

# Universidad de las Ciencias Informáticas

## FACULTAD 6



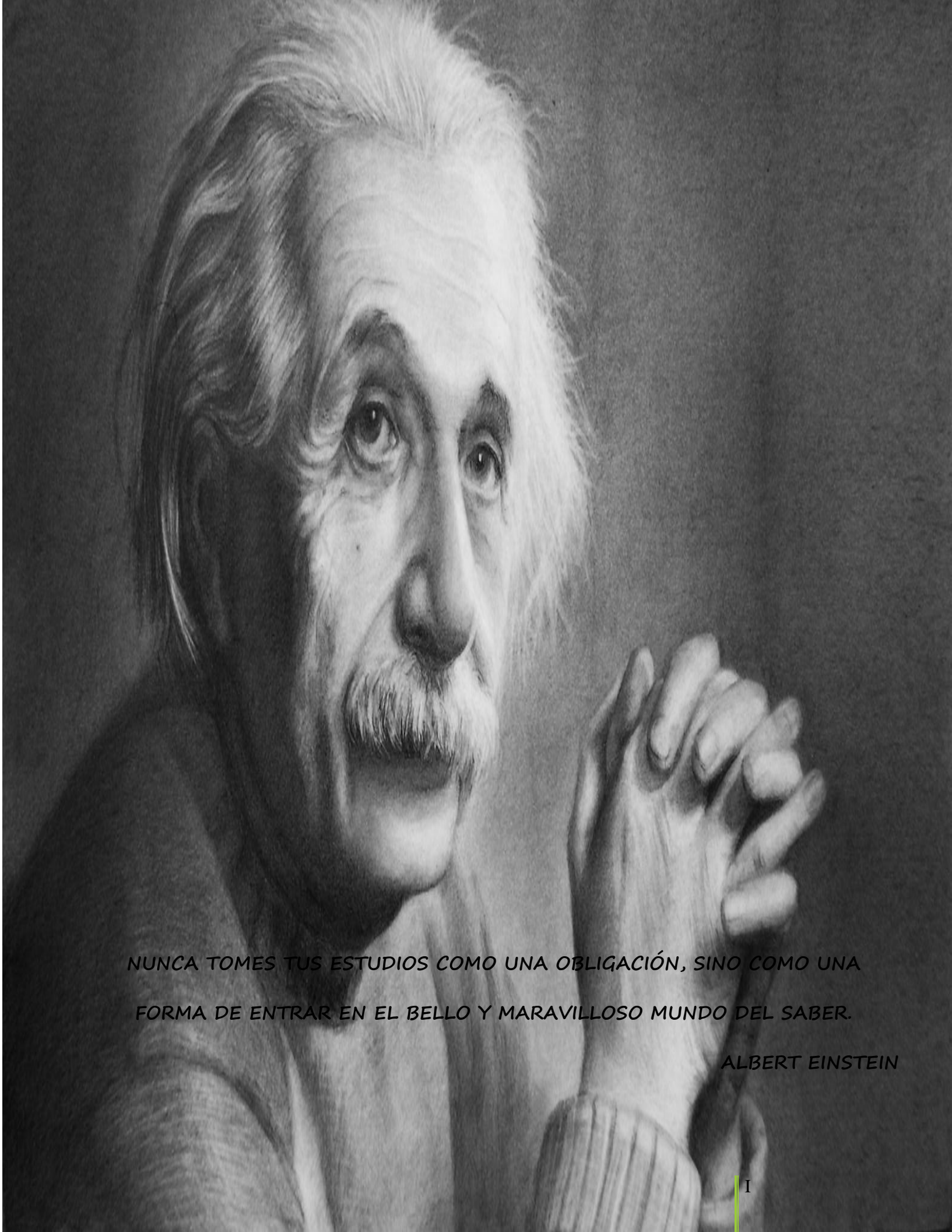
### **Título: Capa de Servicios Web para la aplicación FCM-Decision.**

Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas

**Autor(es):** Aylín Santana Muiño  
Erichel Almanza Castellanos

**Tutor(es):** MSc. Tonysé De La Rosa Martín  
Ing. Oscar Reimar Cedeño Delgado  
Ing. Yosbel Fonseca Mendoza

*La Habana, junio de 2014*  
*“Año 56 de la Revolución”*



*NUNCA TOMES TUS ESTUDIOS COMO UNA OBLIGACIÓN, SINO COMO UNA  
FORMA DE ENTRAR EN EL BELLO Y MARAVILLOSO MUNDO DEL SABER.*

*ALBERT EINSTEIN*

# DECLARACIÓN DE AUTORÍA



## DECLARACIÓN DE AUTORÍA

Declaramos ser los únicos autores de la presente tesis y reconocemos 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 del Autor

Aylín Santana Muiño

---

Firma del Autor

Erichel Almanza Castellanos

---

Firma del Tutor

Ing. Oscar Reimar Cedeño Delgado

---

Firma del Tutor

Ing. Yosbel Fonseca Mendoza

## DATOS DE CONTACTO



### DATOS DE CONTACTO

#### Tutores:

##### **MSc. Tonysé De La Rosa Martin**

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Máster en Bioinformática y Biología Molecular

Email: [tdelarosa@uci.cu](mailto:tdelarosa@uci.cu)

**Ing. Oscar Reimar Cedeño Delgado:** Profesor graduado de Ingeniero en Ciencias Informáticas en el año 2008 en la UCI. Categoría: Instructor. Ha impartido las asignaturas Programación 2, Programación 4, Programación 5 y Práctica Profesional. Se desempeña como Arquitecto del proyecto para la Contraloría General de la República (AUDAT).

Email: [orcedeno@uci.cu](mailto:orcedeno@uci.cu)

**Ing. Yosbel Fonseca Mendoza:** Graduado de Ingeniero en Ciencias Informáticas en el año 2013 en la UCI. Se desempeña como desarrollador del proyecto para la Contraloría General de la República (AUDAT).

Email: [yfons@uci.cu](mailto:yfons@uci.cu)

#### Autores:

##### **Aylín Santana Muiño**

Universidad de las Ciencias Informáticas.

La Habana, Cuba.

Email: [amuino@estudiantes.uci.cu](mailto:amuino@estudiantes.uci.cu)

##### **Erichel Almanza Castellanos**

Universidad de las Ciencias Informáticas.

La Habana, Cuba.

Email: [ealmanza@estudiantes.uci.cu](mailto:ealmanza@estudiantes.uci.cu)

## AGRADECIMIENTOS

*A mi mamá y mi papá por haber sido mi máximo ejemplo, por haberme guiado y alentado siempre a dar lo mejor de mí, por amarme tanto, por apoyarme siempre y por estar presente en cada momento de mi vida.*

*A mis hermanos por haberme dado unos sobrinos tan lindos y por estar siempre conmigo.*

*A mis abuelos que los quiero mucho.*

*A mi abuelito Inginio y a mi tío Humberto que yo sé que desde el cielo ellos están felices por mí.*

*A mi tío Angeló que siempre se ha preocupado mucho por mí y me ha mandado provisiones.*

*A una persona que no es mi familia de sangre pero siempre se ha portado como una tía para mí y tía y amiga siempre será pase lo que pase. Ella siempre me ha estado ayudando en todo lo que ha podido y aunque no pueda estar hoy le quiero agradecer, Yudenia muchísimas gracias por siempre estar apoyándome y creer en mí.*

*A mi novio Yoandí por estar a mi lado 4tro años dándome su apoyo y compañía y aunque no puede estar presente sé que es lo que más hubiera querido, te quiero mi amor.*

*A mis primos, mis tías y mis tíos, en general toda mi familia, por confiar en mí.*

*A Erichel mi compañero de tesis que a pesar de los malos momentos y de las largas discusiones con él, y todas sus loqueras siempre fuimos un equipo.*

*A mis amistades que aunque muchos de ellas no están aquí por su trabajo se merecen un lugarcito en este día por a ver estado conmigo en muchos momentos buenos y malos en estos 5 años no mencionare nombre para que no se pongan celosos.*

*A Melvin no podría dejar de mencionarte a ti, muchísimas gracias por tu sincera amistad, por esta siempre conmigo cuando más necesite de una amiga, gracias por esos momentos, por esta siempre presente.*

## AGRADECIMIENTOS

*A todas aquellas personas que de alguna forma han contribuido a nuestra formación y al desarrollo exitoso de nuestra tesis.*

*Para ustedes todo mi cariño, de corazón, muchas gracias.*

*Aylin Santana Muño*

*Humildemente quiero darle muchas gracias a Dios por ser mi fuerza, mi ayuda, amigo, consejero, consolador en momentos difícil, por haber estado siempre en el transcurso de estos 5 años, por ver cada petición de mi corazón y haber respondido conforme a su voluntad. A ti señor gracias.*

*A mis padres Juan Nelson Almanza y Liury's Virgen Castellanos por ser quienes son, por estar siempre atentos a mis sueños y compartir y apoyar mis ideas. A los dos con mucho amor y cariño que Dios los bendiga mucho.*

*A mi hermano Raydel Almanza por haber influido de una manera especial a la hora de la selección de mi carrera, por estar firme y darme fuerza cuando en primer año me creía incapacitado para comenzar la universidad, por su atención maravillosa como hermano y por enseñarme muchas cosas de lo que hoy soy.*

*A mi cuñada por ser una hermana más por todo el tiempo que lleva en casa ayudándome a ser cada día mejor, por ser la madre de la persona más especial de mi vida: mi sobrina Erika María que es mi motor para seguir viviendo y alcanzar sueños imposibles, a tí tata que Dios este siempre contigo y encuentres siempre gracia delante de él y de los ojos de los hombres.*

*A mi abuela Addis por ser tan especial por todos los momentos que pase a tu lado cuando niño, como olvidar esos viajes al campo a la playa donde todas las vacaciones salir contigo era mi mayor felicidad en ese tiempo, gracias le doy a Dios por aún mantenerte viva y puedas ver el fruto de lo sembrado en mí .*

## AGRADECIMIENTOS

*A mis abuelos Joseito, Fillo y Esther que hoy ya no están presentes físicamente gracias a Dios por el corto tiempo que disfrute la presencia de ustedes y por haber sido tan especial en mi vida a ustedes que Dios los tenga en su presencia.*

*A mi primo Jorge Andrés por ser mi amigo y hermano y por ayudarme tanto en todos los momentos que pudiste.*

*A mis tías y mis primos por haberme ayudado aconsejado y fortalecido en los momentos que me sentí tan solo en la universidad.*

*A Mónica y Marbelly por ser cómo una madre para mí en todo el transcurso de estos años en la universidad a las dos que Dios las bendiga mucho a ustedes y a sus familiares.*

*A todos los compañeros de aula y de apartamento que he tenido por cada uno aportar su grano de arena en mi por haberme ayudado en cada momentos que los he necesitado.*

*A todos los profesores que me han impartido clases gracias por su apoyo y confianza en mí.*

*A mis compañeros de la iglesia muchas gracias a todos por su atención, sus consejos sus oraciones y su ayuda en los momentos que los he necesitado.*

*A todos muchas gracias y que Dios los bendiga mucho*

*Érichel Almanza Castellanos*

## DEDICATORIA

*A mis Padres, Hermanos, a mis abuelos, a mi tío Angelo, a mi novio, y a Yudenia.*

*Aylin Santana Muño*

*A mi mamá, a mi papá, a mi hermano a mi abuela a mi cuñada y a mi sobrina.*

*Érichel Almanza Castellanos*



## Resumen

Con la evolución de la ciencia y las nuevas tecnologías, el desarrollo de la Informática se ha incrementado considerablemente. Este trae consigo el creciente desarrollo de la web y las comunicaciones las cuales han proporcionado el intercambio de datos entre aplicaciones accesibles desde cualquier parte del mundo, pero aun así la comunicación entre aplicaciones desarrolladas en lenguajes de programación y plataformas diferentes es compleja. Valiosas aplicaciones desarrolladas en el mundo son totalmente desconocidas o resultan inaccesibles por esta razón. Tal es el caso de la aplicación de escritorio para el modelado, la simulación y el análisis de los Mapas Cognitivos Difusos, FCM-Decision. Para utilizar esta aplicación se necesita contar con el ejecutable de la misma por lo que es imposible acceder a sus servicios desde distintos lugares al mismo tiempo. Para darle solución a problemáticas como estas surgen los servicios web. La presente investigación se centra en el desarrollo de una capa de servicios web para la aplicación FCM-Decision, que permita centralizar, estandarizar y ofrecer información detallada de las funcionalidades que brinda dicha aplicación. Para el desarrollo se utilizaron tecnologías libres, multiplataforma<sup>1</sup> y una arquitectura por capas. Se utilizó Java como lenguaje de programación, Eclipse 3.2 como entorno de desarrollo y XP como metodología para guiar el proceso de desarrollo de software. Con el fin de garantizar la calidad que requiere el proceso de desarrollo de la capa de servicios web se elaboraron pruebas para validar el correcto funcionamiento de las funcionalidades implementadas.

## PALABRAS CLAVES

Mapas Cognitivos Difusos, FCM-Decision, servicios web.

---

<sup>1</sup> Multiplataforma: Es un término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas. Por ejemplo, una aplicación multiplataforma podría ejecutarse en Windows en un procesador x86, en GNU/Linux. (24)

# ABSTRACT



## ABSTRACT

With the development of science and new technologies, the development of Informatics has increased considerably. This brings the increasing development of the website and communications which have provided data exchange between applications accessible from anywhere in the world, but still the communication between applications developed in programming languages and different platforms and applications is complex. Valuable applications developed in the world are totally unknown or inaccessible for this reason. Such is the case of the desktop application for modeling, simulation and analysis of Fuzzy Cognitive Maps, FCM-Decision. To use this application you need to have the executable so it is impossible to access services from different locations at the same time. For solving problems like those web services arise. This research focuses on the development of a web services layer for FCM-Decision, allowing application to centralize, standardize and provide detailed information on the functionalities that the application offers. For developing this application free technologies, multiplatform, and a layered architecture is used. Java was used as the programming language, Eclipse 3.2 as development environment and XP as a methodology to guide the software development process. In order to guarantee the quality required by the development process of web services layer, tests were developed to validate the correct operation of the implemented functionalities.

## KEYWORDS

Fuzzy Cognitive Maps, FCM-Decision, web services.



## Índice de Contenido

INTRODUCCIÓN .....	1
CAPÍTULO 1: FUNDAMENTOS TEÓRICOS .....	7
1.1 Principales Conceptos.....	7
1.2 Sistemas vinculados al campo de acción .....	9
1.2.1 FCMappers .....	9
1.2.2 MaGraDa.....	9
1.2.3 Herramienta de creación de grafos de caminos para la biblioteca “SceneToolkit” .....	10
1.2.4 FCM-Decisión.....	10
1.3 Metodología y herramientas a utilizar en el desarrollo del servicio web .....	11
1.3.1 Metodología XP:.....	11
<input type="checkbox"/> Las Historias de Usuario .....	12
<input type="checkbox"/> Roles XP.....	12
<input type="checkbox"/> Procesos y prácticas.....	13
<input type="checkbox"/> Algunas de las Ventajas de la Metodología XP son: .....	13
1.3.2 Lenguaje Unificado de Modelado (UML) 2.0.....	13
1.3.3 Herramienta CASE Visual Paradigm 8 .....	14
1.3.4 Entorno de Desarrollo Eclipse 3.2 .....	14
1.3.5 Lenguaje de programación Java .....	15
1.3.6 Middleware Axis2 .....	16
1.3.7 ApacheTomcat 7 .....	16
1.3.8 MySQL Workbench .....	16
1.3.9 Hibernate.....	17
CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA.....	19
2.1 Modelo de dominio.....	19
2.2 Descripción del sistema propuesto.....	21

# ÍNDICE DE CONTENIDO

2.2.1 Las Historias de Usuario .....	21
2.2.2 Plan de iteraciones .....	32
2.3 Modelo de diseño .....	33
2.3.1 Tarjetas CRC .....	33
2.4 Diagramas de Clases .....	36
2.5 Arquitectura base principal de la aplicación.....	38
2.5.1 Patrón de arquitectura: .....	38
2.5.2 Patrones de Diseño: .....	39
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA .....	43
3.1 Capa de Servicios Web para la aplicación FCM-Decision .....	43
3.1.1 Métodos Públicos .....	43
3.1.2 Consumo de los servicios.....	43
3.2 Tareas de ingeniería .....	44
3.3 Estándar de codificación .....	48
3.3.1. Indentación.....	49
3.3.2. Comentarios.....	49
3.3.3. Sentencias .....	50
3.3.4. Espacios en blanco .....	51
3.3.5. Convenciones de nombres.....	51
3.4 Implementación de las historias de usuario.....	52
3.5. Pruebas.....	52
3.5.1. Pruebas de aceptación.....	52
3.5.2. Diseño de casos de prueba y registro de no conformidades.....	54
CONCLUSIONES GENERALES.....	61
RECOMENDACIONES .....	62
REFERENCIAS .....	63
BIBLIOGRAFÍA.....	65

# ÍNDICE DE CONTENIDO



GLOSARIO DE TÉRMINOS .....	67
ANEXOS.....	¡ERROR! MARCADOR NO DEFINIDO.



