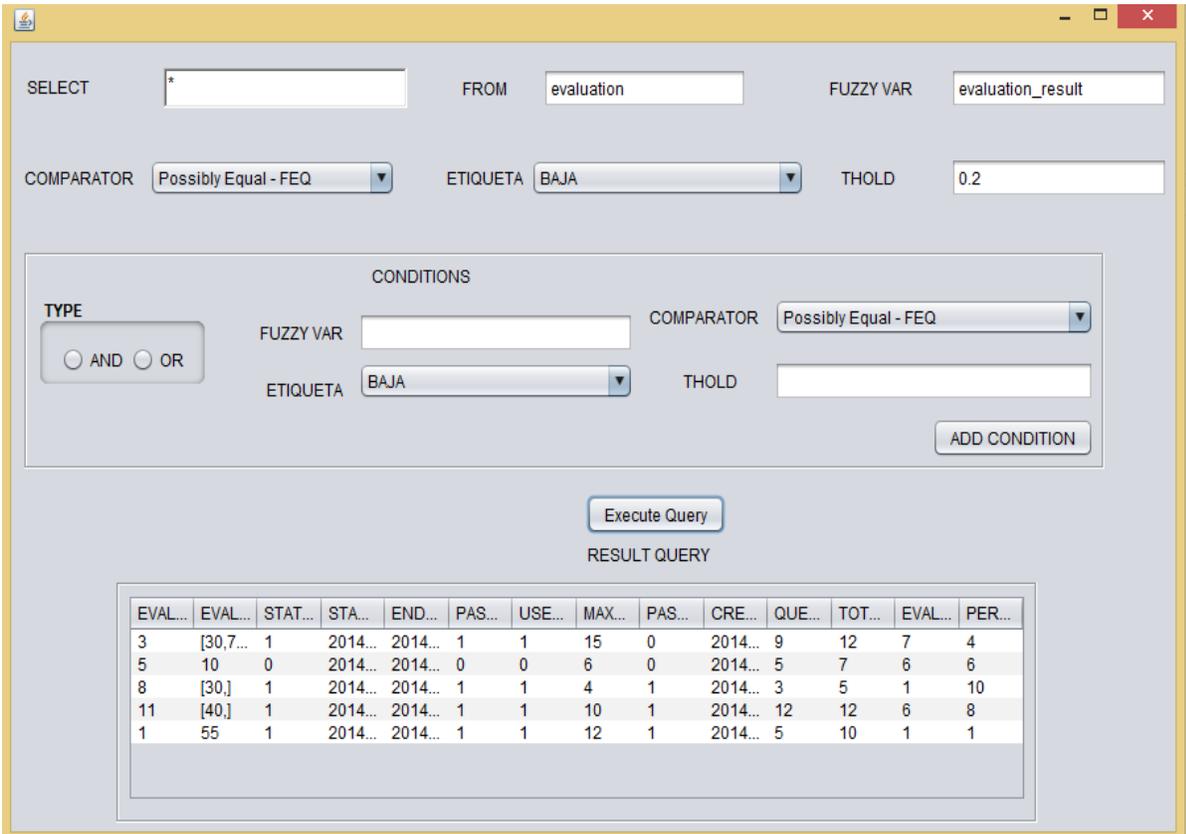
Tabla 2 Requisitos no funcionales

Código	Clasificación
	RNF de Software
RNF#1	La base de datos se implementará en el SGBD Oracle10g.
	RNF de Hardware
RNF#2	Se requiere un procesador de 500 MHz (mínimo), 512 MB mínimo de RAM y un disco duro 80 GB o superior por el volumen de datos que se genera.
	Restricciones de diseño y la implementación
RNF#3	Se debe tener instalado un servidor FSQL o de lo contrario tener acceso a alguno.
	RNF de Portabilidad
RNF#4*	La BD debe ser compatible con el sistema operativo Windows y Linux.

2.2.2 Historias de Usuario.

Durante la fase de planificación se generan las Historias de Usuario (HU), que son los documentos de especificación funcional de una aplicación, son escritas por el cliente en su propio lenguaje con descripciones cortas de lo que debe hacer la capa de acceso a datos. A continuación se describen algunos ejemplos de historias de usuario a desarrollar de un total de 31, el resto se encuentra en los artefactos anexos del trabajo:

Tabla 3 HU Traducir consulta FSQL a SQL

Historia de Usuario																																																																																					
Número: 1	Usuario: Aylín																																																																																				
Nombre historia: Traducir consulta FSQL a SQL.																																																																																					
Referencia: RF#31																																																																																					
Prioridad en el negocio: Muy Alta	Nivel de complejidad: Muy Alta																																																																																				
Tiempo de Estimación: 3	Iteración asignada: 1																																																																																				
Descripción: Esta funcionalidad en la capa de acceso a datos, permite introducir todos los campos que deben ser llenados para construir la consulta difusa y luego se traduce al SQL y se ejecuta, obteniendo los resultados de la misma.																																																																																					
Observaciones:																																																																																					
 <p style="text-align: center;">Execute Query</p> <p style="text-align: center;">RESULT QUERY</p> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th>EVAL...</th> <th>EVAL...</th> <th>STAT...</th> <th>STA...</th> <th>END...</th> <th>PAS...</th> <th>USE...</th> <th>MAX...</th> <th>PAS...</th> <th>CRE...</th> <th>QUE...</th> <th>TOT...</th> <th>EVAL...</th> <th>PER...</th> </tr> </thead> <tbody> <tr> <td>3</td> <td>[30,7...</td> <td>1</td> <td>2014...</td> <td>2014...</td> <td>1</td> <td>1</td> <td>15</td> <td>0</td> <td>2014...</td> <td>9</td> <td>12</td> <td>7</td> <td>4</td> </tr> <tr> <td>5</td> <td>10</td> <td>0</td> <td>2014...</td> <td>2014...</td> <td>0</td> <td>0</td> <td>6</td> <td>0</td> <td>2014...</td> <td>5</td> <td>7</td> <td>6</td> <td>6</td> </tr> <tr> <td>8</td> <td>[30.]</td> <td>1</td> <td>2014...</td> <td>2014...</td> <td>1</td> <td>1</td> <td>4</td> <td>1</td> <td>2014...</td> <td>3</td> <td>5</td> <td>1</td> <td>10</td> </tr> <tr> <td>11</td> <td>[40.]</td> <td>1</td> <td>2014...</td> <td>2014...</td> <td>1</td> <td>1</td> <td>10</td> <td>1</td> <td>2014...</td> <td>12</td> <td>12</td> <td>6</td> <td>8</td> </tr> <tr> <td>1</td> <td>55</td> <td>1</td> <td>2014...</td> <td>2014...</td> <td>1</td> <td>1</td> <td>12</td> <td>1</td> <td>2014...</td> <td>5</td> <td>10</td> <td>1</td> <td>1</td> </tr> </tbody> </table>		EVAL...	EVAL...	STAT...	STA...	END...	PAS...	USE...	MAX...	PAS...	CRE...	QUE...	TOT...	EVAL...	PER...	3	[30,7...	1	2014...	2014...	1	1	15	0	2014...	9	12	7	4	5	10	0	2014...	2014...	0	0	6	0	2014...	5	7	6	6	8	[30.]	1	2014...	2014...	1	1	4	1	2014...	3	5	1	10	11	[40.]	1	2014...	2014...	1	1	10	1	2014...	12	12	6	8	1	55	1	2014...	2014...	1	1	12	1	2014...	5	10	1	1
EVAL...	EVAL...	STAT...	STA...	END...	PAS...	USE...	MAX...	PAS...	CRE...	QUE...	TOT...	EVAL...	PER...																																																																								
3	[30,7...	1	2014...	2014...	1	1	15	0	2014...	9	12	7	4																																																																								
5	10	0	2014...	2014...	0	0	6	0	2014...	5	7	6	6																																																																								
8	[30.]	1	2014...	2014...	1	1	4	1	2014...	3	5	1	10																																																																								
11	[40.]	1	2014...	2014...	1	1	10	1	2014...	12	12	6	8																																																																								
1	55	1	2014...	2014...	1	1	12	1	2014...	5	10	1	1																																																																								

Capa de acceso a datos a través del Mapeo Relacional de Objetos Difusos en la herramienta para la evaluación de competencias ANAELIX

Tabla 4 HU Insertar Competencia

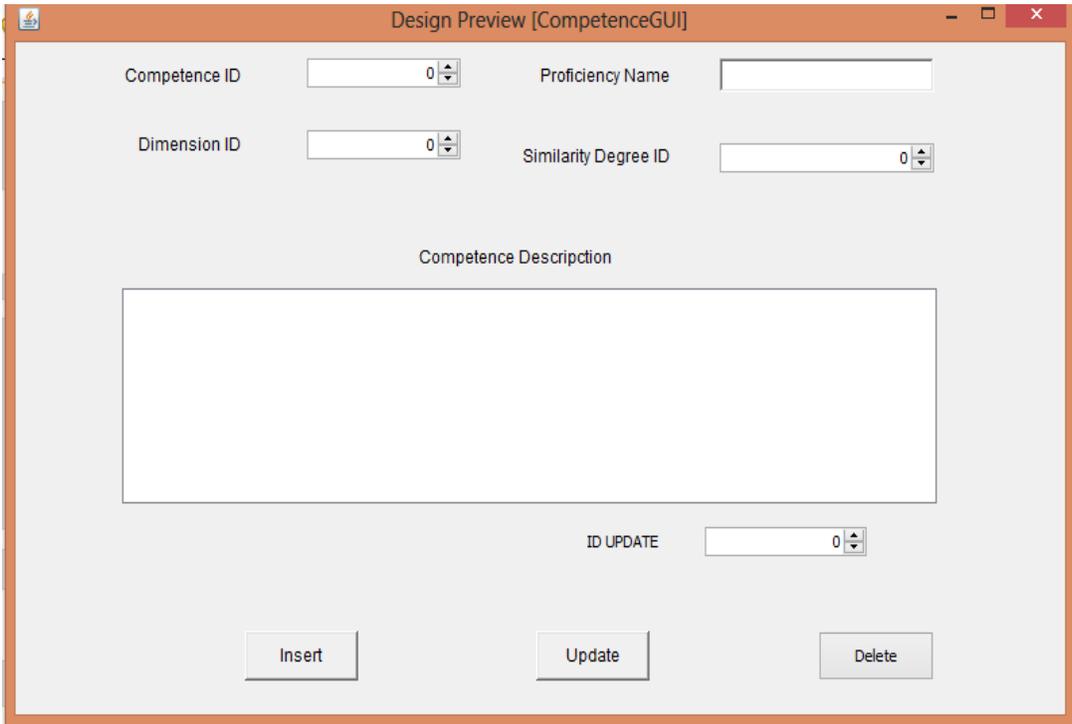
Historia de Usuario	
Número: 2	Usuario: Aylín
Nombre historia: Insertar Competencias.	
Referencia: RF#1	
Prioridad en el negocio: Media	Nivel de complejidad: Media
Tiempo de Estimación: 1/2	Iteración asignada: 3
Descripción: En la capa de acceso a datos para insertar una nueva competencia se necesitan todos los campos que deben ser llenados y luego se guarda en la BD la nueva competencia.	
Observaciones:	
	

Tabla 5 HU Modificar Competencia

Historia de Usuario	
Número: 3	Usuario: Aylín

Capa de acceso a datos a través del Mapeo Relacional de Objetos Difusos en la herramienta para la evaluación de competencias ANAELIX

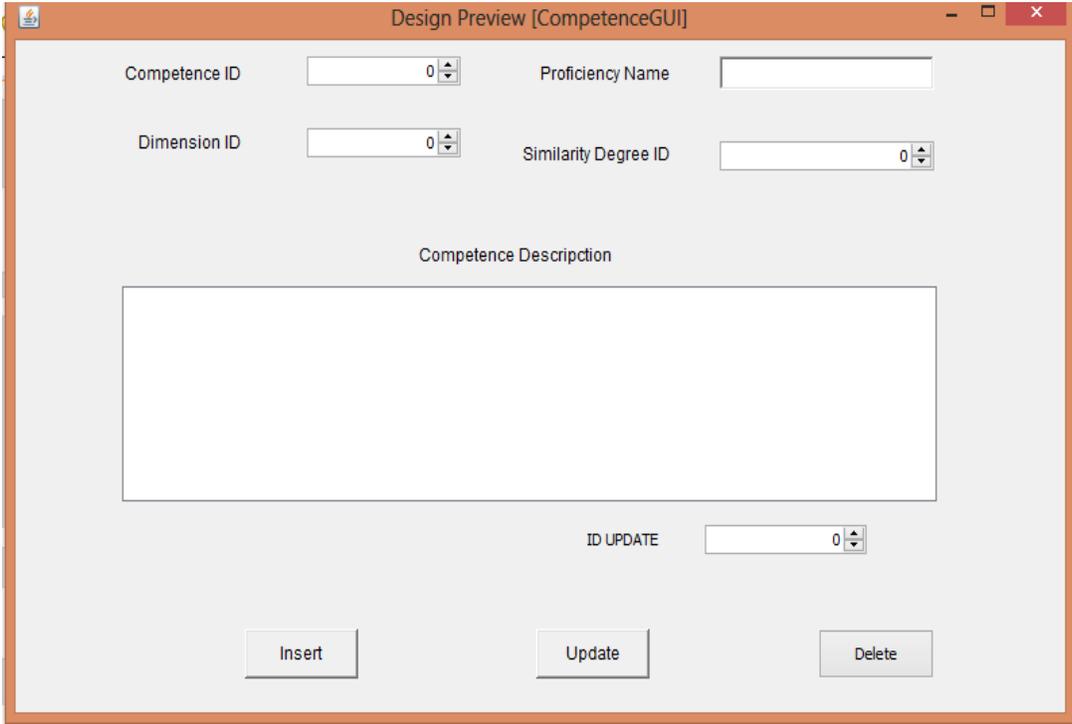
Nombre historia: Modificar Competencia.	
Referencia: RF#2	
Prioridad en el negocio: Media	Nivel de complejidad: Media
Tiempo de Estimación: 1/2	Iteración asignada: 3
Descripción: En la capa de acceso a datos para modificar una competencia se necesita introducir el identificador de la que se desea modificar y los campos a modificar, luego se guardan los cambios en la BD para la competencia especificada.	
Observaciones:	
	