## Índice de Figuras

Fig. 1 Modelo de Dominio .....	20
Fig. 2 Diagrama de Clases.....	37
Fig. 3 Diagrama de la arquitectura Modelo Vista Controlador .....	39
Fig. 4 Patrón Experto .....	40
Fig. 5 Patrón Creador .....	41
Fig. 6 Alta cohesión .....	41
Fig. 7 Patrón bajo acoplamiento.....	42
Fig. 8 Comentario del método calcular Centralidad.....	50
Fig. 9 Ejemplo correcto de sentencia WHILE .....	50
Fig. 10 Ejemplo correcto de sentencia IF, IF-ELSE.....	50
Fig. 11 Ejemplo de espacios en blanco entre las funciones de la interface .....	51
Fig. 12 Ejemplo correcto de aplicación de espacios en blanco.....	51
Fig. 13 Ejemplo correcto de convención de nombres de las funciones de la interface. ....	51
Fig. 14 Prueba de caja negra.....	53
Fig. 15 Resultados de las pruebas.....	60

## Índice de Tablas

Tabla. 1 Descripción de los conceptos del diagrama de dominio .....	20
Tabla. 2 Historia de usuario Agregar matrices .....	22
Tabla. 3 Historia de usuario Listar matrices. ....	22
Tabla. 4 Historia de usuario Obtener matriz a través de su ID .....	23
Tabla. 5 Historia de usuario Centralidad de factores por id Matriz. ....	24
Tabla. 6 Historia de usuario Centralidad de factores por matriz. ....	25
Tabla. 7 Historia de usuario Realizar simulación por ID matriz.....	26
Tabla. 8 Historia de usuario Realizar simulación por matriz .....	27
Tabla. 9 Historia de usuario obtener resultado de la simulación.....	28
Tabla. 10 Historia de usuario obtener gráfica.....	29
Tabla. 11 Historia de usuario Eliminar matriz. ....	30
Tabla. 12 Lista de reserva del producto. ....	31
Tabla. 13 Plan de iteraciones.....	33
Tabla. 14 Tarjeta CRC Agregar matrices. ....	34
Tabla. 15 Tarjeta CRC Listar matrices. ....	34
Tabla. 16 Tarjeta CRC Obtener según id matriz.....	34
Tabla. 17 Tarjeta CRC Centralidad de Factores por ID matriz .....	34
Tabla. 18 Tarjeta CRC Centralidad de Factores por matriz.....	35
Tabla. 19 Tarjeta CRC obtener resultado de centralidad según ID. ....	35
Tabla. 20 Tarjeta CRC Realizar simulación por ID.....	35
Tabla. 21 Tarjeta CRC Realizar simulación por matriz.....	35
Tabla. 22 Tarjeta CRC Obtener resultado de simulación. ....	36
Tabla. 23 Tarjeta CRC Obtener gráfica.....	36
Tabla. 24 Tarjeta CRC Eliminar matriz.....	36
Tabla. 25 Tarea de Ingeniería Agregar Modelo .....	44

# ÍNDICE DE TABLAS

Tabla. 26 Tarea de Ingeniería Agregar Nodo .....	45
Tabla. 27 Tarea de Ingeniería Agregar Conector .....	45
Tabla. 28 Tarea de Ingeniería Listar Matrices .....	45
Tabla. 29 Tarea de Ingeniería obtener matriz dado un id .....	46
Tabla. 30 Tarea de Ingeniería Centralidad de factores por id matriz .....	46
Tabla. 31 Tarea de Ingeniería Centralidad de factores por matriz .....	46
Tabla. 32 Tarea de Ingeniería Realizar simulación por ID matriz .....	47
Tabla. 33 Tarea de Ingeniería Realizar simulación por matriz .....	47
Tabla. 34 Tarea de Ingeniería Obtener resultados de la simulación .....	47
Tabla. 35 Tarea de Ingeniería Obtener gráfica .....	48
Tabla. 36 Tarea de Ingeniería Eliminar matriz .....	48
Tabla. 37 Secciones para probar la historia de usuario Agregar Matriz .....	54
Tabla. 38 Descripción de variables de la historia de usuario Agregar Matriz .....	56
Tabla. 39 Datos de la historia de usuario Agregar Matriz .....	57
Tabla. 40 No conformidades detectadas en la aplicación del caso de pruebas Agregar matriz, específicamente el escenario agregar modelo. ....	58



## INTRODUCCIÓN

Los comportamientos inteligentes del hombre involucran percepción, razonamiento, aprendizaje, comunicación y actuación en ambientes complejos. Para lograr que las computadoras emulen estos comportamientos obtenidos del conocimiento de expertos, se utiliza la Inteligencia Artificial (IA). La IA, como disciplina encargada de construir procesos, que al ser ejecutados producen resultados con un rendimiento determinada, se basa en las secuencias de entradas recibidas y en el conocimiento almacenado.

En Inteligencia Artificial, la lógica difusa, o borrosa se utiliza para la resolución de problemas variados. Los sistemas basados en lógica difusa imitan la forma en que los humanos toman decisiones, con la ventaja de ser más rápidos. Estos sistemas son generalmente robustos y tolerantes a imprecisiones. (1)

En la actualidad se encuentran problemas prácticos donde es necesario representar el grado de influencia entre conceptos, resultando difícil establecer una relación probabilística entre los mismos. Ante estas circunstancias la lógica difusa constituye una alternativa pues permite representar la causalidad entre conceptos. Para considerar la causalidad desde un punto de vista computacional, se requiere la obtención de modelos causales imprecisos, esto trae consigo la utilización de técnicas de Software de Computadoras. Dentro de estas se encuentran las Redes Bayesianas (RB) y los Mapas Cognitivos Difusos (MCD). (2)

Una red bayesiana se utiliza para representar el conocimiento y el razonamiento probabilístico de forma gráfica. Está representada por grafos formados por dos elementos: los nodos y los arcos. Los nodos representan las variables y los arcos son las relaciones probabilísticas entre los nodos.

Por otro lado los Mapas Cognitivos Difusos son estructuras de grafos difusos utilizados para representar razonamiento causal. Su aplicación es recomendable para amplios dominios del conocimiento, como el político, el militar, la historia, entre otros donde los conceptos y sus relaciones son principalmente difusos. Los Mapas Cognitivos Difusos son una herramienta de representación causal, que están compuestos por conceptos y relaciones causales entre los distintos conceptos, los cuales usan la teoría de lógica difusa para describir su estructura. (3)

En correlación con lo planteado anteriormente los MCD proveen esquemas más realistas para la representación del conocimiento con respecto a las RB. La posibilidad de representar ciclos, retroalimentación y el tiempo se encuentra entre los elementos que permiten una representación más

# INTRODUCCIÓN

realista del conocimiento. Estas características han propiciado que los MCD puedan colaborar en la toma de decisiones y se apliquen en diversas ramas de la ciencia. Se han aplicado en la medicina, en el reconocimiento de patrones, en el control de sistemas, en la visión por ordenadores y en la informática.

En el campo de la informática se han desarrollado diversas herramientas vinculadas con los MCD. Sin embargo uno de los retos actuales de la informática es realizar aplicaciones independientes del hardware y del sistema operativo. Aún, cuando las aplicaciones son cada vez más independientes y accesibles desde cualquier parte del mundo, la heterogeneidad de los sistemas, los cuales están escritos en lenguajes distintos, siendo su interoperabilidad<sup>2</sup> difícil, constituye una problemática. Debido a estas situaciones nace una solución que se nutre de diversas tecnologías, y de alguna forma soluciona el problema de la comunicación entre aplicaciones que se encuentran en ambientes distribuidos; y que se desarrollan absolutamente de forma diferente. Esta solución es acreditada como Servicios Web.

Los servicios web facilitan el intercambio de datos entre aplicaciones de software ejecutadas sobre cualquier plataforma. Permiten que los servicios y software de diferentes compañías que se encuentran en diversos lugares del mundo sean combinados fácilmente para proveer servicios inteligentes. Los servicios web fomentan los estándares y protocolos basados en texto, haciendo más fácil acceder a su contenido y entender su funcionamiento. (4)

Las principales compañías del mundo han comenzado a desarrollar soluciones utilizando la tecnología de los Servicios Web, ejemplo de esto es Microsoft, la cual brinda servicios como el MapPoint.Net, otra compañía es International Business Machines IBM por sus siglas en inglés pues ha implementado una solución basada en los Servicios Web llamada e-Business on Demand, por otro lado en Cuba, específicamente en la Universidad de las Ciencias Informáticas (UCI), se encuentra en desarrollo la aplicación de escritorio para el modelado, la simulación y el análisis de los MCD, FCM-Decision. Esta aplicación forma parte de la investigación de la tesis de maestría de un profesor del departamento PostgreSQL del centro DATEC, la cual se ha utilizado como herramienta de ayuda a la toma de decisiones por otros especialistas. Esta aplicación se puede aplicar en cualquier dominio o área y colabora con la toma de decisiones.

---

<sup>2</sup> La interoperabilidad es la capacidad que tiene un producto o un sistema, cuyas interfaces son totalmente conocidas, para funcionar con otros productos o sistemas existentes o futuros y eso sin restricción de acceso o de implementación. (26)

# INTRODUCCIÓN

La toma de decisiones no es más que elegir entre varias opciones cual es la más conveniente en un determinado momento. Desde el punto de vista laboral, para los médicos, informáticos y otros especialistas, la toma de decisiones es sin duda una de las mayores responsabilidades que debe asumir. La toma de decisiones, considerar un problema y llegar a una conclusión válida, significa que se han examinado todas las alternativas y que la elección ha sido correcta.

En el mundo se han desarrollado aplicaciones que colaboran con la toma de decisiones, algunas de estas son aplicaciones de escritorio donde solo puede acceder un usuario y desde un lugar específico, un ejemplo de estas aplicaciones es FCM-Decisión. Esta es una aplicación de escritorio y resulta necesario para los usuarios que no poseen el ejecutable de la misma acceder a los servicios que brinda al mismo tiempo y desde cualquier lugar. Estas dificultades, unidas a la insuficiente disponibilidad de herramientas web tanto comerciales como libres que brinden soporte al modelado y la simulación de los MCD, son algunas de las limitaciones que se evidencian en la actualidad.

Debido a la problemática planteada anteriormente el **problema de la investigación** es: ¿Cómo realizar el diseño y manejo de las funcionalidades de la aplicación FCM-Decision?

Teniendo en cuenta el problema de la investigación se ha definido como **objeto de estudio**: Las aplicaciones informáticas para la gestión de información de Mapas Cognitivos Difusos.

El estudio se centra en el **campo de acción** definido por: Las aplicaciones que consumen servicios web para la gestión de información de Mapas Cognitivos Difusos.

En correspondencia con el problema planteado el **objetivo general** de esta investigación consiste en: Desarrollar Servicios Web que permitan centralizar, estandarizar y ofrecer información detallada de las funcionalidades que brinda la aplicación FCM-Decision.

Del mismo se derivan los siguientes **objetivos específicos**:

1. Analizar el marco teórico conceptual respecto a las tecnologías actuales para el desarrollo de Servicios Web.
2. Realizar el análisis y diseño de los Servicios Web identificados para centralizar, estandarizar y ofrecer información detallada de las funcionalidades que brinda la aplicación FCM-Decision.

# INTRODUCCIÓN

3. Implementar los Servicios Web identificados para centralizar, estandarizar y ofrecer información detallada de las funcionalidades que brinda la aplicación FCM-Decision.
4. Realizar pruebas que validen el correcto funcionamiento los Servicios Web implementados.

Con el fin de guiar la investigación se definieron las siguientes **preguntas científicas**:

1. ¿Cuáles son las bases que definirán la fundamentación teórica que demostrará el porqué de la realización de la investigación y la propuesta de solución?
2. ¿Cómo guiar el diseño de la propuesta de solución a partir de la descripción de la propuesta de solución?
3. ¿Cómo caracterizar la construcción de la propuesta de solución?
4. ¿Cómo validar la solución construida con el fin de garantizar el correcto funcionamiento de la misma?

Para dar cumplimiento al objetivo planteado, se definen las siguientes **tareas de la investigación**:

1. Revisión bibliográfica, principales elementos teóricos y conceptos que permitan analizar el estado actual de las tecnologías para el desarrollo de servicios web.
2. Análisis de las funcionalidades que ofrece la aplicación FCM-Decision para identificar los servicios web a implementar.
3. Definición de las herramientas y de la metodología de desarrollo de software que va a regir el desarrollo de los Servicios Web.
4. Identificación de los requisitos funcionales y no funcionales.
5. Documentación de los artefactos generados durante los procesos de análisis, diseño e implementación de los Servicios Web según la metodología seleccionada.
6. Implementación de los servicios web identificados.
7. Desarrollo de una aplicación cliente que consuma los Servicios Web desarrollados.
8. Realización de pruebas que validen el correcto funcionamiento de los Servicios Web implementados.

Como resultado de la presente investigación se obtendrá una capa de Servicios Web que permite exponer las principales funcionalidades de la aplicación de escritorio FCM-Decision.

## Los métodos teóricos

Analítico - Sintético

# INTRODUCCIÓN

Su objetivo en la investigación fue analizar las teorías, documentos, manuales, paginas relacionadas con el tema de los mapas cognitivos difusos y sintetizarlo en otras palabras resumirlo para un mejor entendimiento. Permitiendo la extracción de los elementos más importantes que se relacionan con el objeto de estudio.

## Histórico - Lógico

Su objetivo en la investigación fue constatar teóricamente cómo ha evolucionado el tema de los mapas cognitivos difusos y su aplicación en un período de tiempo, se realizó un estudio de la aplicación de escritorio FCM-Decision la cual sirvió de base para el desarrollo de la presente investigación.

## **Los métodos empíricos**

### La entrevista

En la presente investigación se usó la entrevista ya que permite una conversación planificada para obtener información. Su uso constituyó un medio para el conocimiento del tema a continuación algunas preguntas hechas en la entrevista.

¿Utilización de los Mapas cognitivos difusos?

¿Porque es necesario llevar la idea mapas cognitivos difusos a una plataforma web?

Este documento se encuentra estructurado en tres capítulos, los cuales se describen a continuación:

Capítulo 1: En este capítulo se exponen los conceptos y definiciones principales para la comprensión del tema a abordar. Contiene una vista amplia de los principales marcos tecnológicos empleados para soluciones a través de Servicios Web, así como las principales metodologías y herramientas utilizadas en la realización de un Servicio Web, para poder integrar dichos elementos y dar solución al problema en cuestión.

Capítulo 2: En este capítulo, se identifican los requisitos funcionales y no funcionales a implementar en el proceso de desarrollo del servicio web. Se organiza la estructura y el comportamiento del sistema para guiar de una manera más sencilla a los clientes y desarrolladores a su comprensión. Se muestra una descripción de las historias de usuario, se identifican los actores, casos de uso y las relaciones existentes entre estos, proporcionando como resultado la realización del diagrama de casos de uso del sistema.

# INTRODUCCIÓN



Capítulo 3: En este capítulo se describe el modelo de despliegue, se obtienen todos los algoritmos necesarios para la implementación de las nuevas funcionalidades y por último se expone cómo fueron concebidas las pruebas para garantizar la calidad de los servicios web implementados.

## CAPÍTULO 1: FUNDAMENTOS TEÓRICOS

En el presente capítulo se abordarán los conceptos fundamentales relacionados con los Servicios Web, se enfatizará en las características de la metodología utilizada, así como el estado del arte actual para las aplicaciones a utilizar, analizando de cada una de ellas sus ventajas y desventajas a fin de seleccionar las más adecuadas para llevar a cabo la implementación del Servicio Web.

### 1.1 Principales Conceptos

La cantidad y variedad de aplicaciones de la lógica difusa han crecido considerablemente. En dicho trabajo a pesar de no ser utilizada es necesario conocer en qué consiste. Pues es una de las técnicas de software de computadoras. Es un tipo de lógica basada en la teoría de conjuntos y que trata de imitar la forma en que usualmente los humanos toman decisiones.

**Lógica Difusa:** Según Escolano y otros “La lógica fuzzy o “difusa” es una extensión de la lógica clásica diseñada para permitir el razonamiento sobre conceptos imprecisos.” Afirma Escolano y otros “Básicamente, la lógica difusa es una lógica multivariada que permite una graduación continua en el valor de verdad de una proposición, al poder utilizar cualquier valor en el intervalo  $[0, 1]$ .” Pajares y Santos indican que: “Una de las características esenciales de la lógica borrosa es que permite utilizar el lenguaje ordinario como lenguaje de descripción en un ordenador, incorporando así el conocimiento de un experto en una tarea determinada. Introduce la imprecisión y la subjetividad propia de la actividad humana en un procedimiento automatizado. (5)

Al igual que la lógica difusa los mapas cognitivos son una técnica de software de computadoras. Los cuáles tienen un gran peso en el presente trabajo de diploma, por lo que a continuación se enuncia su concepto.

**Mapas Cognitivos:** El mapa cognitivo (MC) es un constructo que abarca aquellos procesos que posibilitan a las personas adquirir, codificar, almacenar, recordar y manipular la información sobre la naturaleza de su entorno. Esta información se refiere a los atributos y localizaciones relativas de la gente y los objetos del entorno, y es un componente esencial en los procesos adaptativos y de toma de decisiones espaciales. (6)

# CAPÍTULO 1: FUNDAMENTOS TEÓRICOS

Los mapas cognitivos son estructuras de grafos difusos que permiten hacer representaciones del conocimiento, ayudando en la toma de decisiones. Los MC y sus modelos causales<sup>3</sup> son imprescindibles en el desarrollo de este trabajo por lo que resulta necesario comprender el concepto de causalidad para obtener entendimiento del tema abordado.

En la presente investigación se desarrolla una capa de servicios web para la aplicación de modelado, simulación y análisis de los MCD, FCM – Decision. Como los servicios web constituyen el núcleo fundamental de la misma, a continuación se enuncia su definición y características.

**Servicios web:** Un servicio web (en inglés, Web Service o Web services) es una tecnología que utiliza un conjunto de protocolos y estándares que se utilizan para intercambiar datos entre distintas aplicaciones de software, desarrolladas en lenguajes de programación diferentes con el objetivo de ofrecer servicios. Estos pueden ser utilizados para intercambiar datos en redes de ordenadores. Los proveedores brindan sus servicios como procedimientos remotos y los usuarios los piden llamando a estos procedimientos a través de la web. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Para proporcionar interoperabilidad y extensibilidad entre las aplicaciones, y que su combinación al mismo tiempo sea posible para realizar operaciones complejas, es necesaria una arquitectura de referencia estándar entre distintas aplicaciones, que interactúan entre sí para presentar información dinámica al usuario.

Tienen como característica que combinan lo mejor de la tecnología de componentes y de la tecnología Web. Presentan una funcionalidad de caja negra que puede ser reutilizada sin preocuparse de cómo es implementada y ello proporciona interfaces bien definidas. La interacción entre el cliente y el proveedor del servicio es empaquetada en unidades autodescriptivas denominadas mensajes. Dicha interacción se describe en función de los mensajes intercambiados. (7)

---

<sup>3</sup> Causales: **Causalidad:** La causalidad es el principio o el origen de algo. El concepto se utiliza para nombrar a la relación entre una causa y su efecto y puede utilizarse en el ámbito de la física, la estadística y la filosofía. De acuerdo al principio de causalidad, todo efecto siempre tiene una causa. El principio de uniformidad agrega que, en idénticas circunstancias, una causa siempre produce el mismo efecto. (25)



## 1.2 Sistemas vinculados al campo de acción

Antes de iniciar el desarrollo de un software es imprescindible investigar sobre sistemas existentes nacionales como internacionales que tengan semejanzas con el software a desarrollar. Actualmente existen aplicaciones que ofrecen soluciones informáticas para el modelado de MCD. La investigación de sistemas semejantes constituye un inicio donde a través del mismo se pueden realizar análisis en cuanto a sus funcionalidades y arquitectura. A continuación se muestran los sistemas analizados.

### **Sistemas internacionales:**

#### **1.2.1 FCMappers**

FCMappers es una red de análisis de la red, un grupo de personas que comparten sus conocimientos, información y pensamientos acerca de la asignación cognitiva difusa. Es una comunidad, una red de científicos interesados en la cartografía cognitiva difusa y la conexión a la red de análisis y modelado de sistemas. Su objetivo es crear una plataforma para compartir experiencias e información sobre mapas cognitivos difusos. Dado las metas que tiene esta comunidad sus fundadores están interesados en desarrollar aún más FCMappers, el primer software libre disponible para el análisis y la visualización de los mapas cognitivos difusos.

#### **1.2.2 MaGraDa**

El paquete de software MaGraDa (Grafos para Matemática Discreta) es una aplicación informática programada en lenguaje Java utilizando como herramienta de desarrollo el Netbeans en su versión 6.9. Está diseñada específicamente para trabajar con grafos dirigidos, no dirigidos, ponderados y no ponderados. La filosofía de MaGraDa permite representar grafos con un máximo de 50 vértices. Este paquete es sencillo y cómodo de manejar, consta de dos pantallas de visualización: una en modo texto donde se trabaja con los datos del grafo y otra en modo gráfico en la que los grafos se pueden observar gráficamente.

Los grafos se crean a través de la interacción del usuario con la aplicación, quien puede crear un grafo manualmente o a través de una matriz de adyacencia y además puede abrir un grafo creado anteriormente con MaGraDa. Con esta se pueden averiguar una serie de características o propiedades básicas, tales como: grado de un vértice, matriz de adyacencia o pesos y ver aristas (o arcos) del grafo. Dispone de algoritmos muy conocidos en el mundo de los grafos tales como, Dijkstra, Floyd–Warshall, Kruskal y Prim.

# CAPÍTULO 1: FUNDAMENTOS TEÓRICOS

MaGraDa, por ser un proyecto sencillo, no se encarga de garantizar que los trazados de los grafos sean estéticamente buenos, por lo que pueden existir solapamientos de nodos y cruces entre aristas. (8)

Estas herramientas no son adecuadas debido a que no se ajustan al problema propuesto, pues no se cuenta con el código fuente de ninguna de ellas, en el caso de MaGraDa está diseñada específicamente para trabajar con grafos dirigidos, no dirigidos, ponderados y no ponderados. Y su filosofía permite representar grafos con un máximo de 50 vértices solamente. FCMappers tiene como objetivo crear una plataforma para compartir experiencias e información sobre mapas cognitivos difusos es una red de análisis de la red, por lo que no se aprueba la utilización de algunos de estos dos sistemas.

## **Sistemas nacionales:**

### ***1.2.3 Herramienta de creación de grafos de caminos para la biblioteca “SceneToolkit”***

En el año 2008 el departamento de Entornos Virtuales de la facultad 5 en la UCI, presentó un trabajo de diploma titulado: Herramienta de creación de grafos de caminos para la biblioteca “SceneToolkit”. La herramienta está implementada utilizando como el lenguaje de programación C++, siguiendo los principios de la programación orientada a objetos. Las interfaces gráficas basadas en widgets están diseñadas en la aplicación QT Designer y utiliza la biblioteca SceneToolkit para el tratamiento de entornos virtuales, la que a su vez hace uso de la biblioteca gráfica libre de más bajo nivel OpenGL. Para construir la aplicación visual utiliza la biblioteca de clases QT mientras que el framework STK Editor se encarga de la vinculación QT–STK. Para la representación de las estructuras de datos, hace uso de la STL de C++. El sistema permite además importar y exportar ficheros de mundo 3D y en formato xml respectivamente, gestiona nodos y aristas, convirtiéndolos en grafos dirigidos y no ponderados y aplica algoritmos de planificación de movimientos a grafo de caminos. (8)

Esta herramienta no es adecuada debido a que no se ajusta al problema propuesto, gestiona nodos y aristas, convirtiéndolos en grafos dirigidos y no ponderados y aplica algoritmos de planificación de movimientos a grafo de caminos. Por lo que no se aprueba la utilización de este sistema.

### ***1.2.4 FCM-Decisión***

Permite ajustar un grupo de parámetros relacionados al modelo como son la selección de la función de activación, las medidas de distancia, los vectores de pesos relacionados con los operadores de agregación y las demás opciones relacionadas fundamentalmente con el aprendizaje y el análisis orientado a metas.

# CAPÍTULO 1: FUNDAMENTOS TEÓRICOS

FCM-Decisión permite que los modelos sean almacenados en una base de datos, lo que posibilita su consulta y actualización, funcionando como un repositorio de modelos causales. Esta funcionalidad facilita su empleo en la ingeniería y gestión del conocimiento en las organizaciones. Siendo la herramienta idónea para la utilización en la solución a desarrollar.

El software desarrollado ha tenido aplicación en el desarrollo de sistemas basados en el conocimiento para la resolución de problemas concretos, contribuyendo a simplificar el proceso de ingeniería del conocimiento. También ha sido utilizado como medio de enseñanza para el desarrollo de habilidades prácticas en la construcción de sistemas inteligentes y la ingeniería de software.

Seleccionando esta última FCM-Decisión como la herramienta idónea pues se tiene acceso a sus funcionalidades y a diferencia de las otras se pueden realizar análisis estáticos y dinámicos sin límites en la cantidad de nodos y colabora en la toma de decisiones.

## **1.3 Metodología y herramientas a utilizar en el desarrollo del servicio web**

Las metodologías de desarrollo de software describen los pasos que se deben seguir para la producción de un producto informático, con objetivo de lograr alta calidad en dicho producto. Están formadas por un conjunto de procedimientos, reglas, técnicas y herramientas que guían a los desarrolladores en el proceso de cómo crear un software, a través de patrones. Las herramientas son importantes en el desarrollo de un software, una buena selección de las mismas minimiza los esfuerzos y permite desarrollar las aplicaciones en tiempo. Fue necesario realizar un análisis de las herramientas, tecnologías y lenguajes con el fin de seleccionar aquellas que se adapten a los objetivos que se persiguen.

A continuación se presenta la metodología, las herramientas, tecnologías y lenguajes de programación existentes que fueron seleccionados y analizados, puesto que se ajustan a la solución a desarrollar. Así como las ventajas y desventajas de su aplicación.

### **1.3.1 Metodología XP:**

XP: Extreme Programming por sus siglas en inglés es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Se basa en retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios.

# CAPÍTULO 1: FUNDAMENTOS TEÓRICOS

Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (9)

Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. Kent Beck, el padre de XP, describe la filosofía de XP sin cubrir los detalles técnicos y de implantación de las prácticas. Posteriormente, otras publicaciones de experiencias se han encargado de dicha tarea. A continuación se presentan las características esenciales de XP organizadas en los apartados siguientes: Historias de usuario, Roles, Proceso y Prácticas.

- **Las Historias de Usuario**

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales.

El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento las historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas.

- **Roles XP**

Aunque en otras fuentes de información aparecen algunas variaciones y extensiones de roles XP, los roles de acuerdo con la propuesta original de Beck son:

- El programador, quien escribe las pruebas unitarias y produce el código del sistema, debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo, el cliente, pieza básica en desarrollos XP.
- Define especificaciones, influye sin controlar, confía en el grupo de desarrollo, define pruebas funcionales, encargado de pruebas:
- Apoya al cliente en la preparación/realización de las pruebas funcionales, ejecuta las pruebas funcionales y publica los resultados, entrenador (Coach), experto en XP:
- Responsable del proceso en su conjunto, identifica las desviaciones y reclama atención sobre las mismas, guía al grupo de forma indirecta (sin dañar su seguridad ni confianza), interviene directamente si es necesario, detiene rápidamente el problema, consultor:

# CAPÍTULO 1: FUNDAMENTOS TEÓRICOS

- Apoya al equipo XP en cuestiones puntuales, jefe del proyecto: favorece la relación entre usuarios y desarrolladores, confía en el equipo XP, cubre las necesidades del equipo XP, asegura que alcanza sus objetivos.
- **Procesos y prácticas**

Un proyecto XP tiene éxito cuando el cliente selecciona el valor de negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo. El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

**Fase I:** Exploración Laboratorio de Sistemas de Información. Departamento de Sistemas Informáticos y Computación.

**Fase II:** Planificación de la Entrega.

**Fase III:** Iteraciones.

**Fase IV:** Producción Laboratorio de Sistemas de Información. Departamento de Sistemas Informáticos y Computación.

**Fase V:** Mantenimiento.

**Fase VI:** Muerte del Proyecto.

- **Algunas de las Ventajas de la Metodología XP son:**
  - Programación organizada.
  - Menor tasa de errores.
  - Satisfacción del programador.
  - Solución de errores de programas.
  - Versiones nuevas.
  - Implementa una forma de trabajo que se adapta fácilmente a las circunstancias.

Debido a las características y ventajas se escogió la metodología XP pues es una metodología ágil con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico, la programación extrema es una buena alternativa, si se quiere diseñar un software en corto tiempo. No solo la rapidez es una de sus ventajas, sino también las habilidades sociales, el trabajo en equipo y la disciplina.

## **1.3.2 Lenguaje Unificado de Modelado (UML) 2.0**

UML es uno de los lenguajes de modelado de procesos más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. Ofrece un estándar para

# CAPÍTULO 1: FUNDAMENTOS TEÓRICOS

describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como, procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Se escogió el (UML) en su versión 2.0 por la facilidad que brinda al usarse, por la fácil manipulación y por ser uno de los lenguajes de modelados más utilizados y al mismo tiempo uno de los más conocidos en la actualidad.

### **1.3.3 Herramienta CASE Visual Paradigm 8**

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo de desarrollo de software: análisis y diseño orientado a objetivos, construcción, prueba y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagrama y generar documentación.

De la herramienta CASE Visual Paradigm se seleccionaron las siguientes características:

- Apoyo UML 2.x.
- Diagrama de Clases.
- Diagrama de Casos de Uso.
- Diagrama de Secuencia.
- Diagrama de Comunicación.

Se utilizó la herramienta Visual Paradigm 8 para el modelado por la facilidad que brinda a los desarrolladores y diseñadores que la utilizan, por ser una potente herramienta que soporta el ciclo de vida de desarrollo de un software ayudando a diseñar, modelar, probar y desplegar con una rápida, ágil, fácil construcción y calidad el desarrollo de un software.

### **1.3.4 Entorno de Desarrollo Eclipse 3.2**

Eclipse es un programa informático compuesto por un conjunto de herramientas de programación de código abierto y multiplataforma para desarrollar "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, es utilizada para desarrollar entornos de desarrollo integrados (del inglés IDE), ejemplo de ello es el IDE llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el

# CAPÍTULO 1: FUNDAMENTOS TEÓRICOS

mismo Eclipse). Sin embargo, también se puede usar para otros tipos de aplicaciones cliente, como BitTorrent o Azureus. (10)

Eclipse dispone de un editor de texto con resaltado de sintaxis. La compilación es en tiempo real. Tiene pruebas unitarias con JUnit, control de versiones con CVS, integración con Ant, asistentes (wizards) para creación de proyectos, clases, tests, refactorización. Asimismo, a través de "plug-ins" libremente disponibles es posible añadir control de versiones con Subversion.4 integración con Hibernate.