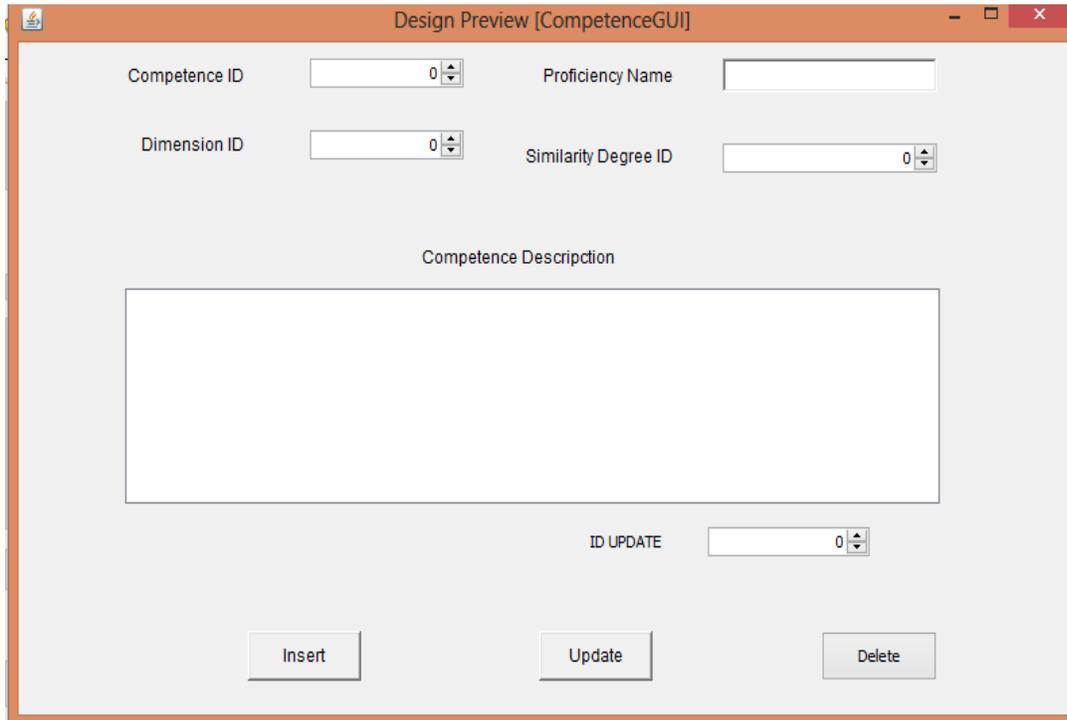
Tabla 6 HU Eliminar Competencia

Historia de Usuario	
Número: 4	Usuario: Aylín
Nombre historia: Eliminar Competencia.	
Referencia: RF#3	
Prioridad en el negocio: Media	Nivel de complejidad: Media
Tiempo de Estimación: 1/2	Iteración asignada: 3

Capa de acceso a datos a través del Mapeo Relacional de Objetos Difusos en la herramienta para la evaluación de competencias ANAELIX

Descripción: En la capa de acceso a datos para eliminar una competencia se necesita introducir el identificador de la que se desea eliminar y luego se elimina de la BD la competencia especificada.

Observaciones:



2.2.3 Estimación de esfuerzos por historias de usuario.

En esta fase se determina cuánto tiempo en semanas se necesita para realizar cada una de las historias de usuario, lo que permite establecer la velocidad y el tiempo en que se desarrolla el software. La siguiente tabla muestra la estimación del esfuerzo requerido para cada HU:

Tabla 7 Estimación de esfuerzos por historias de usuario

No. HU	Historia de Usuario	Puntos de estimación(semanas)
1	Traducir consulta FSQL a SQL.	3

Capa de acceso a datos a través del Mapeo Relacional de Objetos Difusos en la herramienta para la evaluación de competencias ANAELIX

2	Insertar Competencia.	1/2
3	Modificar Competencia.	1/2
4	Eliminar Competencia.	1/2
5	Insertar Requerimiento.	1
6	Modificar Requerimiento.	1
7	Eliminar Requerimiento.	1
8	Insertar Evaluación.	1
9	Modificar Evaluación.	1
10	Eliminar Evaluación.	1
11	Insertar Evaluador.	1
12	Modificar Evaluador.	1
13	Eliminar Evaluador.	1
14	Insertar Grado de Similitud.	1
15	Modificar Grado de Similitud.	1
16	Eliminar Grado de Similitud.	1
17	Insertar Dimensión.	1/2
18	Modificar Dimensión.	1/2
19	Eliminar Dimensión.	1/2
20	Insertar Ocupación.	1/2
21	Modificar Ocupación.	1/2
22	Eliminar Ocupación.	1/2
23	Insertar Persona.	1/2
24	Modificar Persona.	1/2
25	Eliminar Persona.	1/2
26	Insertar Pregunta.	1/2
27	Modificar Pregunta.	1/2
28	Eliminar Pregunta.	1/2
29	Insertar Tipo de Pregunta.	1/2
30	Modificar Tipo de Pregunta.	1/2
31	Eliminar Tipo de Pregunta.	1/2

2.2.4 Plan de duración de las iteraciones.

Capa de acceso a datos a través del Mapeo Relacional de Objetos Difusos en la herramienta para la evaluación de competencias ANAELIX

Luego de establecer y especificar las HU, se continúa con la elaboración del plan de iteraciones, donde se puntualizan cuáles HU serán implementadas en cada iteración. Para el desarrollo de esta capa de acceso a datos se establecieron 3 iteraciones:

Iteración 1: En esta iteración se implementan las HU que se consideraron con mayor prioridad relacionadas con la traducción de las consultas fundamentalmente. Al concluir esta iteración se habrá logrado la traducción y por lo tanto la ejecución de las consultas difusas, al tener implementada la HU 1.

Iteración 2: En esta iteración se dará cumplimiento de las HU que se consideraron con prioridad Alta, las mismas tienen relación con las operaciones de insertar, eliminar y modificar los atributos de cada una de las tablas de la BDD con información imprecisa. Al concluir la iteración se debe haber cumplido con las funcionalidades descritas en las HU 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16. Al finalizar esta iteración se habrá logrado el mapeo relacional de objetos difusos.

Iteración 3: En esta iteración se dará cumplimiento a las HU que se consideraron con una prioridad media. Al concluir esta iteración se contará con todas las funcionalidades descritas en las HU 2, 3, 4, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 y 31, en las cuales se hace referencia a las operaciones de insertar, eliminar y modificar los atributos de cada una de las tablas de la BDD con información solamente precisa.

A continuación se muestra una tabla que resume todo lo anteriormente descrito:

Tabla 8 Plan de duración de las iteraciones

Iteración	Historias de usuarios	Duración total(semanas)
1	<ul style="list-style-type: none"> • Traducir consulta FSQL a SQL. 	3
2	<ul style="list-style-type: none"> • Insertar Requerimiento. • Modificar Requerimiento. • Eliminar Requerimiento. • Insertar Evaluación. • Modificar Evaluación. 	12

	<ul style="list-style-type: none"> • Eliminar Evaluación. • Insertar Evaluador. • Modificar Evaluador. • Eliminar Evaluador. • Insertar Grado de Similitud. • Modificar Grado de Similitud. • Eliminar Grado de Similitud. 	
3	<ul style="list-style-type: none"> • Insertar Competencia. • Modificar Competencia. • Eliminar Competencia. • Insertar Dimensión. • Modificar Dimensión. • Eliminar Dimensión. • Insertar Ocupación. • Modificar Ocupación. • Eliminar Ocupación. • Insertar Pregunta. • Modificar Pregunta. • Eliminar Pregunta. • Insertar Persona. • Modificar Persona. • Eliminar Persona. • Insertar Tipo de Pregunta. • Modificar Tipo de Pregunta. • Eliminar Tipo de Pregunta. 	9
TOTAL		24

2.3 Atributos Difusos existentes en la base de datos.

En la sección 1.1.1, se mencionaron los tipos de datos que pueden ser almacenados en una base de datos difusa. Esta sección describe con más detalles cada uno de estos atributos

difusos, a la vez que muestra cómo representarlos en una tabla de la propia base de datos difusa.

Cada uno de estos atributos difusos utilizados en la capa de acceso a datos es de Tipo 2 (8). Los mismos permiten almacenar y representar valores difusos o distribuciones de posibilidad.

La representación en el Modelo Relacional de este tipo de atributo difuso se realiza mediante cinco atributos (T, 1, 2, 3, 4) (8), donde cada uno de ellos pueden tomar distintos valores en dependencia de su significado como se muestra en la siguiente tabla:

Tabla 9 Representación interna de atributos difusos Tipo 2.

Tipos de Valores	Atributos de la BD para cada Tipo 2				
	FT	F1	F2	F3	F4
UNKNOWN	0	NULL	NULL	NULL	NULL
UNDEFINED	1	NULL	NULL	NULL	NULL
NULL	2	NULL	NULL	NULL	NULL
CRISP	3	D	NULL	NULL	NULL
LABEL	4	FUZZY_ID	NULL	NULL	NULL
INTERVALO [n,m]	5	N	NULL	NULL	m
APROXIMADAMENTE(d)	6	D	d - margen	d + margen	margen
TRAPECIO[$\alpha, \beta, \gamma, \delta$]	7	A	$\beta - \alpha$	$\gamma - \delta$	δ

Como se puede apreciar en la tabla anterior, mediante esos cinco atributos se pueden representar todos los valores o funciones permitidas en una BDD. Este trabajo utiliza principalmente los atributos de tipo (FT) 3, 4, 5, 6 y 7. A continuación se muestran algunas tablas con sus atributos difusos:

Tabla 10 Atributos difusos de la tabla “evaluator”.

Atributos difusos	Descripción
-------------------	-------------

<i>certaintyT</i> <i>certainty1</i> <i>certainty2</i> <i>certainty3</i> <i>certainty4</i>	<u>Grado de confiabilidad:</u> En este atributo se almacena el grado de confiabilidad que posee un instrumento evaluador determinado (dígase <i>test</i> , encuestas, entre otros), para de esta forma tener una idea de hasta qué punto podría ser confiable dicho evaluador a la hora de realizar las evaluaciones.
---	--

Tabla 11 Atributos difusos de la tabla “*requirement*”.

Atributos difusos	Descripción
<i>fulfillment_degreeT</i> <i>fulfillment_degree1</i> <i>fulfillment_degree2</i> <i>fulfillment_degree3</i> <i>fulfillment_degree4</i>	<u>Grado de cumplimiento:</u> El grado de cumplimiento se puede definir como el nivel mínimo requerido en una competencia para poder desempeñar un rol determinado, en una ocupación, de manera exitosa.
<i>importance_degreeT</i> <i>importance_degree1</i> <i>importance_degree2</i> <i>importance_degree3</i> <i>importance_degree4</i>	<u>Grado de importancia:</u> En este atributo se almacena el grado de importancia que posee una competencia determinada para un rol específico en una ocupación.

Tabla 12 Atributos difusos de la tabla “*similarity_degree*”.

Atributos difusos	Descripción
<i>similarity_degreeT</i> <i>similarity_degree1</i> <i>similarity_degree2</i> <i>similarity_degree3</i> <i>similarity_degree4</i>	<u>Grado de similitud:</u> El grado de similitud se puede definir como la similitud que tiene una competencia determinada con todas las demás.

Tabla 13 Atributos difusos de la tabla “*evaluation*”.

Atributos difusos	Descripción
-------------------	-------------

<p><i>evaluation_resultT</i></p> <p><i>evaluation_result1</i></p> <p><i>evaluation_result2</i></p> <p><i>evaluation_result3</i></p> <p><i>evaluation_result4</i></p>	<p><u>Resultado de la evaluación:</u></p> <p>En este atributo se almacena el resultado que obtuvo una persona en su evaluación.</p>
--	---

2.4 Fase de Diseño.

La representación del diseño de la capa de acceso a datos se realiza mediante tarjetas CRC (Contenido, Responsabilidad, Colaboración) para facilitar la comprensión del mismo.