Por tanto se decide utilizar como IDE de desarrollo Eclipse en su versión 3.6 por las ventajas que brinda para el desarrollo de servicios web con distintos lenguajes como PHP, HTML5, JAVASCRIPT y Java. Además por su capacidad de ser multiplataforma y por contar con un excelente completamiento de código que ayuda a los programadores en el desarrollo de sus aplicaciones y permite establecer conexiones con las bases de datos de PostgreSQL, Oracle, Mysql, entre otras.

## **1.3.5 Lenguaje de programación Java**

El lenguaje para la programación avanzada en Java, fue desarrollado por la compañía Sun Microsystems, con la idea original de usarlo para la creación de páginas web. Esta programación avanzada en Java tiene muchas similitudes con el lenguaje C y C++, por lo que si se tiene conocimiento de este lenguaje, el aprendizaje de la programación avanzada en Java será de fácil comprensión por un programador que haya realizado programas en estos lenguajes.

Con la programación avanzada en Java, se pueden realizar distintos aplicativos, como son applets, que se ejecutan dentro de un navegador al ser cargada una página HTML en un servidor web. Otra de las utilidades de la programación avanzada en Java es el desarrollo de aplicaciones, 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. (11)

La programación avanzada en Java, permite el desarrollo de aplicaciones bajo el esquema de Cliente Servidor, distribuidas, y permite conectar dos o más computadoras ejecutando tareas simultáneamente, y de esta forma logra distribuir el trabajo a realizar. De acuerdo a las características antes enunciadas, se selecciona el lenguaje de programación Java para desarrollar el servicio web.

# CAPÍTULO 1: FUNDAMENTOS TEÓRICOS

## **1.3.6 Middleware Axis2**

Es un software que asiste a una aplicación para interactuar o comunicarse con otras aplicaciones, software, redes, hardware y/o sistemas operativos. Funciona como una capa de abstracción de software distribuida, que se sitúa entre las capas de aplicaciones y las capas inferiores (sistemas operativos y red). (12)

El middleware se abstrae de la complejidad y heterogeneidad de las redes de comunicaciones subyacentes, así como de los sistemas operativos y lenguajes de programación, proporcionando una API para la fácil programación y manejo de aplicaciones distribuidas. Dependiendo del problema a resolver y de las funciones necesarias, serán útiles diferentes tipos de servicios de middleware. Por lo general el middleware del lado cliente está implementado por el sistema operativo, el cual posee las bibliotecas que ejecutan todas las funcionalidades para la comunicación a través de la red. (12)

Se utilizó Axis2 pues simplifica el trabajo de los programadores en la compleja tarea de generar las conexiones que son necesarias en los sistemas distribuidos. De esta forma se provee una solución que mejora la calidad de servicio, seguridad, envío de mensajes, directorio de servicio, entre otras.

## **1.3.7 ApacheTomcat 7**

Apache, también conocido como simplemente Tomcat 7, es un servidor web multiplataforma que funciona como contenedor de servlets y que se desarrolla bajo el proyecto denominado Jakarta perteneciente a la Apache Software Foundation bajo la licencia Apache 2.0 y que implementa las especificaciones de los servlets y de JavaServer Pages o JSP de Sun Microsystem. Dicho servidor es mantenido y desarrollado por miembros de la fundación y voluntarios independientes, los cuales tienen libre acceso al código fuente bajo los términos establecidos por la Apache Software Foundation. (13)

Se utilizó ApacheTomcat 7 debido que en él se pueden montar aplicaciones de misión crítica a través de la Web. También permite montar servicios web mediante Axis.

## **1.3.8 MySQL Workbench**

MySQL Workbench es una herramienta que permite modelar diagramas de entidad-relación para bases de datos MySQL. La aplicación elabora una representación visual de las tablas, vistas, procedimientos almacenados y claves extranjeras de la base de datos. Además, es capaz de sincronizar el modelo en desarrollo con la base de datos real.



# CAPÍTULO 1: FUNDAMENTOS TEÓRICOS

MySQL Workbench también puede generar el script necesario para crear la base de datos que se ha dibujado en el esquema; es compatible con los modelos de base de datos de Designer 4 y soporta las novedades incorporadas en MySQL 5. (14) (15)

## **Estas son algunas de las características principales de MySQL Workbench:**

- Edición de diagramas basada en Cairo, con posibilidad de realizar una salida en los formatos como OpenGL, Win32, X11, Quartz, PostScript, PDF.
- Proporciona una representación visual de las tablas, vistas, procedimientos y funciones almacenadas y claves foráneas.
- Permite acceso a bases de datos e ingeniería inversa de las mismas para crear los SQL de creación.
- Ofrece sincronización con la base de datos y el modelo.
- Permite generar los scripts SQL a partir del modelo creado.
- Ofrece una arquitectura extensible.
- Tiene soporte para exportar los datos como script SQL CREATE.
- Permite importar modelos de DBDesigner4.
- Ofrece soporte completo a las características de MySQL 5.

## **Las plataformas principales dónde MYSQL WORKBEACH se encuentra funcionando son:**

- Mac OSX.
- Windows XP, Windows Vista y Windows 7.
- Ubuntu.

Se utilizó MySQL Workbench después de haber realizado un estudio, pues puede se puede utilizar para diseñar el esquema de una base de datos nueva, documentar una ya existente o realizar una migración compleja.

### **1.3.9 Hibernate**

Es un conjunto de proyectos relacionados que permiten a los desarrolladores utilizar estilos de modelos de dominio en sus aplicaciones de manera que se extienden mucho más allá del mapeo objeto-relacional. (16) Se decide utilizar Hibernate pues permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. Genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución.

# CAPÍTULO 1: FUNDAMENTOS TEÓRICOS

## Conclusiones parciales

En la investigación realizada se consultaron diversas fuentes bibliográficas que fundamentan teórica y metodológicamente varias herramientas. Según lo antes expuesto para dar respuesta al problema planteado, se realizará un servicio web sobre el middleware Axis2 el cual se desplegará sobre un servidor de aplicaciones web Apache Tomcat. Para el desarrollo del cliente que consumirá el servicio web se utilizará la metodología XP por ser una ágil, utilizada en proyectos cortos con equipos pequeños en cuanto a roles. Como herramienta de modelado se hará uso del Visual Paradigm para UML 8 que es una herramienta que cubre todo el ciclo de desarrollo del software y además es de licencia libre. El lenguaje de programación a utilizar será el Java por ser un lenguaje multiplataforma, orientado a objetos y de licencia libre. Como IDE de desarrollo será utilizado el Eclipse Helios 3.2, el cual permite trabajar con el lenguaje de programación propuesto, es fácil de integrar con Axis2 y es un software de licencia libre. Para trabajar con la base de datos se utilizará la herramienta MySQL Workbench, se Puede utilizar para diseñar el esquema de una base de datos nueva, documentar una ya existente o realizar una migración compleja. Y el Hibernate 3.3.1 como herramienta de mapeo objeto-relacional, por ser de software libre y distribuido bajo los términos de la licencia GNU/LGPL.

### CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA

En el presente capítulo, se definen y detallan los requisitos a implementar en el proceso de desarrollo del software. Se organiza la estructura y el comportamiento del sistema para guiar de una manera más sencilla a los clientes y desarrolladores a su comprensión. Se muestra un modelo de dominio con el fin de proveer un entendimiento más detallado de los principales conceptos del negocio, así como una descripción de las historias de usuario. Se identificaron los actores, casos de uso y las relaciones existentes entre estos, proporcionando como resultado la realización del diagrama de casos de uso del sistema.

#### 2.1 Modelo de dominio

El modelo de dominio se utiliza como fuente de inspiración para el diseño de los objetos de software. Este muestra a los usuarios, clases conceptuales significativas en un dominio de problema; es el artefacto más importante que se crea durante el análisis orientado a objetos.

Un modelo del dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. También se les denomina modelos conceptuales (término utilizado en la primera edición del libro de Larman). Su objetivo es enfocarse en una parte del negocio relacionándola con el ambiente del proyecto. A pesar que el modelo de dominio no se corresponde con la metodología XP se decide diseñar para un mayor entendimiento de la lógica del negocio. A continuación se presenta el Modelo de Dominio para el diseño del sistema (ver figura 1).

## CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA

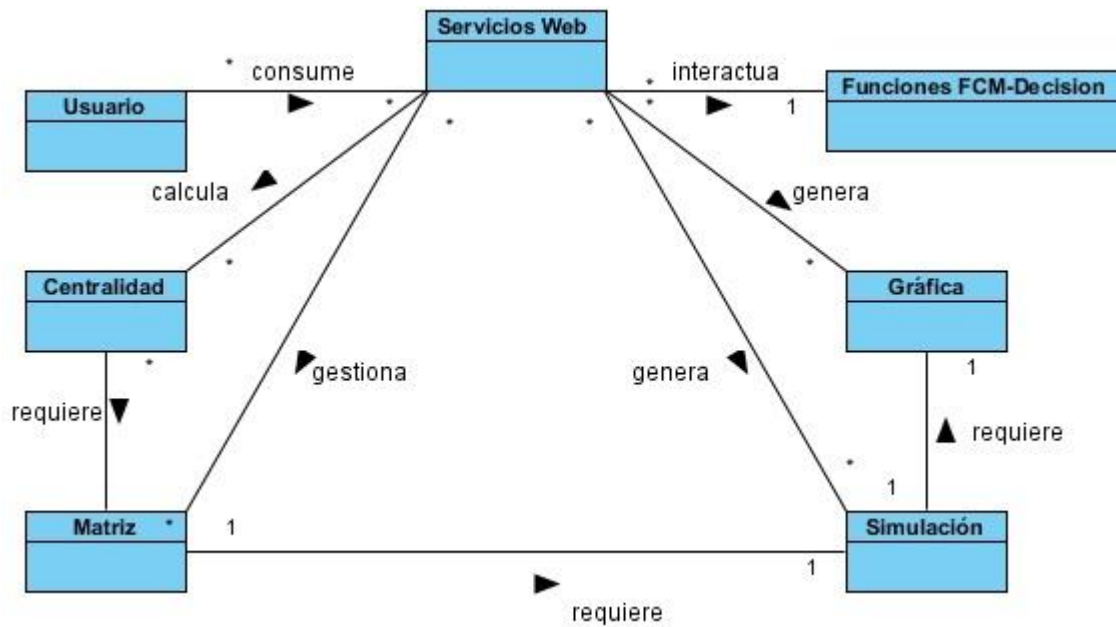


Fig. 1 Modelo de Dominio

Tabla. 1 Descripción de los conceptos del diagrama de dominio

Concepto	Descripción
Usuario	Persona que interactúa con el Servicio Web.
Servicio Web	Servicio Web para centralizar, estandarizar y ofrecer información detallada de las funcionalidades que brinda la aplicación FCM-Decision.
Funciones FCM-Decision	Funciones de la aplicación de escritorio FCM-Decision.
Matriz	Servicios Web relacionados con la matriz (agregar matriz, listar matriz, Eliminar matriz)
Centralidad	Resultados obtenidos del cálculo de los valores de entrada y salida de cada uno de los nodos del mapa cognitivo difuso o grafo.
Simulación	Simulación se realiza para tomar decisiones luego del proceso teniendo en cuenta la función que se utiliza, esto da como resultado los nodos más importantes y como

## CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA

	puedes variar la relación entre los mismos para obtener el resultado deseado.
Gráfica	Muestra un reporte gráfico de las simulaciones obtenidas.

### 2.2 Descripción del sistema propuesto

Se propone implementar la versión 1.0 Capa de Servicios Web para la aplicación FCM-Decision, basándose en las funciones de la aplicación de escritorio FCM-Decision. Esta capa de servicio web implementara las principales funcionalidades de la aplicación FCM-Decision como son: cálculo de la centralidad de factores, diseño de mapas cognitivos difusos por medio de matriz de adyacencia, simulación de los mapas cognitivos difusos y la traficación de los resultados obtenidos luego de la simulación, entre otras. Estas funcionalidades permitirán a los especialistas o usuarios de los servicios tomar decisiones para elegir entre varias opciones cual es la más conveniente en un determinado momento.

#### 2.2.1 Las Historias de Usuario

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales.

El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento las historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas.

Se utilizaron las historias de usuarios pues siguen los principios básicos de los requerimientos ágiles:

- Potencian la participación del equipo en la toma de decisiones.
- Se crean y evolucionan a medida que el proyecto avanza.
- Son peticiones concretas y pequeñas.
- Contienen información imprescindible.
- Apoyan la cooperación, colaboración y conversación entre los miembros del equipo, lo que es fundamental.

En las tablas siguientes se muestran las historias de usuario definidas para el desarrollo de la propuesta de solución de la investigación.

## CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA

Tabla. 2 Historia de usuario Agregar matrices

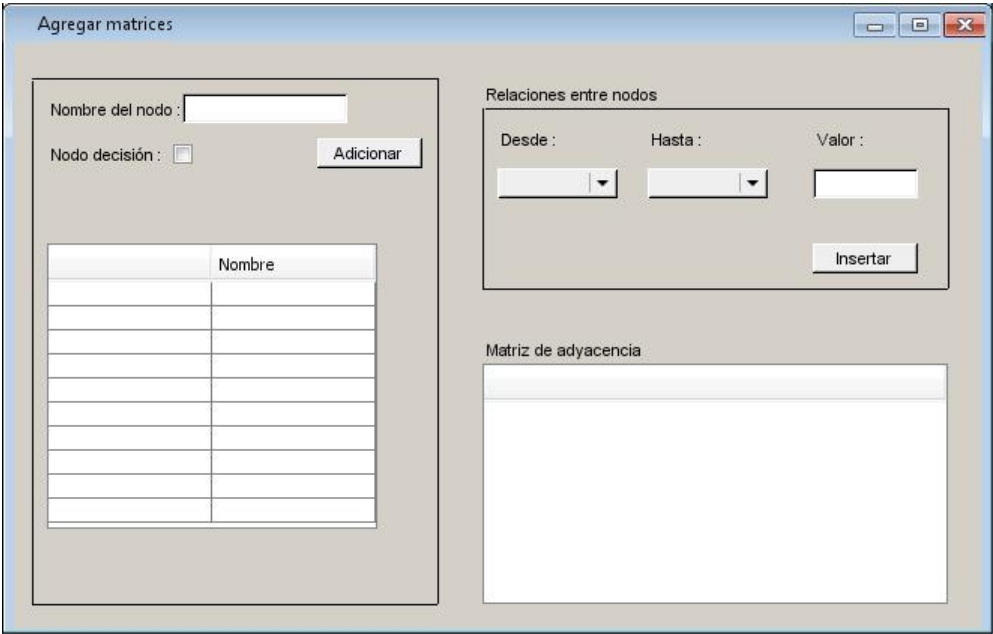
Historia de Usuario	
<b>Número:</b> 1	<b>Nombre:</b> Agregar matrices
<b>Cantidad de modificaciones:</b> 2	
<b>Usuario:</b> Erichel Almanza Castellanos	<b>Iteración asignada:</b> 1
<b>Prioridad en negocio:</b> Muy Alta	<b>Puntos estimados:</b> 1
<b>Riesgo en desarrollo:</b> Alto	<b>Puntos reales:</b> 1
<p><b>Descripción:</b> Permite al usuario agregar una matriz, introduciendo el nombre de cada nodo y seleccionando si es aun nodo decisión o no, Al final se mostrarán en una tabla todos los nodos adicionados. Permite adicionar las relaciones entre nodos y el valor de las aristas, mostrando al final la matriz de adyacencia.</p>	
<b>Observaciones:</b>	
<p><b>Prototipo de interfaz:</b></p> 	

Tabla. 3 Historia de usuario Listar matrices.