2.4.1 Tarjetas Contenido, Responsabilidad, Colaboración.

El uso de este tipo de tarjetas es una técnica de modelado que permite identificar las clases, sus atributos y responsabilidades. El objetivo es obtener un diseño simple, elegante y fácil de comprender por parte de los programadores. A continuación se muestran algunas de las tarjetas CRC de la capa de acceso a datos a desarrollar, el resto se encuentran en los artefactos anexos al trabajo.

Tabla 14 Tarjeta CRC GenericDAO

Tarjeta CRC	
Clase: GenericDAO.	
Responsabilidad	Colaboración
<p>Funciones que realiza son:</p> <ul style="list-style-type: none"> • save (Permite insertar en la BD una nueva fila en la entidad que se desee, con los parámetros introducidos por el usuario.) • delete (Permite eliminar de la BD la fila que haga referencia al identificador introducido por el usuario de la entidad que se desee.) • update (Permite actualizar en la BD la fila que haga referencia al identificador introducido por el usuario de la entidad que se desee.) • list (Permite mostrar una lista con todas las filas que 	<ul style="list-style-type: none"> • JdbcQueryManager

<p>existen en la BD de la entidad que se desee).</p> <ul style="list-style-type: none"> • selectFSQL (Permite construir la consulta difusa a partir de los parámetros introducidos por el usuario). 	
--	--

Tabla 15 Tarjeta CRC JdbcQueryManager

Tarjeta CRC	
Clase: JdbcQueryManager	
Responsabilidad	Colaboración
<p>Funciones que realiza son:</p> <ul style="list-style-type: none"> • translate2SQL (Permite traducir la consulta FSQL entrada por parámetros, al lenguaje SQL, haciendo uso del paquete FSQL2SQL del Servidor FSQL). • displayFSQLErrors (Devuelve la cantidad de errores encontrados en la consulta FSQL). 	<ul style="list-style-type: none"> • AbstractQueryManager

2.4.2 Modelo de Datos.

En la siguiente figura se representa el modelo de datos, donde se refleja la relación entre las tablas de la BD y los atributos difusos que se describieron en la sección anterior:

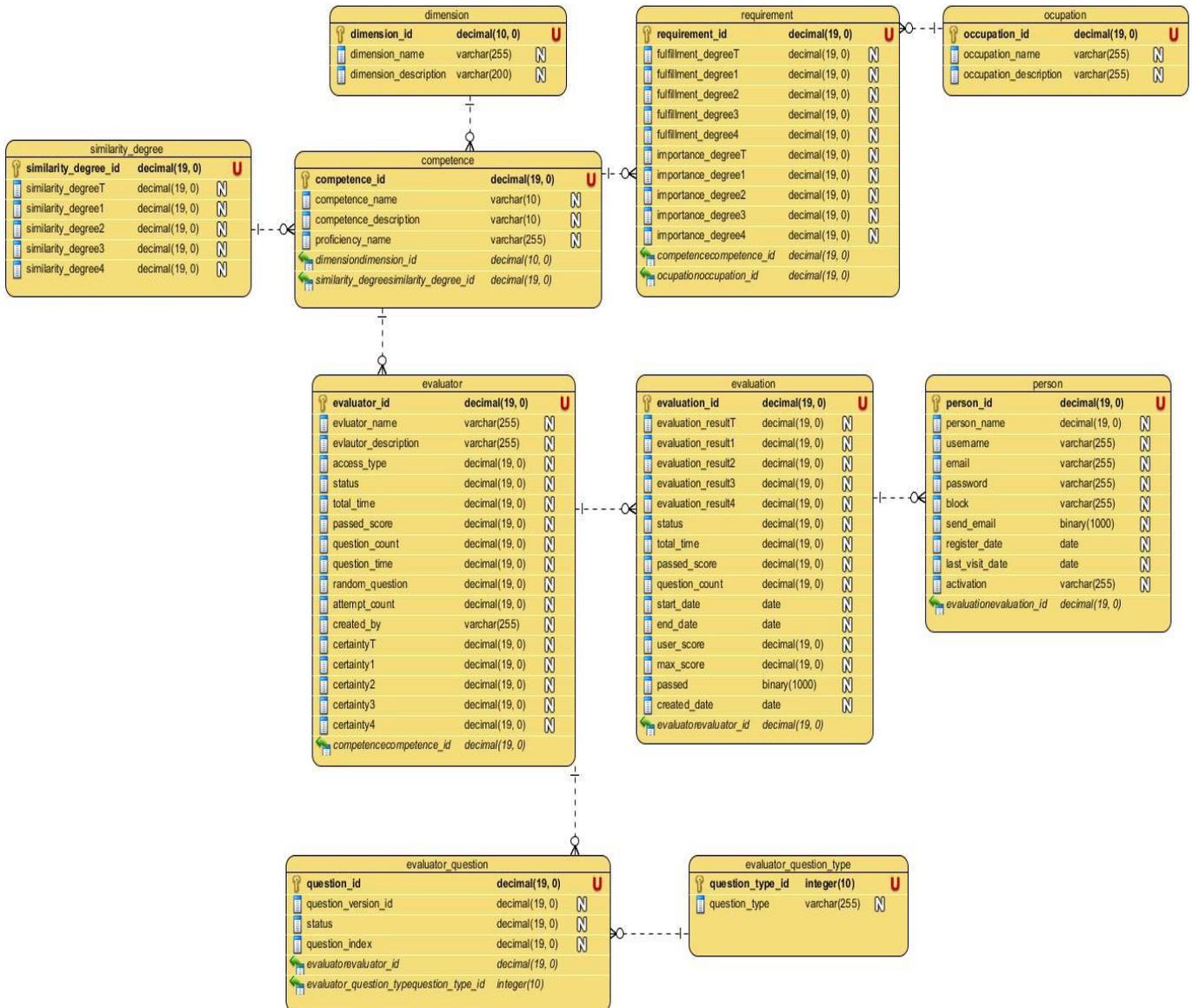


Figura 3 Modelo de Datos.

2.4.3 Patrón de Diseño.

Un patrón de diseño es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores, debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias. Los patrones de diseño pretenden evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente, proporcionando un catálogo de elementos reusables en el *Capa de acceso a datos a través del Mapeo Relacional de Objetos Difusos en la herramienta para la evaluación de competencias ANAELIX*

diseño de sistemas de software. Formaliza un lenguaje común entre diseñadores, estandarizando también el modo en que se realiza el diseño. **(30)**

En la tarea de desarrollar el acceso a datos de un sistema informático, existe un patrón de diseño que su utilización se ha hecho indispensable, este es el patrón de diseño Objeto para el Acceso a Datos (DAO, por sus siglas en inglés) es uno de los patrones de diseño de mayor uso en el desarrollo del acceso a datos de un sistema informático. Se utiliza para separar las operaciones de bajo nivel de acceso a datos, de las operaciones de alto nivel de la lógica del negocio. **(30)**

Este patrón permite implementar las operaciones definidas por una interfaz DAO de diversas maneras utilizando diferentes tecnologías, sin producir cambios en el código que utiliza las operaciones expuestas por esta interfaz, ya que cada implementación concreta de una interfaz DAO es transparente a quien utilice dicha interfaz. **(30)**

Este patrón se utiliza en el presente trabajo con la finalidad de implementar para cada una de las entidades de la BD una clase, con las operaciones de insertar, eliminar, actualizar y obtener los datos de la BD, logrando de esta forma encapsular la fuente de datos, y de ocultar la forma de acceder a los datos, por lo tanto contribuye a mantener la seguridad e integridad del sistema. Por lo que resulta muy útil como base para la capa de acceso a datos que se desea crear.

2.4.4 Diagrama de los paquetes de la capa de acceso a datos.

A continuación se representa una figura, en la cual se muestran los paquetes que conforman la capa de acceso a datos, y se realiza una breve descripción de cada uno de ellos.

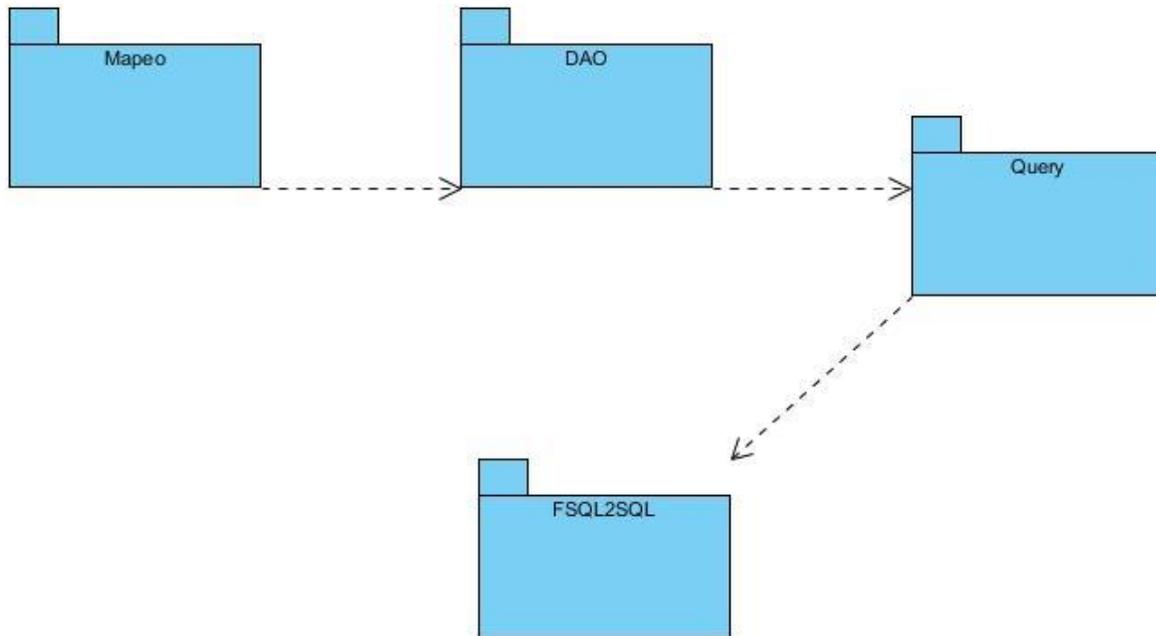


Figura 2 Diagrama de paquetes de la capa de acceso a datos.

Mapeo: En este paquete se encuentra el mapeo relacional, realizado mediante el *framework* Hibernate, utilizando anotaciones. Cuando se accede a las clases de este paquete la información obtenida es la que se encuentra almacenada en la base de datos. La información la solicita a través de las entidades DAO.

DAO: Representan las clases que se implementaron utilizando el patrón de acceso a datos DAO y encapsulan el acceso a la base de datos. En estas clases es donde se encuentran las consultas que se le realizan a la base de datos para obtener la información. En este paquete además se implementó para cada una de las entidades de la BD una clase, con las operaciones de insertar, eliminar, actualizar y obtener los datos de la BD.

Query: Este paquete contiene las clases necesarias para realizar la traducción de las consultas difusas a SQL, haciendo uso para ello del paquete FSQL2SQL del Servidor FSQL.

FSQL2SQL: Este paquete se encuentra en el Servidor FSQL y garantiza la traducción de las consultas difusas, pues realiza un análisis léxico, sintáctico y semántico de la sentencia y si no tiene errores es traducida al SQL, de lo contrario se muestran los errores encontrados. La explicación del algoritmo utilizado para la traducción va más allá del alcance de este trabajo.

2.5 Fase de Implementación.

Capa de acceso a datos a través del Mapeo Relacional de Objetos Difusos en la herramienta para la evaluación de competencias ANAELIX

En esta fase se da comienzo a la implementación de la solución a partir de las tareas de ingeniería necesarias para complementar cada HU y verificar el plan de iteraciones para comprobar si ha sido afectado. La fase de implementación se muestra en tres iteraciones, como se propuso en el capítulo anterior.

2.5.1 Tareas de Programación.

Las tareas de programación son actividades sencillas que se derivan de las historias de usuario para simplificar la implementación de las mismas, se plasman en tarjetas de papel donde se describe qué se debe realizar y son muy dinámicas y flexibles, pueden ser cambiadas por otras más generales o más específicas, agregarse nuevas o modificarse según las necesidades existentes. Cada una de estas tareas podrá ser comprobada a través de los casos de prueba.

En la siguiente tabla se relacionan las tareas de programación para la implementación de la capa de acceso a datos.

Tabla 16 Tareas de Programación

Historia de Usuario	Tareas de programación
<ul style="list-style-type: none"> • Traducir consulta FSQL a SQL 	<ul style="list-style-type: none"> • Realizar la traducción de la consulta difusa haciendo uso del paquete FSQL2SQL del servidor difuso.
<ul style="list-style-type: none"> • Insertar Competencia. 	<ul style="list-style-type: none"> • Permitir la entrada de los parámetros necesarios para insertar una nueva competencia. • Guardar los cambios en la BD.
<ul style="list-style-type: none"> • Modificar Competencia. 	<ul style="list-style-type: none"> • Permitir la entrada del identificador de la competencia que se desea modificar y los campos a modificar. • Guardar los cambios en la BD.
<ul style="list-style-type: none"> • Eliminar Competencia. 	<ul style="list-style-type: none"> • Permitir la entrada del identificador de la competencia que se desea eliminar. • Guardar los cambios en la BD.

<ul style="list-style-type: none"> • Insertar Requerimiento. 	<ul style="list-style-type: none"> • Permitir la entrada de los parámetros necesarios para insertar un nuevo requerimiento. • Guardar los cambios en la BD.
<ul style="list-style-type: none"> • Modificar Requerimiento. 	<ul style="list-style-type: none"> • Permitir la entrada del identificador del requerimiento que se desea modificar y los campos a modificar. • Guardar los cambios en la BD.
<ul style="list-style-type: none"> • Eliminar Requerimiento. 	<ul style="list-style-type: none"> • Permitir la entrada del identificador del requerimiento que se desea eliminar. • Guardar los cambios en la BD.
<ul style="list-style-type: none"> • Insertar Evaluación. 	<ul style="list-style-type: none"> • Permitir la entrada de los parámetros necesarios para insertar una nueva evaluación. • Guardar los cambios en la BD.
<ul style="list-style-type: none"> • Modificar Evaluación. 	<ul style="list-style-type: none"> • Permitir la entrada del identificador de la evaluación que se desea modificar y los campos a modificar. • Guardar los cambios en la BD.
<ul style="list-style-type: none"> • Eliminar Evaluación. 	<ul style="list-style-type: none"> • Permitir la entrada del identificador de la evaluación que se desea eliminar. • Guardar los cambios en la BD.
<ul style="list-style-type: none"> • Insertar Evaluador. 	<ul style="list-style-type: none"> • Permitir la entrada de los parámetros necesarios para insertar un nuevo evaluador. • Guardar los cambios en la BD.
<ul style="list-style-type: none"> • Modificar Evaluador. 	<ul style="list-style-type: none"> • Permitir la entrada del identificador del evaluador que se desea modificar y los campos a modificar. • Guardar los cambios en la BD.
<ul style="list-style-type: none"> • Eliminar Evaluador. 	<ul style="list-style-type: none"> • Permitir la entrada del identificador del evaluador que se desea eliminar. • Guardar los cambios en la BD.

<ul style="list-style-type: none"> • Insertar Grado de Similitud. 	<ul style="list-style-type: none"> • Permitir la entrada de los parámetros necesarios para insertar un nuevo grado de similitud. • Guardar los cambios en la BD.
<ul style="list-style-type: none"> • Modificar Grado de Similitud. 	<ul style="list-style-type: none"> • Permitir la entrada del identificador del grado de similitud que se desea modificar y los campos a modificar. • Guardar los cambios en la BD.
<ul style="list-style-type: none"> • Eliminar Grado de Similitud. 	<ul style="list-style-type: none"> • Permitir la entrada del identificador del grado de similitud que se desea eliminar. • Guardar los cambios en la BD.
<ul style="list-style-type: none"> • Insertar Dimensión. 	<ul style="list-style-type: none"> • Permitir la entrada de los parámetros necesarios para insertar una nueva dimensión. • Guardar los cambios en la BD.
<ul style="list-style-type: none"> • Modificar Dimensión. 	<ul style="list-style-type: none"> • Permitir la entrada del identificador de la dimensión que se desea modificar y los campos a modificar. • Guardar los cambios en la BD.
<ul style="list-style-type: none"> • Eliminar Dimensión. 	<ul style="list-style-type: none"> • Permitir la entrada del identificador de la dimensión que se desea eliminar. • Guardar los cambios en la BD.
<ul style="list-style-type: none"> • Insertar Ocupación. 	<ul style="list-style-type: none"> • Permitir la entrada de los parámetros necesarios para insertar una nueva ocupación. • Guardar los cambios en la BD.
<ul style="list-style-type: none"> • Modificar Ocupación. 	<ul style="list-style-type: none"> • Permitir la entrada del identificador de la ocupación que se desea modificar y los campos a modificar. • Guardar los cambios en la BD.
<ul style="list-style-type: none"> • Eliminar Ocupación. 	<ul style="list-style-type: none"> • Permitir la entrada del identificador de la ocupación que se desea eliminar. • Guardar los cambios en la BD.

<ul style="list-style-type: none"> • Insertar Persona. 	<ul style="list-style-type: none"> • Permitir la entrada de los parámetros necesarios para insertar una nueva persona. • Guardar los cambios en la BD.
<ul style="list-style-type: none"> • Modificar Persona. 	<ul style="list-style-type: none"> • Permitir la entrada del identificador de la persona que se desea modificar y los campos a modificar. • Guardar los cambios en la BD.
<ul style="list-style-type: none"> • Eliminar Persona. 	<ul style="list-style-type: none"> • Permitir la entrada del identificador de la persona que se desea eliminar. • Guardar los cambios en la BD.
<ul style="list-style-type: none"> • Insertar Pregunta. 	<ul style="list-style-type: none"> • Permitir la entrada de los parámetros necesarios para insertar una nueva pregunta. • Guardar los cambios en la BD.
<ul style="list-style-type: none"> • Modificar Pregunta. 	<ul style="list-style-type: none"> • Permitir la entrada del identificador de la pregunta que se desea modificar y los campos a modificar. • Guardar los cambios en la BD.
<ul style="list-style-type: none"> • Eliminar Pregunta. 	<ul style="list-style-type: none"> • Permitir la entrada del identificador del tipo de pregunta que se desea eliminar. • Guardar los cambios en la BD.
<ul style="list-style-type: none"> • Insertar Tipo de Pregunta. 	<ul style="list-style-type: none"> • Permitir la entrada de los parámetros necesarios para insertar un nuevo tipo de pregunta. • Guardar los cambios en la BD.
<ul style="list-style-type: none"> • Modificar Tipo de Pregunta. 	<ul style="list-style-type: none"> • Permitir la entrada del identificador del tipo de pregunta que se desea modificar y los campos a modificar.
<ul style="list-style-type: none"> • Eliminar Tipo de Pregunta. 	<ul style="list-style-type: none"> • Permitir la entrada del identificador del tipo de pregunta que se desea eliminar. • Guardar los cambios en la BD.

2.6 Consulta difusa.

Cada sentencia difusa que se desee ejecutar debe ser previamente analizada por el sistema, para asegurar que la sentencia esté correctamente escrita, que pertenece a la gramática y

Capa de acceso a datos a través del Mapeo Relacional de Objetos Difusos en la herramienta para la evaluación de competencias ANAELIX

que tiene sentido efectuarla. Para ello se utilizan tres tipos de analizadores: léxico, sintáctico y semántico (7). A continuación se muestran los resultados obtenidos al ejecutarse la siguiente consulta difusa:

Devolver todos los datos de la evaluación cuyo resultado es Baja con un grado de pertenencia de 0.2.

Consulta en FSQL:

SELECT evaluation %,cdeg(*) from evaluation where evaluation_result FEQ \$BAJA thold 0.2

Resultado de la ejecución de la consulta mediante la capa de acceso a datos:

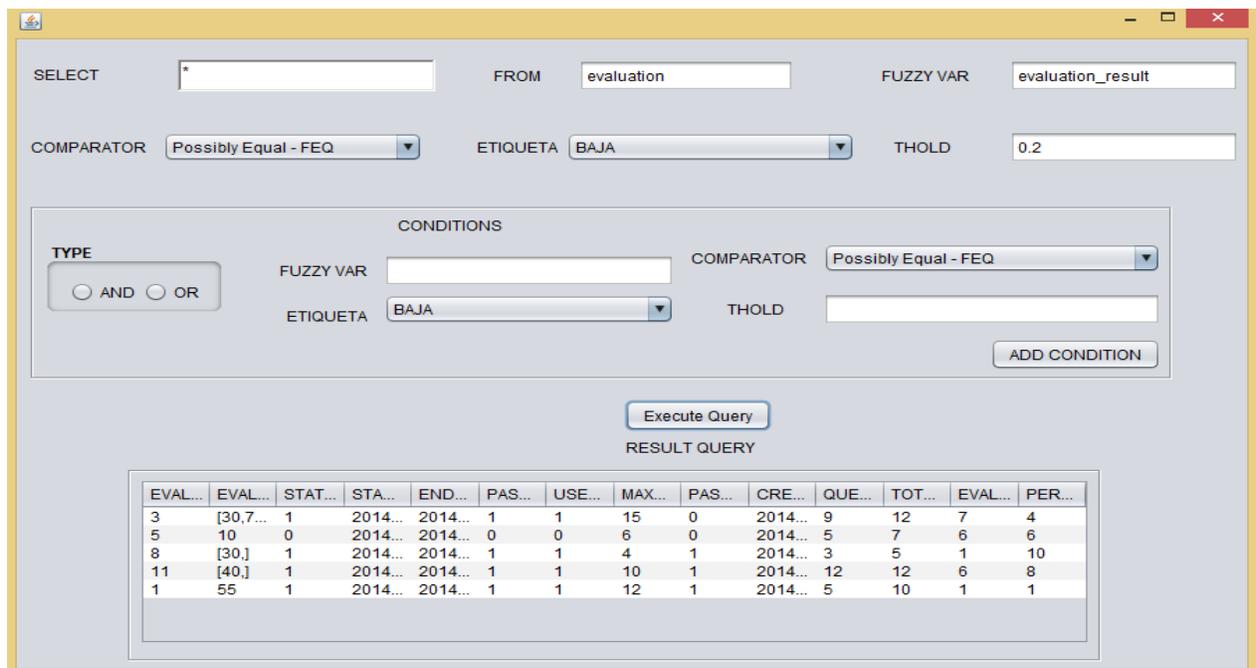


Figura 3 Resultado de la ejecución de una consulta difusa

2.7 Consideraciones del capítulo

- Se realizó el levantamiento de requisitos de la capa de acceso a datos, los cuales permitieron establecer las características con las que debe contar la solución.
- Se diseñó la capa de acceso a datos con la herramienta de mapeo relacional Hibernate, aprovechando así las ventajas que brinda el ORM.

- Se definió para cada una de las fases algunos de los artefactos que propone la metodología XP, lo cual permitió organizar todo el proceso de desarrollo, realizándose en tres iteraciones con una duración de 24 semanas.
- Se logró con éxito el tratamiento de la incertidumbre mediante el uso de una base de datos real, permitiendo tratar a una evaluación como un valor exacto o aproximado, como un rango de valores, como un trapecio probabilístico e incluso como un valor desconocido o indefinido para el dominio en cuestión.
- Se logró la ejecución de cualquier consulta difusa a través de la capa de acceso a datos utilizando el *framework* Hibernate.

CAPÍTULO 3: VALIDACIÓN Y PRUEBAS DEL DISEÑO.

Introducción

En el presente capítulo se desarrollan las pruebas, realizando para ello una validación de la solución propuesta, a través de validaciones teóricas y funcionales. Se confeccionó una sencilla aplicación de escritorio, con las interfaces que se consideraron necesarias mostrar, a las cuales se le realizaron las pruebas de aceptación para mostrar el correcto funcionamiento de la capa de acceso a datos.

3. Fase de Prueba.

El objetivo fundamental que tienen las pruebas de software, es verificar los requisitos del sistema, por lo que son los propios requisitos la principal fuente de información a la hora de construir las pruebas del sistema. (18)

3.1 Validación teórica del diseño.

La integridad en una base de datos es la corrección y exactitud de la información contenida. Cuando esta se modifica con sentencias *INSERT*, *DELETE* o *UPDATE*, la integridad de los datos almacenados puede perderse de muchas maneras diferentes. Pueden añadirse datos no válidos a la base de datos. Pueden modificarse datos existentes tomando un valor incorrecto. Los cambios en la base de datos pueden perderse debido a un error del sistema o a un fallo en el suministro de energía. (31)

3.1.1 Integridad de los datos.

El término **integridad de datos** se refiere a la corrección y complementación de los datos en una base de datos. Una de las funciones importantes de un SGBD es preservar la integridad de sus datos almacenados en la mayor medida posible. (32)

Integridad de dominio

La integridad de dominio (o columna) especifica un conjunto de valores de datos que son válidos para una columna y determina si se permiten valores nulos. La integridad de dominio se suele implementar

mediante el uso de comprobaciones de validez y, también, mediante la restricción del tipo de datos, el formato o el intervalo de los valores posibles permitidos en una columna. (32)

Para garantizar la integridad de dominio en la base de datos difusa se verificó que cada uno de los tipos de datos para cada atributo fuera el correcto, o sea que este tipo de dato se encuentre predefinido en el SGBD Oracle. Además se revisaron los valores que no pueden clasificarse de nulos haciendo uso de la restricción NOT NULL.

Integridad de Entidades

La integridad de entidad define una fila como entidad única para una tabla determinada. Además exige la integridad de las columnas de los identificadores o la clave principal de una tabla, mediante índices y restricciones UNIQUE, o restricciones PRIMARY KEY.