Historia de Usuario	
<b>Número:</b> 2	<b>Nombre:</b> Listar matrices

## CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA

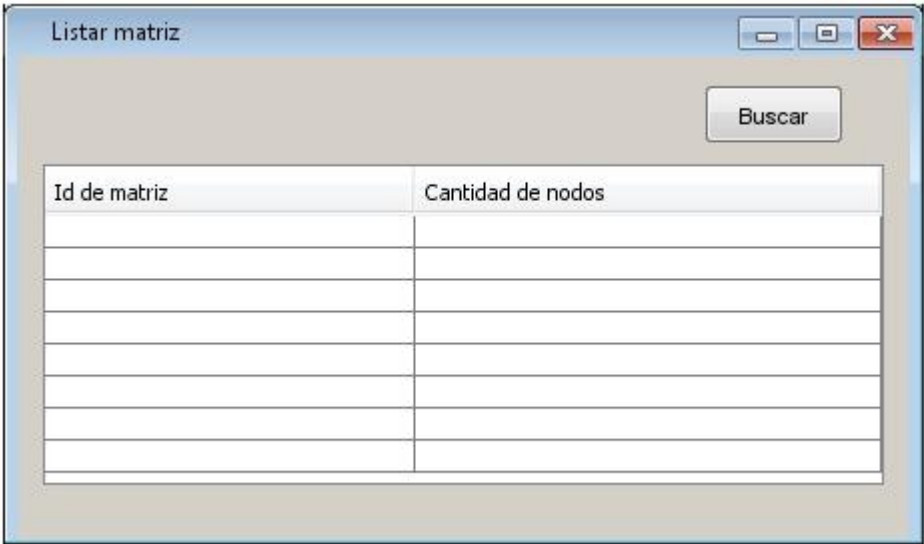
<b>Cantidad de modificaciones:</b> 1	
<b>Usuario:</b> Erichel Almanza Castellanos	<b>Iteración asignada:</b> 2
<b>Prioridad en negocio:</b> Alta	<b>Puntos estimados:</b> 1
<b>Riesgo en desarrollo:</b> Alto	<b>Puntos reales:</b> 1
<b>Descripción:</b> Permite al usuario listar todas las matrices almacenadas en la base de datos, mostrando en una tabla su id y la cantidad de nodos.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	
	

Tabla. 4 Historia de usuario Obtener matriz a través de su ID

Historia de Usuario	
<b>Número:</b> 3	<b>Nombre:</b> Obtener matriz a través de su ID
<b>Cantidad de modificaciones:</b> 1	
<b>Usuario:</b> Erichel Almanza Castellanos	<b>Iteración asignada:</b> 2
<b>Prioridad en negocio:</b> Alta	<b>Puntos estimados:</b> 1
<b>Riesgo en desarrollo:</b> Alto	<b>Puntos reales:</b> 1

## CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA

**Descripción:** Permite al usuario obtener una matriz según un id de las matrices almacenadas en la base de datos, mostrándola debajo mediante el botón Buscar.

**Observaciones:**

**Prototipo de interfaz:**

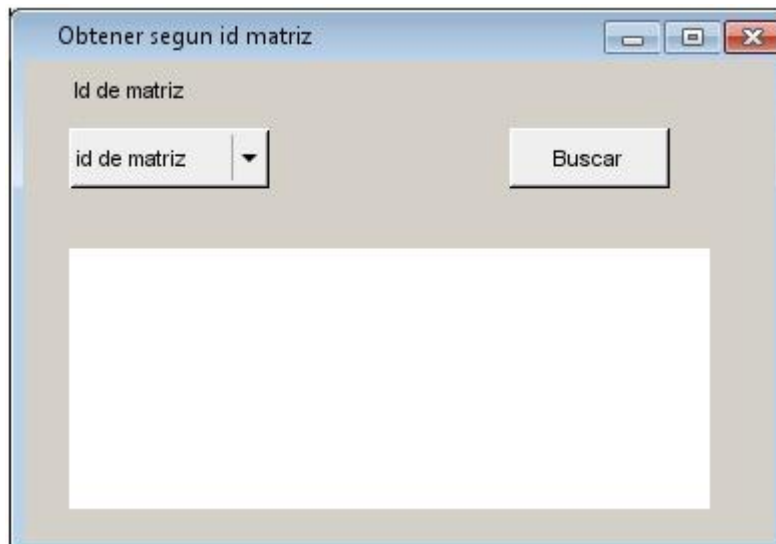


Tabla. 5 Historia de usuario Centralidad de factores por id Matriz.

Historia de Usuario	
<b>Número:</b> 4	<b>Nombre:</b> Centralidad de factores por id matriz
<b>Cantidad de modificaciones:</b> 1	
<b>Usuario:</b> Erichel Almanza Castellanos	<b>Iteración asignada:</b> 1
<b>Prioridad en negocio:</b> Muy alto	<b>Puntos estimados:</b> 1
<b>Riesgo en desarrollo:</b> Alto	<b>Puntos reales:</b> 1
<b>Descripción:</b> Permite al usuario obtener la centralidad de factores de una matriz según su id, la suma de las entradas y las salidas, mostrándola debajo en una tabla compuesta por factores, outdegree, indegree, sumatoria y centralidad mediante el botón aceptar.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	



## CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA

Centralidad de factores por id de matriz

Id de matriz

id\_de\_matriz ▼      Aceptar

Factores	Outdegree	Indegree	Sumatoria	Centralidad

Tabla. 6 Historia de usuario Centralidad de factores por matriz.

Historia de Usuario	
<b>Número:</b> 5	<b>Nombre:</b> Centralidad de factores por matriz
<b>Cantidad de modificaciones:</b> 1	
<b>Usuario:</b> Erichel Almanza Castellanos	<b>Iteración asignada:</b> 1
<b>Prioridad en negocio:</b> Muy alto	<b>Puntos estimados:</b> 1
<b>Riesgo en desarrollo:</b> Alto	<b>Puntos reales:</b> 1
<b>Descripción:</b> Permite al usuario a través del id de una matriz o a partir de una nueva matriz agregada obtener la centralidad de factores, la suma de las entradas y salidas. Se mostrará en una tabla los factores, outdegree, indegree, sumatoria y centralidad.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

## CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA

Tabla. 7 Historia de usuario Realizar simulación por ID matriz.

Historia de Usuario	
<b>Número:</b> 6	<b>Nombre:</b> Realizar simulación por ID matriz
<b>Cantidad de modificaciones:</b> 2	
<b>Usuario:</b> Aylin Santana Muiño	<b>Iteración asignada:</b> 1
<b>Prioridad en negocio:</b> Muy alto	<b>Puntos estimados:</b> 1
<b>Riesgo en desarrollo:</b> Alto	<b>Puntos reales:</b> 1
<b>Descripción:</b> Permite a los usuarios realizar la simulación de la matriz introduciendo los siguientes valores: IDmatriz, vector, landa, función.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

## CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA

Tabla. 8 Historia de usuario Realizar simulación por matriz

Historia de Usuario	
<b>Número:</b> 7	<b>Nombre:</b> Realizar simulación por matriz
<b>Cantidad de modificaciones:</b> 2	
<b>Usuario:</b> Aylin Santana Muiño	<b>Iteración asignada:</b> 1
<b>Prioridad en negocio:</b> Muy alto	<b>Puntos estimados:</b> 1
<b>Riesgo en desarrollo:</b> Alto	<b>Puntos reales:</b> 1
<b>Descripción:</b> Permite a los usuarios realizar la simulación de la matriz Introduciendo los siguientes valores: matriz, vector, landa, función la matriz se adicionaría primero mediante el botón agregar para después poder seleccionarla mediante su id.	
<b>Observaciones:</b>	

## CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA

### Prototipo de interfaz:

Realizar simulación por matriz

Adicionar matriz      Seleccionar matriz  
      id de matriz ▾

Valor de landan :       Función de activación  
 vector            ..Funciones... ▾

Tabla. 9 Historia de usuario obtener resultado de la simulación.

Historia de Usuario	
<b>Número:</b> 8	<b>Nombre:</b> Obtener resultado de la simulación
<b>Cantidad de modificaciones:</b> 1	
<b>Usuario:</b> Aylin Santana Muiño	<b>Iteración asignada:</b> 1
<b>Prioridad en negocio:</b> Muy alto	<b>Puntos estimados:</b> 1
<b>Riesgo en desarrollo:</b> Alto	<b>Puntos reales:</b> 1
<b>Descripción:</b> Permite a los usuarios conocer el resultado de la simulación mediante el id propio y un vector a través del botón simular.	
<b>Observaciones:</b>	

## CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA

Prototipo de interfaz:



Tabla. 10 Historia de usuario obtener gráfica.

Historia de Usuario	
<b>Número:</b> 9	<b>Nombre:</b> Obtener gráfica
<b>Cantidad de modificaciones:</b> 3	
<b>Usuario:</b> Aylin Santana Muiño	<b>Iteración asignada:</b> 1
<b>Prioridad en negocio:</b> Muy Alto	<b>Puntos estimados:</b> 1
<b>Riesgo en desarrollo:</b> Alto	<b>Puntos reales:</b> 1
<b>Descripción:</b> Permite a los usuarios obtener las gráficas correspondientes a cada simulación mediante el id propio a través del botón simular.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

## CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA



Tabla. 11 Historia de usuario Eliminar matriz.

Historia de Usuario	
<b>Número:</b> 10	<b>Nombre:</b> Eliminar matriz
<b>Cantidad de modificaciones:</b> 1	
<b>Usuario:</b> Aylin Santana Muiño	<b>Iteración asignada:</b> 2
<b>Prioridad en negocio:</b> Alto	<b>Puntos estimados:</b> 1
<b>Riesgo en desarrollo:</b> Alto	<b>Puntos reales:</b> 1
<b>Descripción:</b> Permite a los usuarios eliminar una matriz existente mediante su id a través del botón eliminar.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

## CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA

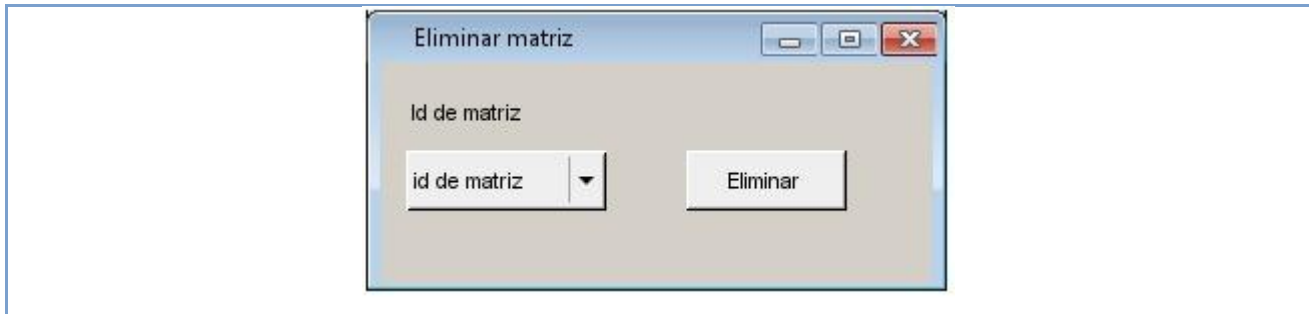


Tabla. 12 Lista de reserva del producto.

Ítem	Descripción	Estimación	Estimado por
<b>Requisitos funcionales</b>			
<b>Prioridad: MUY ALTA</b>			
1	Agregar matrices	1	Analista
2	Realizar centralidad de factores por id matriz	1	Analista
3	Realizar centralidad de factores por matriz	1	Analista
4	Realizar simulación por ID matriz	1	Analista
5	Realizar simulación por matriz	1	Analista
6	Obtener resultado de la simulación	1	Analista
7	Obtener gráfica	1	Analista
<b>Prioridad: ALTA</b>			
8	Listar matrices	1	Analista
9	Obtener matriz a partir de su ID.	1	Analista
10	Eliminar matriz	1	Analista
<b>Requisitos no funcionales</b>			
<b>Software</b>			
11	Sistema Operativo: GNU/Linux preferentemente Ubuntu GNU/Linux 12.4 para montar el servidor y desplegar la capa de servicios.		

## CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA

12	Sistema Operativo: GNU/Linux o Windows en su versión 7 o 8 para el cliente Ubuntu GNU/Linux 12.4 para el cliente		
<b>Hardware</b>			
12	El ordenador debe tener como mínimo: Microprocesador con una velocidad de 500 MHz 512 MB de RAM		
<b>Apariencia o interfaz externa</b>			
13	El sistema debe contar con una interfaz amigable que permita al especialista interactuar de forma cómoda, le agilice y facilite el trabajo con el sistema.		
<b>Portabilidad</b>			
14	El cliente podrá ser ejecutado en cualquier sistema operativo, por su característica de ser multiplataforma, permitiendo una fácil migración haciendo uso de estándares y tecnologías de código abierto.		

### 2.2.2 Plan de iteraciones

El Plan de Iteración recoge la planificación detallada de un período corto de tiempo dentro del proyecto. Expresa de una forma apropiada quién va a realizar una tarea o actividad de acuerdo a qué criterio de evaluación. Éste está compuesto por un conjunto secuencial de tareas y actividades, cada tarea o actividad tiene recursos asignados y puede depender a su vez de otras tareas. Es un plan detallado, que se hace una vez por iteración y el contenido de cada una puede variar, dependiendo de la posición del ciclo de vida y de la naturaleza del proyecto. Se deben identificar las actividades, los riesgos involucrados, los artefactos a actualizar o crear, las necesidades de adquisición de bienes o servicios y los hitos esperados durante la iteración.



## CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA

La implementación del sistema propuesto tendrá una duración de 9 semanas, tiempo en que se llevará a cabo la realización de las iteraciones. La tabla que se muestra a continuación presenta el plan propuesto.

Tabla. 13 Plan de iteraciones.

Iteración	Descripción de la iteración	Orden de la HU a implementar	Duración total
1	En esta iteración se implementarán las historias de usuario que tengan la prioridad en el negocio MUY ALTA.	1, 4, 5, 6, 7,8,9	7 semanas
2	En esta iteración se implementarán las historias de usuario que tengan la prioridad en el negocio ALTA.	2,3,10	3 semanas

### 2.3 Modelo de diseño

Para el diseño de aplicaciones informáticas, la metodología XP no requiere la presentación del sistema mediante diagramas de clases utilizando notación UML. En su lugar se usan otras técnicas como las tarjetas CRC (Class-Responsibility-Collaboration) por sus siglas en inglés.

#### 2.3.1 Tarjetas CRC

La utilización de tarjetas CRC es una técnica de diseño orientado a objetos. El objetivo de la misma es hacer, mediante tarjetas, un inventario de las clases que vamos a necesitar para implementar el sistema y la forma en que van a interactuar, de esta forma se pretende facilitar el análisis y discusión de las mismas por parte de varios actores del equipo de proyecto con el objetivo de que el diseño sea lo más simple posible verificando las especificaciones del sistema. (17)

Para la implementación de las 10 historias de usuario, definidas para dar cumplimiento a los requisitos del cliente, se identificaron 10 tarjetas CRC:

## CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA

Tabla. 14 Tarjeta CRC Agregar matrices.

Tarjeta CRC	
<b>Clase:</b> ImpModelo	
Responsabilidades	Colaboraciones
public void agregarModelo() public String agregarNodo() public String agregarConector()	Modelo, Nodo, Conector, Componente

Tabla. 15 Tarjeta CRC Listar matrices.

Tarjeta CRC	
<b>Clase:</b> ImpModelo	
Responsabilidades	Colaboraciones
public void listar()	Modelo

Tabla. 16 Tarjeta CRC Obtener según id matriz.

Tarjeta CRC	
<b>Clase:</b> ImpModelo	
Responsabilidades	Colaboraciones
public void obtener()	Modelo

Tabla. 17 Tarjeta CRC Centralidad de Factores por ID matriz

Tarjeta CRC	
<b>Clase:</b> ImpModelo	
Responsabilidades	Colaboraciones
public String centralidadFactoresobtener()	Modelo, Nodo , Conector

## CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA

Tabla. 18 Tarjeta CRC Centralidad de Factores por matriz.