- UNIQUE: La regla de integridad de unicidad está relacionada con la definición de clave primaria que establece que toda clave primaria que se elija para una relación no debe tener valores repetidos por lo que el conjunto de atributos CP es la clave primaria de una relación R, entonces la extensión de R no puede tener en ningún momento dos tuplas con la misma combinación de valores para los atributos de CP.
- PRIMARY KEY: La regla de integridad de entidad de la clave primaria dispone que los atributos de la clave primaria de una relación no puedan tener valores nulos. Esta regla es necesaria para que los valores de las claves primarias puedan identificar las tuplas individuales de las relaciones. Si las claves primarias tuviesen valores nulos, es posible que algunas tuplas no se pudieran distinguir. Un SGBD relacional tendrá que garantizar el cumplimiento de esta regla de integridad en todas las inserciones y en todas las modificaciones que afecten a atributos que pertenecen a la clave primaria de la relación. (32)

Tabla 17 Análisis de integridad de entidad.

Nombre de la Tabla	Llave Primaria
<i>Similarity_degree</i>	<i>similarity_degree_id</i>
<i>Evaluator</i>	<i>evaluator_id</i>
<i>Competence</i>	<i>competence_id</i>
<i>Dimension</i>	<i>dimensión_id</i>

<i>Requirement</i>	<i>requirement_id</i>
<i>Evaluation</i>	<i>evaluation_id</i>
<i>Evaluation_question</i>	<i>question_id</i>
<i>Occupation</i>	<i>occupation_id</i>
<i>Person</i>	<i>person_id</i>
<i>Evaluation_question_type</i>	<i>question_type_id</i>

Como se puede apreciar en la tabla anterior, la integridad de entidades se cumple en la BD, debido a que cada tabla tiene definida su clave primaria y esta no es nula ni está repetida para dos filas en una misma tabla, permitiendo que cada tupla se pueda distinguir y utilizar, pues esta es la principal vía que se utiliza para seleccionar la información necesaria.

Integridad Referencial

La integridad referencial protege las relaciones definidas entre las tablas cuando se crean o se eliminan filas. Esta regla se relaciona con el concepto de clave foránea, lo que determina que todos los valores que toma una clave foránea deben ser valores nulos o valores que existen en la clave primaria a la que hace referencia. La necesidad de esta regla es debido a que las claves foráneas tienen por objetivo establecer una conexión con la clave primaria que referencian. (32)

Los conceptos de actualizar registros en cascada y eliminar registros en cascada están asociados a la integridad referencial. El primero hace referencia a que cuando se cambie un valor del campo clave de la tabla principal, automáticamente cambia el valor de la clave foránea de los registros relacionados en la tabla secundaria. El segundo indica al SGBD que cuando se elimina un registro de la tabla principal automáticamente se borran también los registros relacionados en la tabla secundaria. (32)

Con el objetivo de garantizar la integridad referencial en la base de datos difusa, se utilizaron llaves foráneas, que obligan a los valores introducidos en las columnas marcadas por esta restricción a que correspondan con los valores en las tablas referenciadas. Además se definió que la realización de la actualización y eliminación de datos sería en cascada.

3.2 Pruebas de aceptación.

Estas pruebas se llevan a cabo sin tener conocimiento de la estructura/funcionamiento interno del sistema, de ahí su nombre. Quien realiza la prueba sólo conoce las entradas apropiadas que deberá recibir la aplicación, así como las correspondientes salidas, sin llegar a saber cómo es que se realiza este proceso. (33)

Por tanto, las pruebas de aceptación se centran en lo que se espera de un módulo, es decir, intentan encontrar casos en que el módulo no se atiene a su especificación. Por ello se denominan pruebas funcionales, y el probador, en este caso el cliente, se limita a suministrarle datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el módulo por dentro. (33)

En el presente trabajo se desarrolló una sencilla aplicación de escritorio, con las interfaces que se consideraron necesarias mostrar, a las cuales se le realizaron las pruebas de aceptación o pruebas de caja negra, utilizando la técnica partición de equivalencia, para mostrar el correcto funcionamiento de la capa de acceso a datos.

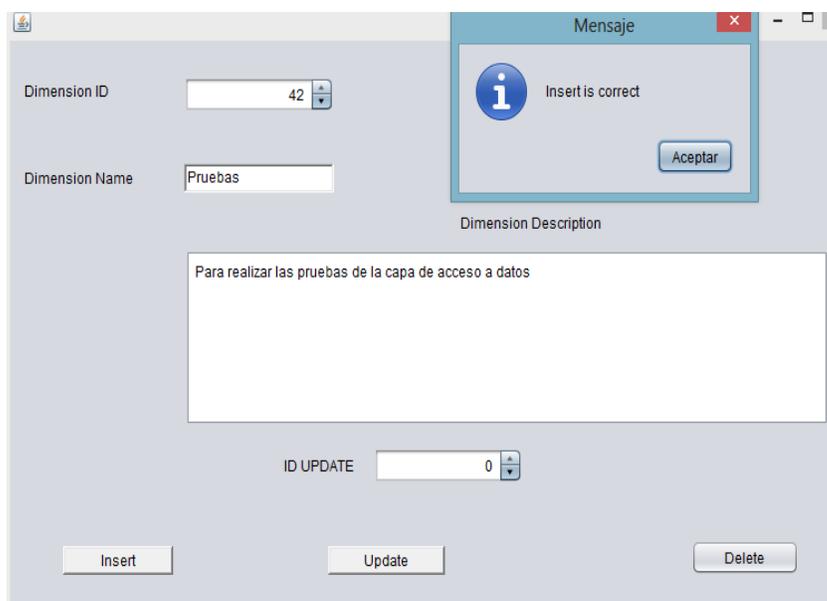
Caso de Prueba para la tabla *Dimension*.

En la presente prueba se verifica el proceso de inserción de una dimensión al insertar todos los datos correctamente en los campos requeridos.

Tabla 18 Caso de Prueba para la tabla *Dimension*

Casos de Prueba
Número de caso de prueba : 01
Nombre del Caso de prueba: Insertar dimensión con campos correctos.
Descripción de la prueba: A través de la capa de acceso a datos se accede a la interfaz de inserción de una dimensión y se llenan todos los campos necesarios para guardar la nueva dimensión en la BD.
Condiciones de ejecución: El usuario no debe estar autenticado.
Entradas: <ul style="list-style-type: none"> • <i>Dimension ID.</i> • <i>Dimension Name.</i> • <i>Dimension Description.</i>

Resultado esperado: Se inserta una nueva dimensión a la base de datos.



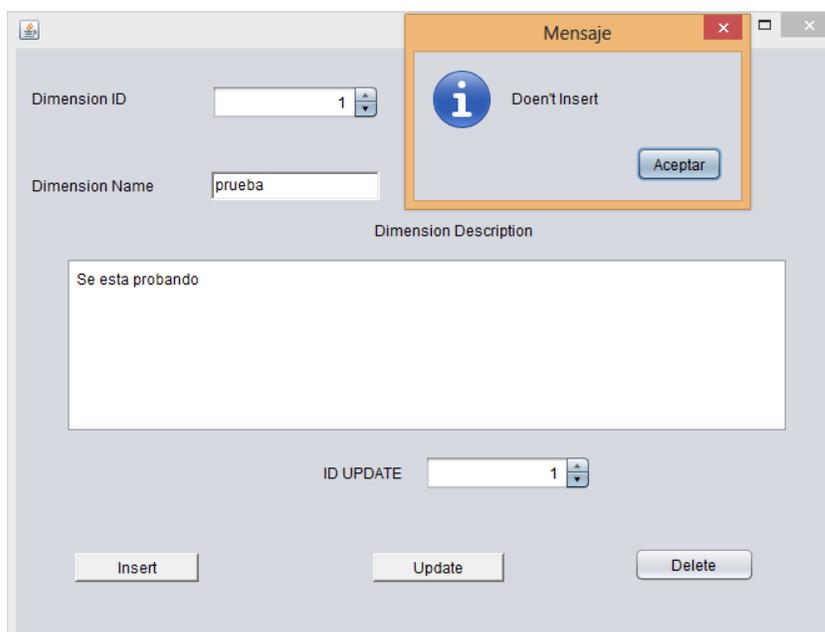
Evaluación: Satisfactorio

En la siguiente prueba se comprueba el proceso de inserción de una dimensión que ya se encuentra registrada en la base de datos.

Tabla 19 Caso de Prueba para la tabla *Dimension*.

Casos de Prueba
Número de caso de prueba : 02
Nombre del Caso de prueba: Insertar Dimension con campos incorrectos.
Descripción de la prueba: A través de la capa de acceso a datos se accede a la interfaz de inserción de una dimensión y se trata de insertar una dimensión cuyo identificador ya existe en la base de datos.
Condiciones de ejecución: El usuario no debe estar autenticado.
Entradas: <ul style="list-style-type: none"> • <i>Dimension ID.</i> • <i>Dimension Name.</i> • <i>Dimension Description.</i>

Resultado esperado: El sistema alerta al usuario de que no se pudo insertar la dimensión deseada.



Evaluación: Satisfactorio.

Caso de Prueba para la Tabla *Competence*.

En la siguiente prueba se verifica el proceso de inserción de una competencia al llenar todos los campos correctamente.

Tabla 20 Caso de Prueba para la Tabla *Competence*.

Casos de Prueba
Número de caso de prueba : 03
Nombre del Caso de prueba: Insertar competencia con campos correctos.
Descripción de la prueba: A través de la capa de acceso a datos se accede a la interfaz de inserción de una competencia y se llenan todos los campos necesarios para guardar la nueva competencia en la BD.
Condiciones de ejecución: El usuario no debe estar autenticado. Debe existir al menos una dimensión y un grado de similitud almacenados en la base de datos.

Entradas:

- *Competence ID.*
- *Proficiency Name.*
- *Dimension ID.*
- *Similarity Degree ID.*
- *Competence Description.*

Resultado esperado: Se inserta una nueva competencia a la base de datos.

Evaluación: Satisfactorio

En la siguiente prueba se verifica el proceso de inserción de una competencia al intentar insertar identificadores del grado de similitud y de la dimensión asociados a esta competencia, que no existen.

Tabla 21 Caso de Prueba para la Tabla *Competence*.

Casos de Prueba
Número de caso de prueba : 04
Nombre del Caso de prueba: Insertar competencia con campos incorrectos.

Descripción de la prueba: A través de la capa de acceso a datos se accede a la interfaz de inserción de una competencia y se insertan identificadores del grado de similitud y de la dimensión asociadas a esta competencia, que no existen.

Condiciones de ejecución: El usuario no debe estar autenticado. Debe existir al menos una dimensión y un grado de similitud almacenados en la base de datos.

Entradas:

- Competence ID.
- Proficiency Name.
- Dimension ID.
- Similarity Degree ID.
- Competence Description.

Resultado esperado: El sistema alerta al usuario de que no se pudo insertar la competencia deseada.

The screenshot shows a web application window with several input fields and buttons. The fields are: 'Competence ID' with a dropdown menu showing '100', 'Proficiency Name' with a text box containing 'prueba', 'Dimension ID' with a dropdown menu showing '250', and 'Similarity Degree ID' with a dropdown menu showing '300'. Below these is a 'Competence Description' text area containing 'Prueba'. At the bottom right, there is an 'ID UPDATE' field with a dropdown menu showing '0'. At the bottom center, there are three buttons: 'Insert', 'Update', and 'Delete'. A modal dialog box titled 'Mensaje' is overlaid on the right side of the window, containing an information icon, the text 'Don't Insert', and an 'Aceptar' button.

Evaluación: Satisfactorio

Caso de Prueba para la Tabla *Evaluator*.

En la siguiente prueba se verifica el proceso de inserción de un instrumento evaluador, al introducir todos los campos necesario para realizar esta acción.

Capa de acceso a datos a través del Mapeo Relacional de Objetos Difusos en la herramienta para la evaluación de competencias ANAELIX

Tabla 22 Caso de Prueba para la Tabla *Evaluator*.

Casos de Prueba
Número de caso de prueba : 05
Nombre del Caso de prueba: Insertar instrumento evaluador con campos correctos.
Descripción de la prueba: A través de la capa de acceso a datos se accede a la interfaz de inserción de un instrumento evaluador y se llenan todos los campos necesarios para guardar el nuevo instrumento evaluador en la BD.
Condiciones de ejecución: El usuario no debe estar autenticado. Debe existir al menos una competencia almacenada en la base de datos.
Entradas: <ul style="list-style-type: none"> • <i>Evaluator ID.</i> • <i>Question Count.</i> • <i>Random Question.</i> • <i>Question Time.</i> • <i>Total Time.</i> • <i>Evaluator Name</i> • <i>Status</i> • <i>Created By</i> • <i>Competence ID</i> • <i>CertaintyT</i> • <i>Certainty1</i> • <i>Certainty2</i> • <i>Certainty3</i> • <i>Certainty4</i>

Resultado esperado: Se inserta un nuevo instrumento evaluador a la base de datos.

Evaluación: Satisfactorio

En la siguiente prueba se verifica el proceso de inserción de un instrumento evaluador, al introducir campos incorrectos.

Tabla 23 Caso de Prueba para la Tabla *Evaluator*.

Casos de Prueba
Número de caso de prueba : 06
Nombre del Caso de prueba: Insertar instrumento evaluador con campos incorrectos.
Descripción de la prueba: A través de la capa de acceso a datos se accede a la interfaz de inserción de un instrumento evaluador y algunos campos que no deben estar nulos se dejan sin llenar.
Condiciones de ejecución: El usuario no debe estar autenticado. Debe existir al menos una competencia almacenada en la base de datos.

Entradas:

- *Evaluator ID.*
- *Question Count.*
- *Random Question.*
- *Question Time.*
- *Total Time.*
- *Evaluator Name*
- *Status*
- *Created By*
- *Competence ID*
- *CertaintyT*
- *Certainty1*
- *Certainty2*
- *Certainty3*
- *Certainty4*

Resultado esperado: El sistema alerta al usuario de que no se pudo insertar el instrumento evaluador deseado.

The screenshot shows a web application interface for creating an evaluator instrument. The form contains the following fields and controls:

- Evaluator ID:** Spin box with value 0.
- Question_Count:** Spin box with value 0.
- Question_Time:** Spin box with value 0.
- Total_Time:** Spin box with value 0.
- Evaluator Name:** Text input field.
- Status:** Dropdown menu with value 1.
- Random_Question:** Dropdown menu with value 1.
- Created By:** Text input field.
- Competence ID:** Spin box with value 0.
- CertaintyT:** Text input field with value 30.
- Certainty1:** Text input field with value 40.
- Certainty2:** Text input field with value 50.
- Certainty3:** Text input field with value 60.
- Certainty4:** Text input field with value 70.
- Evaluator Description:** Large text area.
- ID UPDATE:** Spin box with value 0.

A modal dialog box titled "Mensaje" is overlaid on the form, displaying an information icon and the text "Doent insert" (sic). The dialog has "Aceptar" and "Insert" buttons.

Evaluación: Satisfactorio

Caso de prueba para la traducción de la consulta difusa.

En la siguiente prueba se verifica el proceso de traducción de una consulta difusa, al introducir todos los parámetros necesarios para realizar dicha consulta correctamente.

Tabla 24 Caso de Prueba para la Ejecución de una consulta difusa

Casos de Prueba
Número de caso de prueba : 07
Nombre del Caso de prueba: Traducir Consulta Difusa con datos correctos.
Descripción de la prueba: A través de la capa de acceso a datos se accede a la interfaz de traducción de la consulta difusa y se llenan todos los campos correctamente.
Condiciones de ejecución: El usuario no debe estar autenticado. Debe existir la tabla de la cual se desean seleccionar los datos, esa tabla debe tener una variable difusa de lo contrario la consulta no es difusa, también deben estar definidas las etiquetas en la base de datos asociadas a esa variable difusa.
Entradas: <ul style="list-style-type: none"> • <i>SELECT</i> • <i>FROM.</i> • <i>FUZZY VAR.</i> • <i>COMPARATOR.</i> • <i>ETIQUETA.</i> • <i>THOLD</i> • <i>CONDITIONS</i>

Resultado esperado: Se ejecuta la consulta difusa y se muestran los resultados arrojados por la base de datos.

EVAL...	EVAL...	STAT...	STA...	END...	PAS...	USE...	MAX...	PAS...	CRE...	QUE...	TOT...	EVAL...	PER...
1	55	1	2014...	2014...	1	1	12	1	2014...	5	10	1	1

Evaluación: Satisfactorio

En la siguiente prueba se verifica el proceso de ejecución de una consulta difusa, al introducir campos incorrectos.

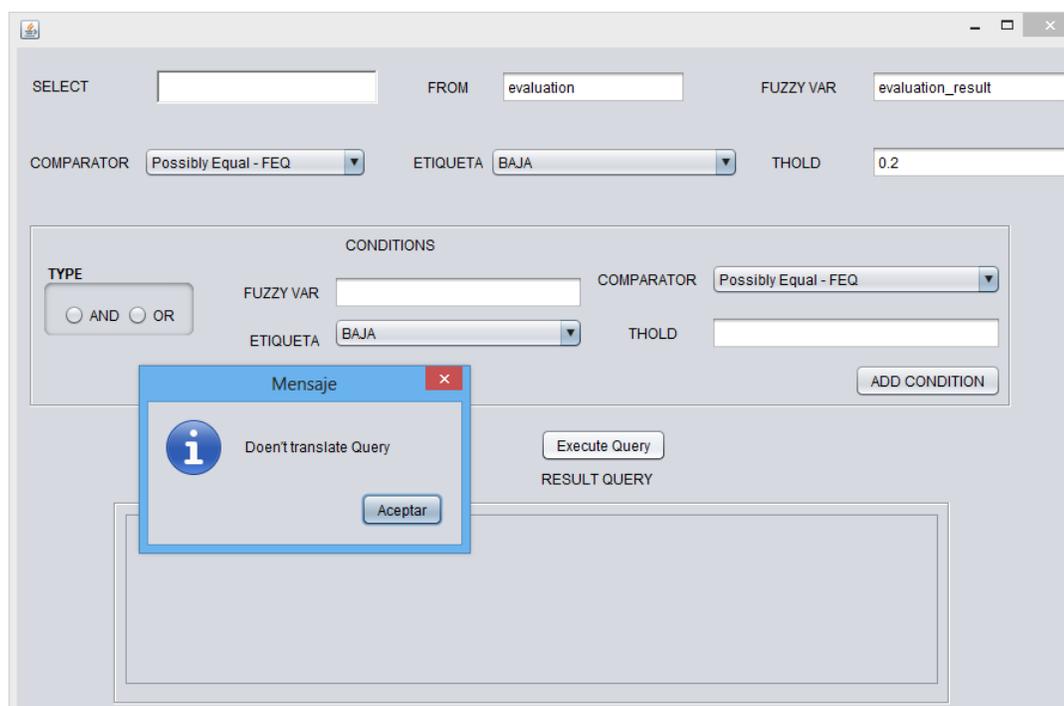
Tabla 25 Caso de Prueba para la ejecución de una consulta difusa.

Casos de Prueba
Número de caso de prueba : 08
Nombre del Caso de prueba: Traducir consulta difusa con campos nulos.
Descripción de la prueba: Se accede a la interfaz de traducción de la consulta difusa y se dejan campos nulos.
Condiciones de ejecución: El usuario no debe estar autenticado. Debe de existir la tabla de la cual se desean seleccionar los datos, esa tabla debe tener una variable difusa de lo contrario la consulta no es difusa, también deben estar definidas las etiquetas en la base de datos asociadas a esa variable difusa.

Entradas:

- *SELECT*
- *FROM.*
- *FUZZY VAR.*
- *COMPARATOR.*
- *ETIQUETA.*
- *THOLD*
- *CONDITIONS*

Resultado esperado: El sistema alerta al usuario de que no se pudo traducir la consulta.



Evaluación: Satisfactorio

3.2.1 Resultados esperados.

Para realizar las pruebas de aceptación se realizaron 4 iteraciones en las cuales se detectaron varios tipos de errores, principalmente los siguientes:

- No se insertaba correctamente en ninguna de las tablas.

Capa de acceso a datos a través del Mapeo Relacional de Objetos Difusos en la herramienta para la evaluación de competencias ANAELIX

- No se ejecutaba la consulta difusa, pues la BDD no tenía definida ninguna etiqueta lingüística.

Se realizó una primera iteración donde se detectaron 5 no conformidades y se les dieron solución, posteriormente se realizó otra iteración donde se detectaron 3 no conformidades y fueron solucionadas, en una tercera iteración se detectó solamente una no conformidad y finalmente se realizó una cuarta iteración, donde se pudo observar como los errores detectados en las pasadas iteraciones se erradicaron completamente haciendo corresponder las funcionalidades implementadas con los requisitos funcionales del software.

3.3 Consideraciones del capítulo

En este capítulo se realizaron pruebas de funcionamiento a la aplicación, dando cumplimiento a los siguientes aspectos:

- El análisis de la integridad de los datos, permitiendo realizar la validación teórica del diseño, comprobando que cumple con los parámetros de integridad de entidad, de dominio y referencial.
- El correcto funcionamiento de la aplicación, lo cual se logró comprobar mediante las pruebas de aceptación realizadas a las interfaces confeccionadas, obteniendo como resultado en la última iteración correspondencia entre las funcionalidades desarrolladas y los requisitos funcionales antes descritos.

CONCLUSIONES GENERALES

Al finalizar el presente trabajo de diploma se logró:

- El mapeo relacional de objetos difusos, sin afectar el tratamiento de la incertidumbre asociada al proceso de evaluación por competencias, eliminando la necesidad de poseer conocimientos de FSQL para interactuar con la base de datos difusa, facilitando de esta forma el mantenimiento futuro de la herramienta ANAELIX.
- Aprovechar la flexibilidad de las consultas difusas a través del mapeo relacional.
- El desarrollo de una capa de acceso a datos que elimina la dependencia directa entre el sistema ANAELIX y la base de datos difusa.
- Crear una capa de acceso a datos que permite la integración de cualquier sistema con la BDD a la cual se hace referencia en la presente investigación.

RECOMENDACIONES

Al culminar el desarrollo del presente trabajo de diploma se recomienda:

- Implementar la traducción de consultas que utilicen cuantificadores difusos universales y existenciales.

BIBLIOGRAFÍA

1. **M, Zavala.** *Las competencias del profesorado universitario.* Madrid: Narcea : s.n., 2003.
2. **Pérez, Lorena Pita.** *Sistema para la gestión del proceso de evaluación por .* La Habana : s.n., 2013.
3. **Galindo, J., Urrutia, A. y Piattini, M.** *Fuzzy Databases: Modeling, Design and Implementation.* Hershey : Idea Group Publishing, 2006.
4. **Gualavisí, Pedro.** [En línea] 16 de mayo de 2012. [Citado el: 29 de noviembre de 2013.] <http://www.buenastareas.com/ensayos/Base-De-Datos-Difusa/4249386.html>.
5. **Fonseca Reyna, Yuniór César, y otros.** *Una mirada a las bases de datos difusas.* 3, La Habana : Grupo Editorial "Ediciones Futuro", 2012, Vol. 6. ISSN: 1994-1536.
6. **Oliva Moreno, Rafael Francisco y Galindo Gómez, José.** *Visual FSQL: Gestión visual de bases de datos difusas en Oracle a través de Internet usando FSQL.* Málaga : s.n., 2003.
7. **Rodríguez, Luis Cuevas.** *Modelo Difuso de Bases de Datos Objeto-Relacional: Propuesta de Implementación en Software Libre.* s.l. : Editorial de la Universidad de Granada, 2009. ISBN: 978-84-692-2251-5.
8. **Galindo Gómez, José.** *Tratamiento de la Imprecisión en Base de Datos Relacionales: Extensión del Modelo de Adaptación de los SGBD actuales.* Granada : s.n., 2000.
9. **Gámez, Evelyn López.** *Desarrollo de una base de datos difusa para el Sistema de Evaluación de Competencias del proyecto Talenmático.* LA Habana : s.n., 2012.
10. **Ibarra, Miguel.** Scribd. [En línea] 2011. [Citado el: 27 de Enero de 2014.] <http://es.scribd.com/doc/51670531/sesion-21-capa-abstraccion-base-de-datos>.
11. **Gómez, Edgar J.** [En línea] 15 de mayo de 2013. [Citado el: 13 de diciembre de 2013.] <http://edgargomez.es/que-es-un-framework/>.
12. **Junta de Andalucía.** Marco de Desarrollo de la Junta de Anadalucía. [En línea] 2013. <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/180>.

Capa de acceso a datos a través del Mapeo Relacional de Objetos Difusos en la herramienta para la evaluación de competencias ANAELIX

13. **Talent Test.** Talent-Test. [En línea] 2012. [Citado el: 12 de diciembre de 2013.] <http://www.talent-test.com.mx /requisitos.html>. 2012.
14. **Crvalho, L. R.** *Planejamento da alocação de recursos humanos em Ambientes de desenvolvimento de software orientados à Organização*. Rio de Janeiro, Brasil : s.n., 2003.
15. **Abreu, Ana Lilian Infante, y otros.** *TEAMSOFTE: Sistema para la gestión del trabajo en equipo en el desarrollo de proyectos de software*. [Documento] La Habana : s.n., 2008.
16. **Muñoz, V., Hernández, D. y Neuland, L.** *Sistema para la gestión del trabajo en equipo en el desarrollo de proyectos de software*. La Habana : s.n., 2007.
17. **Wong Cruz, Christian J. y Flores Velarde, Miluska Yamile.** *Fuzzy Queries. Un framework para realizar consultas difusas*. Arequipa, Perú : s.n., 2005.
18. **Canós, José H., Letelier, Patricio y Penadés, Maria Carmen.** *Métodologías Ágiles en el Desarrollo de Software*. Valencia : s.n., 2010.
19. **Flores Díaz, Mirian Milagros.** [En línea] 2012. [Citado el: 25 de noviembre de 2013.] <http://www.usmp.edu.pe/publicaciones/boletin/fia/info49/articulos/RUP%20vs.%20XP.pdf>.
20. **González Cornejo, José Enrique.** Docirs. [En línea] 2010. [Citado el: 26 de noviembre de 2013.] <http://www.docirs.cl/uml.htm>.
21. **Galindo, Leidy.** [En línea] 18 de noviembre de 2011. [Citado el: 17 de enero de 2014.] <http://www.slideshare.net/legalindo/herramientas-case-10210691>.
22. **Lenguajes de Programación.** Lenguajes de Programación. [En línea] 2009. [Citado el: 20 de noviembre de 2013.] <http://www.lenguajes-de-programacion.com/programacion-java.shtml>.
23. **WordPress.** almamc. [En línea] 25 de Abril de 2013. [Citado el: 20 de Febrero de 2014.] <http://almamc.wordpress.com/2013/01/25/httpalmamc-wordpress-com20130125166/>.
24. **Ramírez, Iván.** Netbeans. [En línea] 22 de octubre de 2012. [Citado el: 10 de diciembre de 2013.] <http://netbeans-ide.softonic.com/>.