Tarjeta CRC	
<b>Clase:</b> ImpModelo	
Responsabilidades	Colaboraciones
public String centralidadFactoresobtener()	Modelo, Nodo , Conector

Tabla. 19 Tarjeta CRC obtener resultado de centralidad según ID.

Tarjeta CRC	
<b>Clase:</b> ImpModelo	
Responsabilidades	Colaboraciones
public String devolverCentralidadFactoresobtener()	Modelo, Nodo, Conector

Tabla. 20 Tarjeta CRC Realizar simulación por ID.

Tarjeta CRC	
<b>Clase:</b> ImpModelo	
Responsabilidades	Colaboraciones
Public string simular()	Modelo, VectorResultante, VectorEntrada, GrafoUtil, HibernateUtil

Tabla. 21 Tarjeta CRC Realizar simulación por matriz.

Tarjeta CRC	
<b>Clase:</b> ImpModelo	
Responsabilidades	Colaboraciones
Public string simular()	Modelo, VectorResultante, VectorEntrada, GrafoUtil, HibernateUtil

## CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA

Tabla. 22 Tarjeta CRC Obtener resultado de simulación.

Tarjeta CRC	
Clase: ImpModelo	
Responsabilidades	Colaboraciones
Public string devolversimular()	Modelo, VectorResultante, VectorEntrada, GrafoUtil, HibernateUtil

Tabla. 23 Tarjeta CRC Obtener gráfica.

Tarjeta CRC	
Clase: ImpModelo	
Responsabilidades	Colaboraciones
Public void grafica()	

Tabla. 24 Tarjeta CRC Eliminar matriz.

Tarjeta CRC	
Clase: ImpModelo	
Responsabilidades	Colaboraciones
Public String Eliminar()	Modelo, Conector, Nodo, HibernateUtil

### 2.4 Diagramas de Clases

Los diagramas de clases son diagramas de estructura estática que muestran las clases del sistema y sus interrelaciones incluyendo herencia, agregación, asociación. Los diagramas de clases son el pilar básico del modelado con UML, siendo utilizados tanto para mostrar lo que el sistema puede hacer, como para mostrar cómo puede ser construido. El diagrama de clases de más alto nivel, será lógicamente un dibujo de los paquetes que componen el sistema. Las clases se documentan con una descripción de lo que hacen, sus métodos y sus atributos. Las relaciones entre clases se documentan con una descripción de su propósito, sus objetos que intervienen en la relación. (18)

A pesar que la metodología XP no genera los diagramas de clases, se diseñó un diagrama de clases para mostrar de manera más simplificada el uso de los patrones de diseño y arquitectura. A continuación se

## CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA

presenta el diagrama de clases correspondiente al desarrollo de la aplicación, el cual muestra las relaciones entre las clases que conforman el funcionamiento interno de la misma. (Ver figura 2)

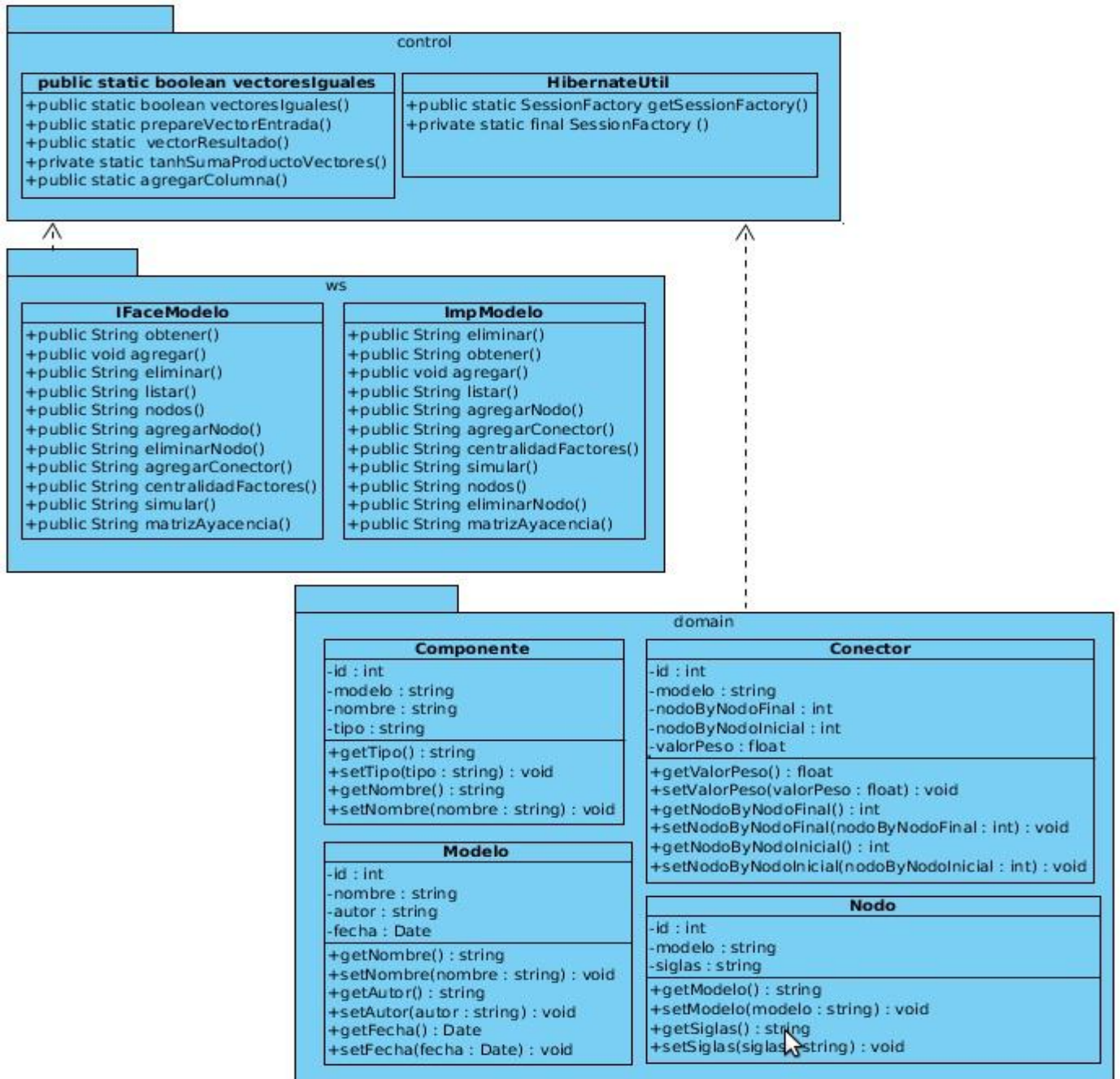


Fig. 2 Diagrama de Clases

## CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA

En el diagrama anterior puede observarse las clases que definen el funcionamiento interno de la aplicación que forma parte de esta solución. Este diagrama está dividido en tres partes fundamentales. La primera de ellas es la que corresponde a las clases modelo que las contiene el paquete domain encargadas de representar y manejar los datos del programa, en la que se identificaron un total de 4 clases. La segunda está formada por las clases controladora que es la que recibe las órdenes del usuario y actúa sobre los datos del modelo estas clases la contiene el paquete control y una tercera parte en la que se identificaron 2 clases vistas que genera una representación visual del modelo estas clases la contiene el paquete ws.

### 2.5 Arquitectura base principal de la aplicación

#### 2.5.1 Patrón de arquitectura:

Los patrones de arquitectura están orientados a representar los diferentes elementos que componen una solución de software y las relaciones entre ellos. A diferencia de los patrones de diseño de software que están orientados a objetos y clases patrones creacionales, estructurales, de comportamiento, de interacción, los patrones de arquitectura están a un mayor nivel de abstracción. (19)

Los patrones de arquitectura forman parte de la llamada Arquitectura de Software (arquitectura lógica de un sistema), que de forma resumida comprende:

- El diseño de más alto nivel de la estructura del sistema.
- Los patrones y abstracciones necesarios para guiar la construcción del software de un sistema.
- Los fundamentos para que analistas, diseñadores, programadores etc. trabajen en una línea común que permita cubrir restricciones y alcanzar los objetivos del sistema.
- Los objetivos del sistema, no solamente funcionales, sino de mantenimiento, auditoría, flexibilidad e interacción con otros sistemas.
- Las restricciones que limitan la construcción del sistema acorde a las tecnologías disponibles para su implementación.

Uno de los patrones más utilizados por empresas desarrolladoras de software para crear productos informáticos, es el patrón MVC (Modelo-Vista-Controlador) el cual se decide utilizar para el desarrollo de la presente solución por la facilidad de organizar y estructurar las clases con este patrón, dividiéndose en las siguientes partes:

## CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA



Fig. 3 Diagrama de la arquitectura Modelo Vista Controlador

**Modelo:** es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones, define las reglas del negocio y recoge todas las funcionalidades del sistema en nuestra solución, el objeto modelo es el paquete domain que se encuentra en la figura 2.

**Vista:** genera una representación visual del modelo y muestra los datos al usuario, el objeto vista es el paquete ws que se encuentra en la figura 2.

**Controlador:** es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del modelo o por alteraciones de la vista. Ejecuta la lógica del modelo y las vistas, el objeto controlador es el paquete control que se encuentra en la figura 2.

### 2.5.2 Patrones de Diseño:

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que se debe reutilizar, lo que significa que es aplicable a diferentes problemas de diseño en circunstancias parecidas. Estableciendo patrones de diseño, se hace más fácil construir soluciones específicas que tienen determinadas estructuras o piezas en común, reutilizando al mismo tiempo experiencias adquiridas. (20)

## CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA

Los patrones de diseños pretenden:

- Proporcionar catálogos de elementos que se puedan reutilizar en el diseño de sistemas de software.
- Evitar la reiteración de la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.
- Formalizar un vocabulario común entre diseñadores.
- Estandarizar el modo en que se realiza el diseño.
- Facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento ya existente.

### Patrones Grasp

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones. GRASP es un acrónimo que significa General Responsibility Assignment Software Patterns. El nombre se eligió para indicar la importancia de captar estos principios, si se quiere diseñar eficazmente el software orientado a objetos.

Los patrones GRASP que se aplican en el desarrollo de la presente solución son:

### Patrón Experto

Consiste en asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.

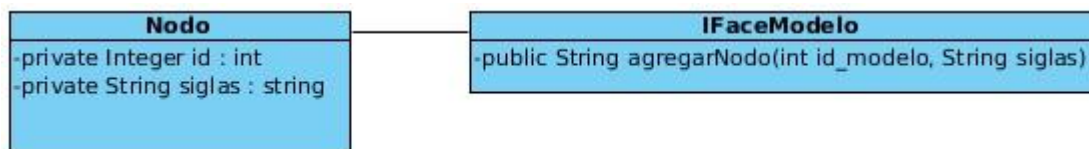


Fig. 4 Patrón Experto

En esta imagen se puede evidenciar el patrón experto, donde la clase nodo es la clase contenedora de la información y en ifModelo se utilizara esta información correspondiente a la clase nodo.

### Patrón Creador

Consiste en asignar a la clase B la responsabilidad de crear una instancia de clase A.

B agrega los objetos A.

B contiene los objetos A.



## CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA

B registra las instancias de los objetos A.

B utiliza específicamente los objetos A.

B tiene los datos de inicialización que serán transmitidos a cuando este objeto sea creado (así que B es un experto respecto a la creación de A).

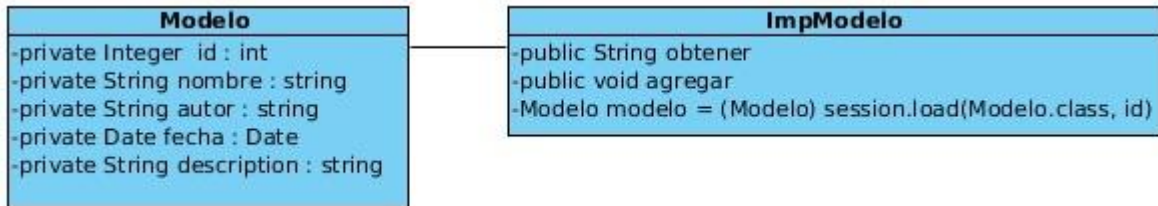


Fig. 5 Patrón Creador

En esta imagen se puede evidenciar el patrón creador, donde la clase ImpModelo como clase B es la responsable de crear una instancia de la clase Modelo que será en este caso la clase A.

### Alta Cohesión

Asigna responsabilidades de tal forma que la información que almacena una clase sea coherente y esté relacionada con la clase. Este patrón se pone de manifiesto durante todo el desarrollo del servicio web, pues cada clase se encarga solamente de hacer lo que le corresponde, evitando la dependencia entre las clases.

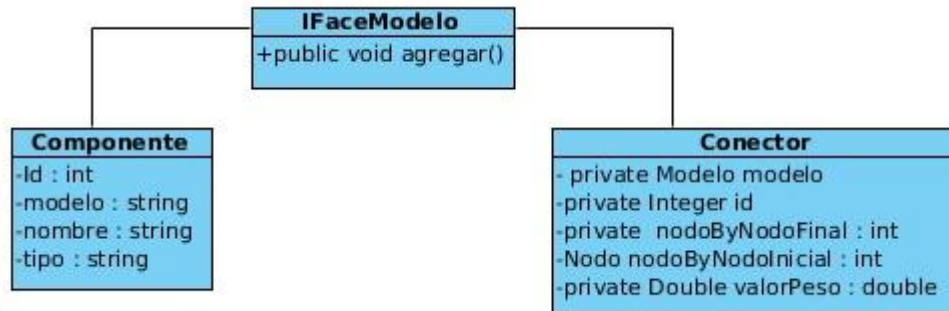


Fig. 6 Alta cohesión

En esta figura se evidencia el patrón de diseño Alta cohesión en la implementación de la solución, puesto que la información que almacena las clases Componente y Conector es coherente y está a la medida de lo posible relacionada con la clase.

### Bajo acoplamiento

## CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL SISTEMA

Este patrón está diseñado con el objetivo de tener las clases lo menos ligadas entre sí, de forma tal que debe haber pocas dependencias entre las clases. En caso de producirse una modificación en alguna de ellas, se tenga la mínima recuperación posible en las resto de las clases, para potenciar, y disminuir la dependencia entre las clases. Este patrón está presente siempre que se asegure la alta cohesión.

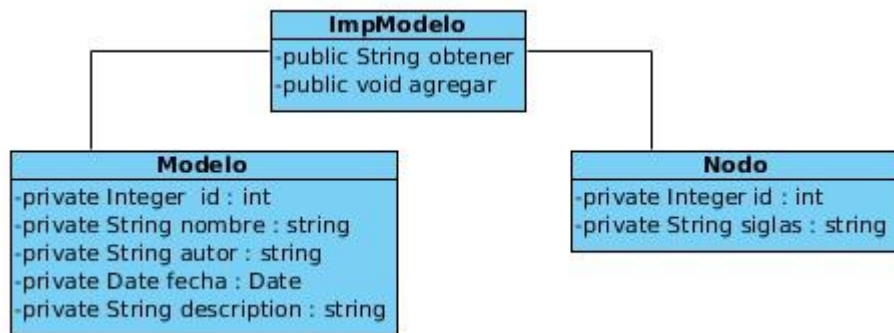


Fig. 7 Patrón bajo acoplamiento

En esta figura se evidencia el patrón de diseño Bajo acoplamiento en la implementación de la solución, puesto que las clases **Modelo** y **Nodo** están lo menos ligadas entre sí que se pueda, de tal forma que, en caso de producirse una modificación en alguna de ellas, tenga la mínima repercusión posible en el resto de las clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases.

### Conclusiones parciales

En este capítulo se crearon los artefactos generados con el empleo de la metodología XP, que contribuirán al éxito del desarrollo de la nueva versión del Servicio web para la simulación y análisis de Mapas Cognitivos Difusos. Se confeccionó un modelo de dominio para entender el funcionamiento del negocio. Fueron identificadas 10 historias de usuario desglosadas en 10 requisitos funcionales agrupados en la lista de reserva del producto. Se confeccionaron 10 tarjetas CRC para la implementación de las historias de usuario definidas. Se elaboró además un plan de iteraciones, con el que se concluye que la duración de la implementación de la extensión tendrá una duración de 10 semanas.

### CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

En el desarrollo de este capítulo se detalla el proceso de implementación de la capa de Servicios Web para la aplicación FCM-Decision. La implementación y pruebas de un producto informático inician a partir de los resultados obtenidos en el diseño implementando. Como resultado de estas fases se obtiene un producto práctico para entregar a los clientes, el que debe responder al cumplimiento de los requisitos acordados con la calidad requerida. Se muestran las tareas de ingeniería determinadas para cada historia de usuario con vistas a su implementación. Se presenta además el estándar de codificación utilizada, así como una breve descripción de las pruebas realizadas a la solución y los resultados de su ejecución.

#### 3.1 Capa de Servicios Web para la aplicación FCM-Decision

##### 3.1.1 Métodos Públicos

La capa de Servicios Web para la aplicación FCM-Decision presenta los siguientes métodos públicos:

```
public String obtener (int id)
public void agregar (String nombre, String autor, String descripcion)
public String eliminar (int id)
public List<HashMap<String, String>> listar ()
public void agregarNodo (String siglas)
public String Agregar_Conector (int id_modelo, int nodo_inicial, int nodo_final, double peso)
public String Centralidad_Factores (int id_modelo)
public String simular (int id_modelo, String vector, int landa, String algoritmo)
```

##### 3.1.2 Consumo de los servicios

Un correcto consumo de la capa de Servicios Web para la aplicación FCM-Decision viene dada por los siguientes pasos.

**Obtener:** se le pasa por parámetro un id, es decir se selecciona desde el cliente una de las matrices existentes en la base de datos y el servicio devuelve la matriz correspondiente a esa selección.

**Agregar:** se le pasa por parámetro todos los datos correspondientes a un modelo, nodos y conectores que conforma una matriz, todos estos datos son recogidos por el cliente y enviados al servicio que los insertará en la base de datos.

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

Listar: desde el cliente se pueden ver todas las matrices existentes en la base de datos e interactuar con ellas.

Eliminar: se le pasa por parámetro un id, es decir se selecciona desde el cliente una de las matrices existentes en la base de datos y el servicio elimina de la base de dato la matriz correspondiente a ese id o selección.

Centralidad de Factores: se le pasa por parámetro una matriz que puede seleccionar desde el cliente, el servicio calcula la centralidad de esa matriz devolviendo los valores relacionados con la centralidad obtenida.

Simular: se le pasa por parámetro una matriz que puede seleccionar desde el cliente, un vector y una función y el servicio simula esa matriz devolviendo los valores relacionados con la simulación obtenida.

Obtener Gráfica: con los valores obtenidos de la simulación de una matriz, se le pasan por parámetros al servicio obtener gráfica y este devuelve al cliente una serie de datos que el cliente lo utiliza para hacer la representación gráfica a su modo.

### 3.2 Tareas de ingeniería

Entre los artefactos que se generan durante la fase de implementación en el desarrollo de software con la metodología XP se encuentran las tareas de ingeniería. Estas surgen de la división de las historias de usuario en tareas más sencillas, para planificar el trabajo desde un punto de vista técnico. A continuación se muestran las tareas de ingeniería para las historias de usuarios identificadas en el capítulo anterior.

**Tabla. 25 Tarea de Ingeniería Agregar Modelo**

Tarea de Ingeniería	
<b>Número tarea:</b> 1	<b>Número de Historia de Usuario:</b> 1
<b>Nombre tarea:</b> Agregar Modelo	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 3-3-2014	<b>Fecha Fin:</b> 7-3-2014
<b>Programador Responsable:</b> Aylin Santana Muiño	

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

**Descripción:** El sistema debe permitir al cliente que lo consume una entrada de los datos relacionados con un modelo, este cliente a su vez interactúa con el servicio Agregar Matriz y Agrega un modelo.

Tabla. 26 Tarea de Ingeniería Agregar Nodo

Tarea de Ingeniería	
<b>Número tarea:</b> 2	<b>Número de Historia de Usuario:</b> 1
<b>Nombre tarea:</b> Agregar Nodo	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 8-3-2014	<b>Fecha Fin:</b> 11-3-2014
<b>Programador Responsable:</b> Aylin Santana Muiño	
<b>Descripción:</b> El sistema debe permitir al cliente que lo consume una entrada de los datos relacionados con los nodos, estos nodos son los que se utilizarán en cada matriz que se vaya a agregar.	

Tabla. 27 Tarea de Ingeniería Agregar Conector

Tarea de Ingeniería	
<b>Número tarea:</b> 3	<b>Número de Historia de Usuario:</b> 1
<b>Nombre tarea:</b> Agregar Conector	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 12-3-2014	<b>Fecha Fin:</b> 17-3-2014
<b>Programador Responsable:</b> Aylin Santana Muiño	
<b>Descripción:</b> El sistema debe permitir al cliente que lo consume una entrada de los datos relacionados con los conectores de los nodos ya agregado la relación de cada nodo y el peso que contiene cada relación.	

Tabla. 28 Tarea de Ingeniería Listar Matrices

Tarea de Ingeniería	
<b>Número tarea:</b> 4	<b>Número de Historia de Usuario:</b> 2
<b>Nombre tarea:</b> ListarMatrices	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA



<b>Fecha inicio:</b> 18-3-2014	<b>Fecha Fin:</b> 24-3-2014
<b>Programador Responsable:</b> Aylin Santana Muiño	
<b>Descripción:</b> El sistema debe permitir al cliente que lo consume Devolver una lista de todas las matrices existentes en la base de datos.	

**Tabla. 29 Tarea de Ingeniería obtener matriz dado un id**

Tarea de Ingeniería	
<b>Número tarea:</b> 5	<b>Número de Historia de Usuario:</b> 3
<b>Nombre tarea:</b> Listar Matrices	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 25-3-2014	<b>Fecha Fin:</b> 31-3-2014
<b>Programador Responsable:</b> Aylin Santana Muiño	
<b>Descripción:</b> El sistema debe permitir al cliente que lo consume Devolver una matriz existente en la base de dato dado un id.	

**Tabla. 30 Tarea de Ingeniería Centralidad de factores por id matriz**

Tarea de Ingeniería	
<b>Número tarea:</b> 6	<b>Número de Historia de Usuario:</b> 4
<b>Nombre tarea:</b> Centralidad de factores por id	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 1-4-2014	<b>Fecha Fin:</b> 5-4-2014
<b>Programador Responsable:</b> Erichel Almanza Castellanos	
<b>Descripción:</b> El sistema debe permitir al cliente que lo consume calcular la centralidad de una matriz almacenada en la base de datos entrando su id y devolver los datos del cálculo.	

**Tabla. 31 Tarea de Ingeniería Centralidad de factores por matriz**

Tarea de Ingeniería	
<b>Número tarea:</b> 7	<b>Número de Historia de Usuario:</b> 5
<b>Nombre tarea:</b> Centralidad de factores por matriz	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 6-4-2014	<b>Fecha Fin:</b> 10-4-2014

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

<b>Programador Responsable:</b> Erichel Almanza Castellanos
<b>Descripción:</b> El sistema debe permitir al cliente que lo consume calcular la centralidad de una matriz entrada como parámetro y devolver los datos del cálculo.

**Tabla. 32 Tarea de Ingeniería Realizar simulación por ID matriz**

Tarea de Ingeniería	
<b>Número tarea:</b> 8	<b>Número de Historia de Usuario:</b> 6
<b>Nombre tarea:</b> Realizar simulación por id matriz	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 11-4-2014	<b>Fecha Fin:</b> 14-4-2014
<b>Programador Responsable:</b> Erichel Almanza Castellanos	
<b>Descripción:</b> El sistema debe permitir al cliente que lo consume realizar la simulación de una matriz dado su id y devolver todas las iteraciones de acuerdo al algoritmo seleccionado.	

**Tabla. 33 Tarea de Ingeniería Realizar simulación por matriz**

Tarea de Ingeniería	
<b>Número tarea:</b> 9	<b>Número de Historia de Usuario:</b> 7
<b>Nombre tarea:</b> Realizar simulación por matriz	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 15-4-2014	<b>Fecha Fin:</b> 19-4-2014
<b>Programador Responsable:</b> Erichel Almanza Castellanos	
<b>Descripción:</b> El sistema debe permitir al cliente que lo consume realizar la simulación de una matriz entrada por parámetro y devolver todas las iteraciones de acuerdo al algoritmo seleccionado.	

**Tabla. 34 Tarea de Ingeniería Obtener resultados de la simulación**

Tarea de Ingeniería	
<b>Número tarea:</b> 10	<b>Número de Historia de Usuario:</b> 8
<b>Nombre tarea:</b> Obtener resultado de la simulación	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 20-4-2014	<b>Fecha Fin:</b> 25-4-2014
<b>Programador Responsable:</b> Erichel Almanza Castellanos	

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

**Descripción:** El sistema debe permitir al cliente que lo consume devolver todas las iteraciones de acuerdo al algoritmo seleccionado.

Tabla. 35 Tarea de Ingeniería Obtener gráfica

Tarea de Ingeniería	
Número tarea: 11	Número de Historia de Usuario: 9
Nombre tarea: Obtener gráfica	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 26-4-2014	Fecha Fin: 30-4-2014
Programador Responsable: Erichel Almanza Castellanos	
Descripción: El sistema debe permitir al cliente que lo consume devolver todos los datos correspondientes a la simulación para ser representados gráficamente.	

Tabla. 36 Tarea de Ingeniería Eliminar matriz

Tarea de Ingeniería	
Número tarea: 12	Número de Historia de Usuario: 10
Nombre tarea: Eliminar Matriz	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 1-5-2014	Fecha Fin: 5-5-2014
Programador Responsable: Erichel Almanza Castellanos	
Descripción: El sistema debe permitir al cliente que lo consume poder eliminar una matriz.	

### 3.3 Estándar de codificación

El código debe ser desarrollado siguiendo estándares de desarrollo para facilitar su lectura y la modificación por cualquier miembro del equipo de desarrollo. Para poder llevar a cabo con éxito la propiedad colectiva del código los estándares de codificación son decisivos. La propiedad colectiva sería incompleta sin una codificación basada en estándares que haga que todo el mundo se sienta cómodo con el código escrito por cualquier otro miembro del equipo.

El propósito fundamental de los estándares de codificación es que **GrafoDifuso** tenga una arquitectura y un estilo consistente, independiente del autor, con lo cual el sistema resulte fácil de entender y por supuesto fácil de mantener. El mayor problema es el tratar de entender el formato y el estilo utilizado en el código



## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

escrito por otros desarrolladores. Los estándares de codificación son un complemento a la programación por pares y no sólo es importante tener un estándar, sino tener un buen estándar que para ello deberá:

- Clarificar más que confundir.
- Promover la intención del código.
- Permitir que los programas se acerquen lo mejor posible al lenguaje natural.
- Incorporar las mejores prácticas de la codificación.

La metodología XP promueve la programación basada en estándares, de manera que sea fácilmente entendible por todo el equipo y que facilite la recodificación. A continuación se presenta cada punto a cumplir del estándar utilizado y un ejemplo de su aplicación.

### 3.3.1. Identación

La indentación del código permite que este sea más legible al ganar en organización. Se deben garantizar los siguientes elementos:

- Longitud de la línea: líneas de menos de 80 caracteres.
- Rompiendo líneas: cuando una expresión tenga más de 80 caracteres, romperla de acuerdo con los siguientes principios:
  - Romper después de una coma.
  - Alinear la nueva línea con el comienzo de la expresión al mismo nivel de la línea anterior.

La siguiente figura muestra un ejemplo de ruptura de línea correcta.

```
List<Double> vector_resultado = GrafoUtil.vectorResultado(matriz, vector_entrada,  
vectors, landa);  
iguales = GrafoUtil.vectoresIguales(vector_entrada, vector_resultado);
```

Figura 3: Ruptura de línea correcta de los servicios.

### 3.3.2. Comentarios

Es conveniente comentar información que pueda ser leída tiempo después por desarrolladores (incluido el mismo que programó), que necesitan entender qué fue lo que se hizo en el fragmento de código. Los comentarios deben ser escritos de forma clara en una sola línea. La figura 4 muestra un ejemplo.

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

```
// Metodo calcular centralidad de factores
@Override
public String centralidadFactores(int id_modelo) {
    SessionFactory sessionFactory = HibernateUtil.getSessionFactory();
    Session session = sessionFactory.openSession();
    Transaction tx = session.beginTransaction();
```

Fig. 8 Comentario del método calcular Centralidad

### 3.3.3. Sentencias

Las sentencias realizadas deben cumplir con las siguientes normas:

- En las sentencias WHILE debe haber un espacio en blanco entre la palabra reservada WHILE y el paréntesis de la condición, así como entre el paréntesis que cierra dicha condición y las llaves de la sentencia. Las sentencias a ejecutar dentro de esta deben aparecer en la línea siguiente, cumpliendo las reglas de indentación según las normas definidas en el epígrafe 3.2.1, como muestra la figura 5.

```
while (condicion) {
    sentencias
}
```

Fig. 9 Ejemplo correcto de sentencia WHILE

- En las sentencias IF-ELSE debe haber siempre un espacio en blanco entre la palabra reservada IF y el paréntesis de la condición, entre el paréntesis que cierra dicha condición y las llaves de la sentencia, así como entre la palabra reservada ELSE y las llaves que le anteceden y prosiguen. Las sentencias a ejecutar deben aparecer en la línea siguiente, cumpliendo las reglas de indentación según las normas definidas en el epígrafe 3.2.1, como muestra la figura 6.

```
if (condición) {
    sentencias
}
if (condición) {
    sentencias
} else {
    sentencias
}
```

Fig. 10 Ejemplo correcto de sentencia IF, IF-ELSE

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

### 3.3.4. Espacios en blanco

Las líneas en blanco mejoran la facilidad de lectura separando secciones de código que están lógicamente relacionadas. Se debe usar siempre una línea en blanco en las siguientes circunstancias:

- Entre funciones.
- Entre las distintas secciones lógicas de una función para facilitar la lectura.

```
public String Agregar_Conector(int id_modelo, int nodo_inicial, int nodo_final, double  
public String Centralidad_Factores(int id_modelo);  
public String simular(int id_modelo, String vector, int landa, String algoritmo);
```

Fig. 11 Ejemplo de espacios en blanco entre las funciones de la interface

Se deben usar espacios en blanco en las siguientes circunstancias:

- Una palabra clave del lenguaje seguida por un paréntesis debe separarse por un espacio.
- Debe aparecer un espacio en blanco después de cada coma en las listas de argumentos.

```
Modelo modelo = (Modelo) session.load (Modelo.class, id_modelo);  
Nodo nodo_i = (Nodo) session.load (Nodo.class, nodo_inicial);  
Nodo nodo_f = (Nodo) session.load (Nodo.class, nodo_final);
```

Fig. 12 Ejemplo correcto de aplicación de espacios en blanco

### 3.3.5. Convenciones de nombres

Para la implementación deben emplearse nombres que den idea de lo que realiza determinada función o lo que almacena determinada variable u objeto, con el fin de que el código sea legible. Se deben cumplir con las siguientes normas para los nombres:

- Los nombres de las funciones deben ser sugerentes, cuando son compuestos tendrán la primera letra de cada palabra que lo forma en mayúsculas y se utilizarán guiones bajos (\_) para separarlas. Se debe intentar mantener los nombres de las clases simples y descriptivas, usar palabras completas, así como evitar acrónimos y abreviaturas. La figura siguiente muestra ejemplos.

```
public String Agregar_Conector(int id_modelo, int nodo_inicial, int nodo_final, double  
public String Centralidad_Factores(int id_modelo);  
public String simular(int id_modelo, String vector, int landa, String algoritmo);
```

Fig. 13 Ejemplo correcto de convención de nombres de las funciones de la interface.

### 3.4 Implementación de las historias de usuario

Como resultado de las tareas de ingeniería, derivadas de las historias de usuario definidas, se obtuvo la Capa de Servicios Web para la aplicación FCM-Decision. La capa está compuesta por 10 servicios, los que basándose en las funcionalidades de la aplicación de escritorio FCM-Decision, permite brindar las ventajas de esta aplicación a cualquier cliente interesado en el tema de los mapas cognitivos difusos.