25. **Lasic, Maga.** [En línea] 1 de agosto de 2013. [Citado el: 2 de diciembre de 2013.] <http://www.slideshare.net/MagaLasic/presentacion-eclipse-grupo-6> .
26. **King, Gavin, y otros, y otros.** [En línea] 15 de septiembre de 2010. [Citado el: 12 de diciembre de 2013.] http://docs.jboss.org/hibernate/core/3.5/reference/es-ES/html_single/.
27. **Junta de Andalucía.** Marco de Desarrollo de la Junta de Andalucía. [En línea] 2012. [Citado el: 6 de marzo de 2014.] <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/97>.
28. **Junta de Andalucía.** Marco de Desarrollo de la Junta Andalucía. [En línea] 2012. [Citado el: 21 de marzo de 2014.] <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/178>.
29. **Aguero Mirabal, Lisseth .** [En línea] 21 de octubre de 2012. [Citado el: 27 de febrero de 2014.] <http://www.slideshare.net/Lismirabal/requerimientos-funcionales-y-no-funcionales>.
30. **Sun Microsystems, Inc.** Oracle Technology Network. [En línea] 2002. [Citado el: 28 de febrero de 2014.] <http://www.oracle.com/technetwork/java/dataaccessobject-138824.html>.
31. **Microsoft .** Developer Network. [En línea] 2014. [Citado el: 8 de marzo de 2014.] <http://msdn.microsoft.com/es-es/library/ms184276.aspx>.
32. **Microsoft.** Microsoft SQL Server. [En línea] 2008. <http://technet.microsoft.com/es-es/library/ms184276%28v=sql.105%29.aspx..>
33. **Globe Testing.** Software Testing and ALM done Right. [En línea] 14 de agosto de 2012. <http://www.globetesting.com/2012/08/pruebas-de-caja-negra/>.

GLOSARIO DE TÉRMINOS

UCI: Universidad de las Ciencias Informáticas.

ORM: Objeto de Mapeo Relacional

Difuso: Ambiguo, vago, impreciso.

BD: Base de Datos.

BDD: Base de Datos Difusa.

FSQL: Fuzzy Structured Query Language, Lenguaje de Consulta Estructurado Difuso.

FIRST: Fuzzy Interface for Relational SysTems, Interfaz Difuso para Sistemas Relacionales.

FMB: Fuzzy Metaknowledge Base, Base de Metaconocimiento Difuso.

Atributo: Es cada una de las cualidades, propiedades o características de un elemento.

Clase: Es la declaración o abstracción de objetos, lo que significa, que una clase es la definición de un objeto.

SGBD: Sistema Gestor de Base de Datos.

UML: Lenguaje Unificado de Modelado.

ANEXOS

Anexo 1. Historias de Usuario

Historia de Usuario	
Número: 5	Usuario: Aylín
Nombre historia: Insertar Requerimiento.	
Referencia: RF#10	
Prioridad en el negocio: Alta	Nivel de complejidad: Alta
Tiempo de Estimación: 1	Iteración asignada: 2
Descripción: En la capa de acceso a datos para insertar un requerimiento se deben llenar todos los campos necesarios y luego se guardan en la BD los cambios con el nuevo requerimiento.	
Observaciones:	

Historia de Usuario	
Número: 6	Usuario: Aylín
Nombre historia: Modificar Requerimiento.	
Referencia: RF#11	
Prioridad en el negocio: Alta	Nivel de complejidad: Alta
Tiempo de Estimación: 1	Iteración asignada: 2
Descripción: En la capa de acceso a datos para modificar un requerimiento se debe introducir el identificador del requerimiento que se desea modificar, con todos los campos para actualizar y luego se guardan en la BD los cambios efectuados sobre el requerimiento especificado.	
Observaciones:	

Historia de Usuario

Capa de acceso a datos a través del Mapeo Relacional de Objetos Difusos en la herramienta para la evaluación de competencias ANAELIX

Número: 7	Usuario: Aylín
Nombre historia: Eliminar Requerimiento.	
Referencia: RF#12	
Prioridad en el negocio: Alta	Nivel de complejidad: Alta
Tiempo de Estimación: 1	Iteración asignada: 2
Descripción: En la capa de acceso a datos para eliminar un requerimiento se debe introducir el identificador del requerimiento que se desea eliminar, y luego se guardan en la BD los cambios efectuados, o sea se elimina el requerimiento especificado.	
Observaciones:	

Historia de Usuario	
Número: 8	Usuario: Aylín
Nombre historia: Insertar Evaluación.	
Referencia: RF#16	
Prioridad en el negocio: Alta	Nivel de complejidad: Alta
Tiempo de Estimación: 1	Iteración asignada: 2
Descripción: En la capa de acceso a datos para insertar una nueva evaluación, se deben llenar todos los campos necesarios y luego se guarda en la BD la nueva evaluación insertada.	
Observaciones:	

Historia de Usuario	
Número: 9	Usuario: Aylín
Nombre historia: Modificar Evaluación.	
Referencia: RF#17	
Prioridad en el negocio: Alta	Nivel de complejidad: Alta
Tiempo de Estimación: 1	Iteración asignada: 2
Descripción: En la capa de acceso a datos para modificar una evaluación, se introduce el identificador de la evaluación que se desea modificar, con todos los	

Capa de acceso a datos a través del Mapeo Relacional de Objetos Difusos en la herramienta para la evaluación de competencias ANAELIX

campos a modificar, y luego se guardan en la BD todos los cambios efectuados en la evaluación especificada.

Observaciones:

Historia de Usuario

Número: 10

Usuario: Aylín

Nombre historia: Eliminar Evaluación.

Referencia: RF#18

Prioridad en el negocio: Alta

Nivel de complejidad: Alta

Tiempo de Estimación: 1

Iteración asignada: 2

Descripción: En la capa de acceso a datos para eliminar una evaluación, se introduce el identificador de la evaluación que se desea eliminar, y luego se guardan en la BD todos los cambios efectuados, o sea se elimina la evaluación especificada.

Observaciones:

Historia de Usuario

Número: 11

Usuario: Aylín

Nombre historia: Insertar Evaluador.

Referencia: RF#19

Prioridad en el negocio: Alta

Nivel de complejidad: Alta

Tiempo de Estimación: 1

Iteración asignada: 2

Descripción: En la capa de acceso a datos para insertar un nuevo evaluador, se deben llenar todos los campos necesarios y luego se guarda en la BD el nuevo evaluador.

Observaciones:

Historia de Usuario

Capa de acceso a datos a través del Mapeo Relacional de Objetos Difusos en la herramienta para la evaluación de competencias ANAELIX

Número: 12	Usuario: Aylín
Nombre historia: Modificar Evaluador.	
Referencia: RF#20	
Prioridad en el negocio: Alta	Nivel de complejidad: Alta
Tiempo de Estimación: 1	Iteración asignada: 2
Descripción: En la capa de acceso a datos para modificar un evaluador, se introduce el identificador del evaluador que se desea modificar, con todos los campos a modificar, y luego se guardan en la BD todos los cambios efectuados en el evaluador especificado.	
Observaciones:	

Historia de Usuario	
Número: 13	Usuario: Aylín
Nombre historia: Eliminar Evaluador.	
Referencia: RF#21	
Prioridad en el negocio: Alta	Nivel de complejidad: Alta
Tiempo de Estimación: 1	Iteración asignada: 2
Descripción: En la capa de acceso a datos para eliminar un evaluador, se introduce el identificador del evaluador que se desea eliminar, y luego se guardan en la BD todos los cambios efectuados, o sea se elimina el evaluador especificado.	
Observaciones:	

Historia de Usuario	
Número: 14	Usuario: Aylín
Nombre historia: Insertar Grado de Similitud.	
Referencia: RF#25	
Prioridad en el negocio: Alta	Nivel de complejidad: Alta
Tiempo de Estimación: 1	Iteración asignada: 2

Descripción: En la capa de acceso a datos para insertar un nuevo Grado de Similitud, se deben llenar todos los campos necesarios y luego se guarda en la BD el nuevo Grado de Similitud.
Observaciones:

Historia de Usuario	
Número: 15	Usuario: Aylín
Nombre historia: Modificar Grado de Similitud.	
Referencia: RF#26	
Prioridad en el negocio: Alta	Nivel de complejidad: Alta
Tiempo de Estimación: 1	Iteración asignada: 2
Descripción: En la capa de acceso a datos para modificar un Grado de Similitud, se introduce el identificador del Grado de Similitud que se desea modificar, con todos los campos a modificar, y luego se guardan en la BD todos los cambios efectuados en el Grado de Similitud especificado.	
Observaciones:	

Historia de Usuario	
Número: 16	Usuario: Aylín
Nombre historia: Eliminar Grado de Similitud.	
Referencia: RF#27	
Prioridad en el negocio: Alta	Nivel de complejidad: Alta
Tiempo de Estimación: 1	Iteración asignada: 2
Descripción: En la capa de acceso a datos para eliminar un Grado de Similitud, se introduce el identificador del Grado de Similitud que se desea eliminar, y luego se guardan en la BD todos los cambios efectuados, o sea se elimina el Grado de Similitud especificado.	

Observaciones:

Historia de Usuario

Número: 17

Usuario: Aylín

Nombre historia: Insertar Dimensión.

Referencia: RF#4

Prioridad en el negocio: Media

Nivel de complejidad: Media

Tiempo de Estimación: 1/2

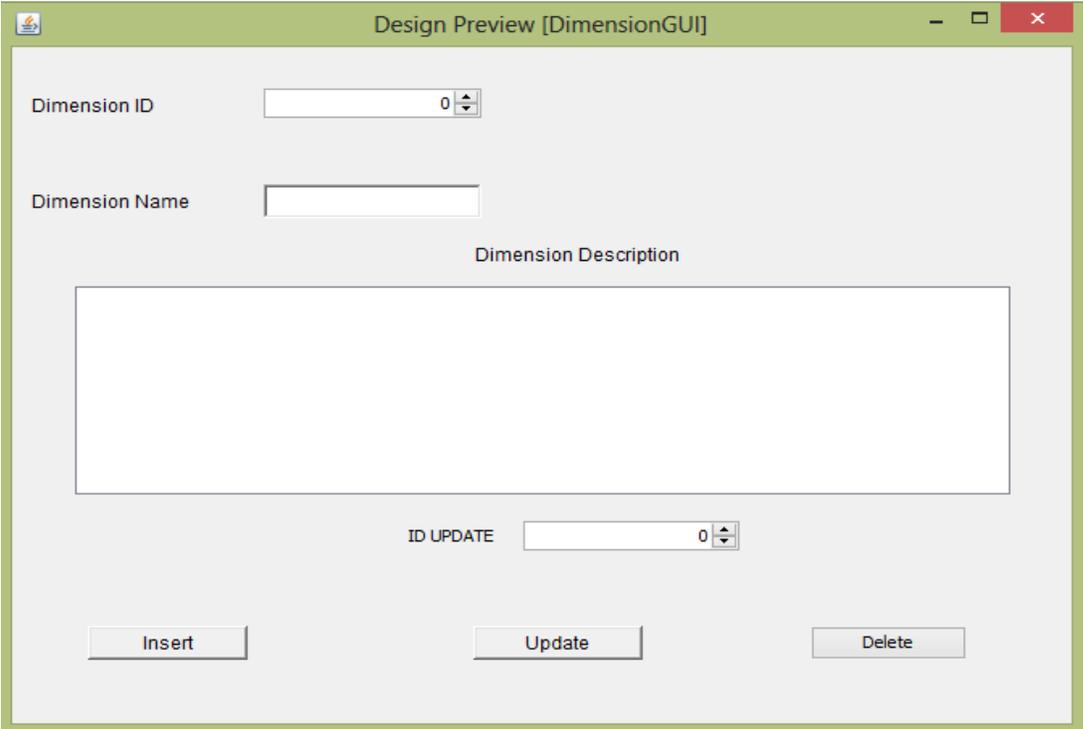
Iteración asignada: 3

Descripción: En la capa de acceso a datos para insertar una nueva dimensión, se deben llenar todos los campos necesarios y luego se guarda en la BD la nueva dimensión insertada.

Observaciones:

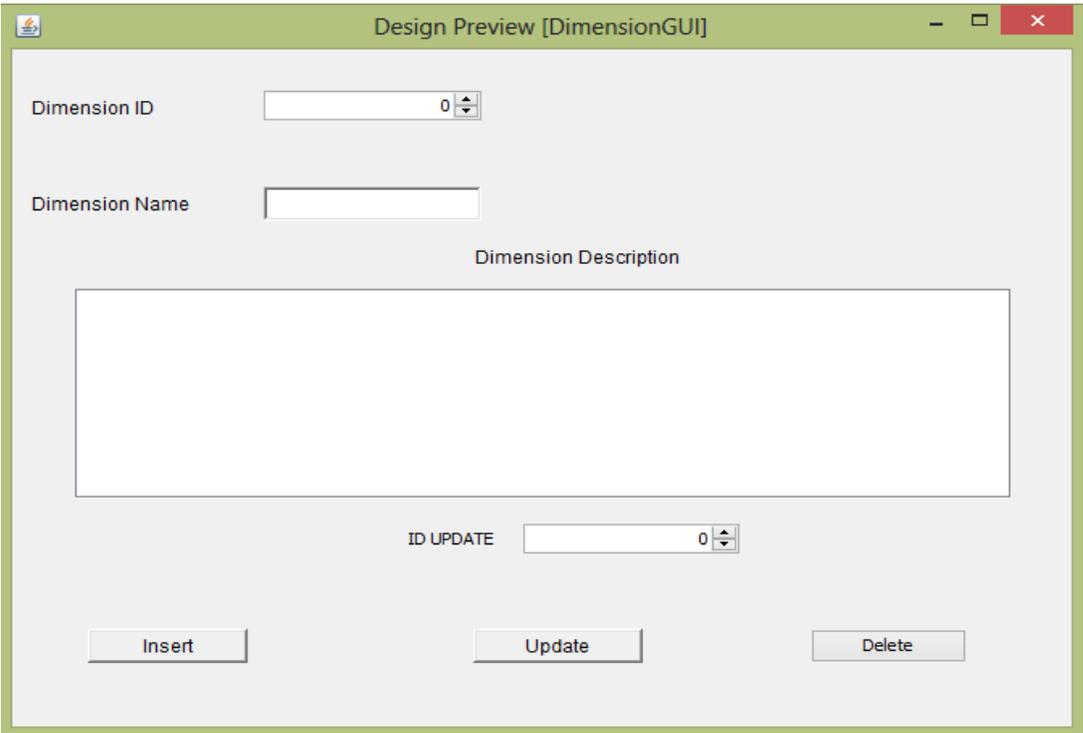
The screenshot shows a window titled "Design Preview [DimensionGUI]". Inside the window, there is a form with the following elements:

- Dimension ID:** A text box containing the value "0" with a small up/down arrow icon to its right.
- Dimension Name:** An empty text box.
- Dimension Description:** A large, empty rectangular text area.
- ID UPDATE:** A text box containing the value "0" with a small up/down arrow icon to its right.
- Buttons:** Three buttons are located at the bottom of the form: "Insert", "Update", and "Delete".

Historia de Usuario	
Número: 18	Usuario: Aylín
Nombre historia: Modificar Dimensión.	
Referencia: RF#5	
Prioridad en el negocio: Media	Nivel de complejidad: Media
Tiempo de Estimación: 1/2	Iteración asignada: 3
<p>Descripción: En la capa de acceso a datos para modificar una dimensión, se introduce el identificador de la dimensión que se desea modificar, con todos los campos a modificar, y luego se guardan en la BD todos los cambios efectuados en la dimensión especificada.</p>	
<p>Observaciones:</p> 	

Historia de Usuario	
Número: 19	Usuario: Aylín
Nombre historia: Eliminar Dimensión.	

Capa de acceso a datos a través del Mapeo Relacional de Objetos Difusos en la herramienta para la evaluación de competencias ANAELIX

Referencia: RF#6	
Prioridad en el negocio: Media	Nivel de complejidad: Media
Tiempo de Estimación: 1/2	Iteración asignada: 3
Descripción: En la capa de acceso a datos para eliminar una dimensión, se introduce el identificador de la dimensión que se desea eliminar, y luego se guardan en la BD todos los cambios efectuados, o sea se elimina la dimensión especificada.	
Observaciones:	
	

Historia de Usuario	
Número: 20	Usuario: Aylín
Nombre historia: Insertar Ocupación.	
Referencia: RF#7	
Prioridad en el negocio: Media	Nivel de complejidad: Media
Tiempo de Estimación: 1/2	Iteración asignada: 3
Descripción: En la capa de acceso a datos para insertar una nueva ocupación, se	

Capa de acceso a datos a través del Mapeo Relacional de Objetos Difusos en la herramienta para la evaluación de competencias ANAELIX

deben llenar todos los campos necesarios y luego se guarda en la BD la nueva ocupación insertada.

Observaciones:

Historia de Usuario

Número: 21	Usuario: Aylín
Nombre historia: Modificar Ocupación.	
Referencia: RF#8	
Prioridad en el negocio: Media	Nivel de complejidad: Media
Tiempo de Estimación: 1/2	Iteración asignada: 3
Descripción: En la capa de acceso a datos para modificar una ocupación, se introduce el identificador de la ocupación que se desea modificar, con todos los campos a modificar, y luego se guardan en la BD todos los cambios efectuados en la ocupación especificada.	
Observaciones:	

Historia de Usuario

Número: 22	Usuario: Aylín
Nombre historia: Eliminar Ocupación.	
Referencia: RF#9	
Prioridad en el negocio: Media	Nivel de complejidad: Media
Tiempo de Estimación: ½	Iteración asignada: 3
Descripción: En la capa de acceso a datos para eliminar una ocupación, se introduce el identificador de la ocupación que se desea eliminar, y luego se guardan en la BD todos los cambios efectuados, o sea se elimina la ocupación especificada.	
Observaciones:	

Historia de Usuario	
Número: 23	Usuario: Aylín
Nombre historia: Insertar Persona.	
Referencia: RF#22	
Prioridad en el negocio: Media	Nivel de complejidad: Media
Tiempo de Estimación: 1/2	Iteración asignada: 3
Descripción: En la capa de acceso a datos para insertar una nueva persona, se deben llenar todos los campos necesarios y luego se guarda en la BD la nueva persona insertada.	
Observaciones:	

Historia de Usuario	
Número: 24	Usuario: Aylín
Nombre historia: Modificar Persona.	
Referencia: RF#23	
Prioridad en el negocio: Media	Nivel de complejidad: Media
Tiempo de Estimación: 1/2	Iteración asignada: 3
Descripción: En la capa de acceso a datos para modificar una persona, se introduce el identificador de la persona que se desea modificar, con todos los campos a modificar, y luego se guardan en la BD todos los cambios efectuados en la persona especificada.	
Observaciones:	

Historia de Usuario	
Número: 25	Usuario: Aylín
Nombre historia: Eliminar Persona.	
Referencia: RF#24	
Prioridad en el negocio: Media	Nivel de complejidad: Media

Tiempo de Estimación: 1/2	Iteración asignada: 3
Descripción: En la capa de acceso a datos para eliminar una persona, se introduce el identificador de la persona que se desea eliminar, y luego se guardan en la BD todos los cambios efectuados, o sea se elimina la persona especificada.	
Observaciones:	

Historia de Usuario	
Número: 26	Usuario: Aylín
Nombre historia: Insertar Pregunta.	
Referencia: RF#13	
Prioridad en el negocio: Media	Nivel de complejidad: Media
Tiempo de Estimación: 1/2	Iteración asignada: 3
Descripción: En la capa de acceso a datos para insertar una nueva pregunta, se deben llenar todos los campos necesarios y luego se guarda en la BD la nueva pregunta insertada.	
Observaciones:	

Historia de Usuario	
Número: 27	Usuario: Aylín
Nombre historia: Modificar Pregunta.	
Referencia: RF#14	
Prioridad en el negocio: Media	Nivel de complejidad: Media
Tiempo de Estimación: 1/2	Iteración asignada: 3
Descripción: En la capa de acceso a datos para modificar una pregunta, se introduce el identificador de la pregunta que se desea modificar, con todos los campos a modificar, y luego se guardan en la BD todos los cambios efectuados en la pregunta especificada.	
Observaciones:	

Capa de acceso a datos a través del Mapeo Relacional de Objetos Difusos en la herramienta para la evaluación de competencias ANAELIX

Historia de Usuario	
Número: 28	Usuario: Aylín
Nombre historia: Eliminar Pregunta.	
Referencia: RF#15	
Prioridad en el negocio: Media	Nivel de complejidad: Media
Tiempo de Estimación: 1/2	Iteración asignada: 3
Descripción: En la capa de acceso a datos para eliminar una pregunta, se introduce el identificador de la pregunta que se desea eliminar, y luego se guardan en la BD todos los cambios efectuados, o sea se elimina la pregunta especificada.	
Observaciones:	

Historia de Usuario	
Número: 29	Usuario: Aylín
Nombre historia: Insertar Tipo de Pregunta.	
Referencia: RF#28	
Prioridad en el negocio: Media	Nivel de complejidad: Media
Tiempo de Estimación: 1/2	Iteración asignada: 3
Descripción: En la capa de acceso a datos para insertar un nuevo Tipo de Pregunta, se deben llenar todos los campos necesarios y luego se guarda en la BD el nuevo Tipo de Pregunta.	
Observaciones:	

Historia de Usuario	
Número: 30	Usuario: Aylín
Nombre historia: Modificar Tipo de Pregunta.	
Referencia: RF#29	
Prioridad en el negocio: Media	Nivel de complejidad: Media

Tiempo de Estimación: 1/2	Iteración asignada: 3
Descripción: En la capa de acceso a datos para modificar un Tipo de Pregunta, se introduce el identificador del Tipo de Pregunta que se desea modificar, con todos los campos a modificar, y luego se guardan en la BD todos los cambios efectuados en el Tipo de Pregunta especificado.	
Observaciones:	

Historia de Usuario	
Número: 31	Usuario: Aylín
Nombre historia: Eliminar Tipo de Pregunta.	
Referencia: RF#30	
Prioridad en el negocio: Media	Nivel de complejidad: Media
Tiempo de Estimación: 1/2	Iteración asignada: 3
Descripción: En la capa de acceso a datos para eliminar un Tipo de Pregunta, se introduce el identificador del Tipo de Pregunta que se desea eliminar, y luego se guardan en la BD todos los cambios efectuados, o sea se elimina el Tipo de Pregunta especificado.	
Observaciones:	

Anexo 2. Tarjetas CRC

Tarjeta CRC	
Clase: Competence.	
Responsabilidad	Colaboración
Funciones que realiza son: <ul style="list-style-type: none"> • getCompetenceld (Permite obtener el identificador de la competencia). • setCompetenceld (Permite modificar el identificador de la competencia). 	<ul style="list-style-type: none"> • Requirement • Similarity Degree • Dimension

<ul style="list-style-type: none"> • getSimilarityDegree (Permite obtener el grado de similitud de esa competencia con otra). • setSimilarityDegree(Permite modificar el grado de similitud de esa competencia con otra). • getDimension (Permite obtener la dimensión que tiene asociada la competencia). • setDimension (Permite modificar la dimensión de la competencia). • getProficiencyName (Permite obtener el nombre de la competencia). • setProficiencyName (Permite modificar el nombre de la competencia). • getCompetenceDescription (Permite obtener la descripción de la competencia). • setCompetenceDescription (Permite modificar el nombre de la competencia). • getRequirements (Permite obtener los requerimientos de esa competencia para ocupar un rol). • setRequirements (Permite modificar los requerimientos de esa competencia para ocupar un rol). • getEvaluators (Permite obtener el instrumento evaluador con el cual se evaluará la competencia). • setEvaluators (Permite modificar el instrumento evaluador con el cual se evaluará la competencia). 	<ul style="list-style-type: none"> • Evaluator
--	---

Tarjeta CRC

Clase: Dimension.

Responsabilidad

Colaboración

<p>Funciones que realiza son:</p> <ul style="list-style-type: none"> • getCompetence (Permite obtener la competencia asociada a la dimensión). • setCompetence (Permite modificar la competencia asociada a la dimensión). • getDimensionId (Permite obtener el identificador de la dimensión). • setDimensionId (Permite modificar el identificador de la dimensión). • getDimensionName (Permite obtener el nombre de la dimensión). • setDimensionName (Permite modificar el nombre de la dimensión). • getDimensionDescription (Permite obtener la descripción de la dimensión). • setDimensionDescription (Permite modificar la descripción de la dimensión). • getCompetences (Permite obtener la competencia asociada a la dimensión). • setCompetences (Permite modificar la competencia asociada a la dimensión). 	<ul style="list-style-type: none"> • Competence
--	--