### 3.5. Pruebas

El uso de cualquier producto de software tiene que estar justificado por las ventajas que ofrece. Sin embargo, antes de empezar a usarlo es muy difícil determinar si sus ventajas realmente justifican su empleo. El único modo adecuado para determinar el estado de la calidad de un producto es el proceso de pruebas. El mejor instrumento para esta determinación es la llamada prueba de aceptación.

#### 3.5.1. Pruebas de aceptación

En esta prueba se evalúa el grado de calidad del producto con relación a todos los aspectos relevantes para que el uso de este se justifique. Estas pruebas las realiza el cliente. Son básicamente pruebas funcionales sobre el sistema completo y buscan una cobertura de la especificación de requisitos y del manual de usuario. (21)

Para determinar la calidad de la extensión desarrollada se aplicarán pruebas de caja negra.

#### Pruebas de caja negra

Las pruebas de caja negra, también denominadas pruebas de comportamiento, se llevan a cabo sobre la interfaz del producto, obviando el comportamiento interno y la estructura del programa. Al aplicar pruebas de caja negra se derivan un conjunto de casos de pruebas que involucran condiciones de entrada que utilizan todos los requisitos funcionales de un programa. Los casos de prueba de la caja negra pretenden demostrar que: (21)

- Las funciones del software son operativas.
- La entrada se acepta de forma correcta.
- Se produce una salida correcta.
- La integridad de la información externa se mantiene.

Las pruebas de caja negra pretenden encontrar los tipos de errores siguientes: (21)

- Funciones incorrectas o ausentes.

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

- Errores en la interfaz.
- Errores en estructuras de datos o en accesos a bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y de terminación.

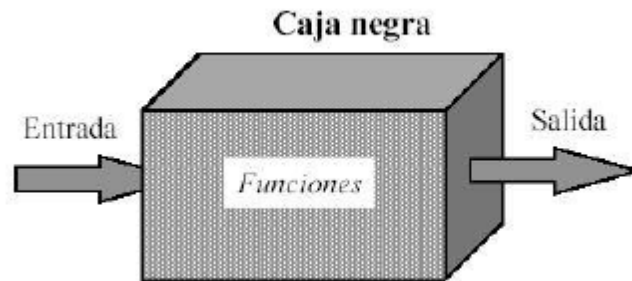


Fig. 14 Prueba de caja negra

Para confeccionar los casos de prueba de caja negra existen distintos métodos, el empleado en este proceso de prueba será el de partición de equivalencia.

### Partición de equivalencia

La partición de equivalencia es un método de pruebas de caja negra que divide el campo de entrada de un programa en clases de datos, de los que se pueden derivar casos de prueba. La partición equivalente se dirige a una definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar.

Este método intenta dividir el dominio de entrada de un programa en un número finito de clases de equivalencia. De tal modo que se pueda asumir razonablemente que una prueba realizada con un valor representativo de cada clase es equivalente a una prueba realizada con cualquier otro valor de dicha clase. Esto quiere decir que si el caso de prueba correspondiente a una clase de equivalencia detecta un error, el resto de los casos de prueba de dicha clase de equivalencia deben detectar el mismo error. Y viceversa, si un caso de prueba no ha detectado ningún error, es de esperar que ninguno de los casos de prueba correspondientes a la misma clase de equivalencia encuentre ningún error. (21)

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

El objetivo principal de la fase de pruebas es encontrar la mayor cantidad de errores y defectos posibles, por lo que es conveniente que las pruebas de aceptación sean realizadas por el propio usuario final y no por el desarrollador del producto. Con ese fin serán aplicadas a los servicios desarrollados pruebas alfa.

### Prueba alfa

Se lleva a cabo, por un cliente en el lugar de desarrollo. Se usa el software de forma natural con el desarrollador como observador del usuario. Las pruebas alfa se llevan a cabo en un entorno controlado. Para que tengan validez, se debe primero crear un ambiente con las mismas condiciones que se encontrarán en las instalaciones del cliente. Una vez logrado esto, se procede a realizar las pruebas. (21)

### 3.5.2. Diseño de casos de prueba y registro de no conformidades

Los casos de prueba se realizan para validar el correcto funcionamiento de la aplicación y así hacer la correcta liberación del producto. A continuación se realiza el caso de prueba de Caja Negra para la historia de usuario Agregar Matriz, el cual consiste en comprobar cómo se comporta el sistema a medida que vamos cambiando el set de datos utilizado.

Para detallar la historia de usuario se utiliza una tabla, donde se desglosan estas en secciones y a su vez en escenarios para hacer más fructífera la ejecución de las pruebas. Esta tabla contiene los campos:

- Nombre de la sección: se especifica el nombre de la sección [SC 1: Nombre de la sección].
- Escenarios de la sección: se especifican los escenarios de cada sección [EC 1.1: Nombre del Escenario].
- Descripción de la funcionalidad: se describe brevemente la funcionalidad del escenario.

El siguiente es un ejemplo donde se detalla la historia de usuario Agregar Matriz.

**Tabla. 37 Secciones para probar la historia de usuario Agregar Matriz**

Nombre de sección	Escenarios de la sección	Descripción de la funcionalidad
SC1: Agregar modelo	EC 1.1 Agregar Modelo con datos correctos	El usuario llena todos los datos correspondientes a un modelo, el sistema muestra un mensaje indicando que el modelo ha sido agregado.
	EC 1.2 Agregar Modelo con datos incorrectos	El usuario llena todos los datos correspondientes a un modelo de forma incorrecta, el sistema muestra un mensaje indicando que ha ocurrido un error.

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

	EC 1.3 Agregar Modelo con datos en blanco	El usuario deja campos en blancos en el formulario.
SC2: Agregar Nodo	EC 2.1 Agregar Nodo con datos correctos	El usuario llena todos los datos correspondientes a un nodo, el sistema muestra un mensaje indicando que el o los nodos han sido agregados de manera correcta.
	EC 2.2 Agregar Nodo con datos incorrectos	El usuario llena todos los datos correspondientes a un nodo de forma incorrecta, el sistema muestra un mensaje indicando que ha ocurrido un error.
	EC 2.3 Agregar Nodo con datos en blanco	El usuario deja campos en blancos en el formulario.
SC3: Agregar Conector	EC 3.1 Agregar Conector con datos correctos	El usuario llena todos los datos correspondientes a los conectores de cada nodo y el peso correspondiente a cada conector, el sistema muestra un mensaje indicando que los conectores han sido agregados.
	EC 3.2 Agregar Conector con datos incorrectos	El usuario llena todos los datos correspondientes a los conectores de cada nodo y el peso correspondiente a cada conector de manera incorrecta, el sistema muestra que ha ocurrido un error.
	EC 3.3 Agregar Conector con datos en blanco	El usuario deja campos en blancos en el formulario.

A partir de esta descripción se detallan las variables asociadas a la historia de usuario. Esta descripción está compuesta por los campos:

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

- No: se enumeraron todos los campos o variables, descritos en el caso de prueba.
- Nombre de campo: se especificó el nombre del campo de entrada.
- Clasificación: se especificó la clasificación según el componente de diseño utilizado, ejemplo: texto, consulta o arreglo.
- Nulo: se especificó si el campo puede ser nulo o no, para ello solo se puso Sí o No.
- Descripción: se describieron brevemente los datos que debían introducirse.

**Tabla. 38 Descripción de variables de la historia de usuario Agregar Matriz**

No	Nombre del campo	Clasificación	Nulo	Descripción
1	nombre	texto	si	Indica el nombre de un modelo
2	autor	texto	si	Indica el autor de ese modelo
3	fecha	date	si	Indica la fecha en que se confecciona ese modelo
4	descripción	texto	si	Indica la descripción correspondiente al modelo
5	siglas	texto	No	Indica las siglas correspondiente a los nodos insertados
6	nodoByNodoFinal	entero	No	Indica el id correspondiente al nodo final de un conector
7	nodoByNodoInicial	entero	No	Indica el id correspondiente al nodo inicial de un conector
8	valorPeso	real	No	Indica el valor correspondiente al peso de un conector entre dos nodos



## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

Esta descripción facilitó que se obtuvieran los datos, donde se evaluó y probó la validez de cada uno de los datos introducidos en el servicio web Agregar matriz. Utilizando un juego de datos válidos e inválidos se identificó el empleo de la técnica de partición de equivalencia.

**Tabla. 39 Datos de la historia de usuario Agregar Matriz**

Id del escenario	Escenario	Variables enumeradas según tabla 40								Respuesta del sistema	Flujo central	
		1	2	3	4	5	6	7	8			
EC1.1	Agregar Modelo con datos correctos	V	V	V	V						El servicio inserta en la base de datos los valores entrados por parámetros. El sistema emite un mensaje mostrando que se han insertado correctamente e los valores en la base de datos	<ol style="list-style-type: none"> <li>1. El usuario llena todos los parámetros correspondientes a un modelo.</li> <li>2. El servicio inserta un modelo y emite un mensaje indicando que se han insertado correctamente los datos en la base de datos.</li> </ol>
		Mprueba	Pepe	-	Probando							
EC1.2	Agregar Modelo con datos incorrecto	I	V	V	V							
		Prueba2	12323	-	Probando						El sistema muestra un mensaje	1.El usuario llena todos los parámetros

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

											indicando un error	2. El sistema emite un mensaje indicando que ha ocurrido un error
EC1.3	Agregar Modelo con datos en blanco	-	-	-	-						El sistema muestra un mensaje indicando Campos en blancos	1. El usuario deja los campos vacíos no entrando parámetros. 2. El sistema emite un mensaje indicando que es necesario llenar los campos antes de insertar.

Como resultado de la aplicación de las pruebas, se detectaron un conjunto de no conformidades. Se muestran en la tabla siguiente las asociadas a la historia de usuario agregar matriz, específicamente el escenario agregar modelo.

- No: se especifica el número de la no conformidad.
- No conformidad NC: se describe la no conformidad.
- Aspecto asociado: se especifica el aspecto correspondiente a la no conformidad.
- Etapa de detección: se especifica la etapa de detección del error.
- Clasificación: se pone una (S), en caso de que la no conformidad estuviera clasificada como significativa, (NS), en caso de que la no conformidad estuviera clasificada como no significativa y una (R), en caso de que la no conformidad solo fuera una recomendación.
- Estado NC: se coloca el estado de la no conformidad y la fecha, cada vez que se revisa se deja el estado anterior y se coloca el nuevo con la fecha en que se revisó [RA: Resuelta, PD: Pendiente, NP: No Procede].

**Tabla.40 No conformidades detectadas en la aplicación del caso de pruebas Agregar matriz, específicamente el escenario agregar modelo.**

No	NC	Aspecto asociado	Etapa de detección	Clasificación	Estado NC	Respuesta
1	Acepta en el campo nombre de modelo	Agregar modelo	Pruebas	S	5/05/2014 PD	Corregido

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

datos de tipo NULL, mostrándose un modelo desconocido cuando lo mandas a llamar				5/05/2014 RA	
--	--	--	--	-----------------	--

### **3.5.3 Resultados de las pruebas de aceptación y prueba Alfa**

La muestra de los resultados de las pruebas de aceptación es importante, la exhibición de los resultados que se obtienen al ejecutar las pruebas de aceptación, generando gráficas que muestren los porcentajes de efectividad obtenidos. Estos valores obtenidos permiten evaluar si el equipo de desarrollo está realizando un buen trabajo o no.

Se acudió a la prueba Alfa, para garantizar una mayor profundidad a la hora de buscar no conformidades, siendo estas pruebas realizadas por los clientes y los desarrolladores como observadores de los usuarios en un ambiente natural como lo define Pressman. En esta investigación se diseñaron un total de 10 casos de pruebas, como resultado de las pruebas aplicadas. Se realizó una primera iteración de pruebas, en la cual fueron probados los casos de pruebas correspondientes a las historias de usuario de prioridad Muy Alta, detectándose un total de 8 no conformidades significativas, 1 no conformidad no significativa y 2 recomendaciones. En una segunda iteración fueron probados los casos de pruebas de prioridad Alta, detectándose 3 no conformidades no significativas y 1 recomendación. Se realizó una tercera iteración, donde fueron probados nuevamente el total de casos de pruebas diseñados para asegurar la calidad del producto, en la cual no se detectaron no conformidades. La siguiente gráfica muestra el resultado de la aplicación de las pruebas.

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

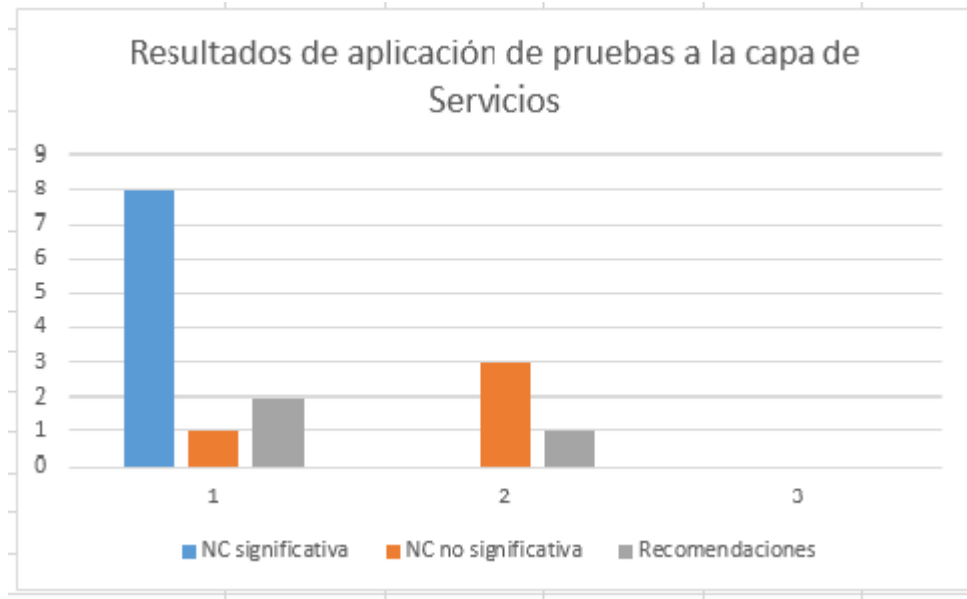


Fig. 15 Resultados de las pruebas

### Conclusiones del capítulo

En este capítulo se definieron 12 tareas de ingeniería correspondientes a las historias de usuario identificadas. Se definió el estándar de codificación, se exponen los resultados de las pruebas de aceptación realizadas a la capa de servicios para la aplicación FCM-Decision mediante la técnica de caja negra, específicamente mediante el método de partición de equivalencia. Se realizó tres iteraciones de prueba en las cuales se encontró para la primera iteración 8 no conformidades significativas, 1 no conformidad no significativa y 2 recomendaciones, para la segunda iteración se detectaron 3 no conformidades no significativas y 1 recomendaciones, en la tercera iteración no se encontraron no conformidades las cuales fueron resueltas demostrando que la capa de servicios desarrollada cumple con las expectativas del cliente.

# CONCLUSIONES GENERALES



## CONCLUSIONES GENERALES

Como resultado de la investigación se obtuvo la capa de servicios para la aplicación FCM-Decision arribándose a las siguientes conclusiones:

- La elaboración del marco teórico referencial permitió la correcta selección de la metodología y las herramientas para el desarrollo de capa de servicios para la aplicación FCM-Decision.
- Se realizó el análisis y el diseño de la capa de servicios para la aplicación FCM-Decision, centralizando, estandarizando y ofreciendo información detallada de las funcionalidades que brinda la aplicación, obteniéndose de esta los artefactos correspondientes a cada fase de la metodología propuesta.
- Se implementó una capa de servicios web para la aplicación FCM-Decision permitiendo a los usuarios acceder a las funcionalidades de esta aplicación desde diferentes estaciones de trabajo de forma simultánea.
- Se realizaron y ejecutaron los casos de prueba para validar el correcto funcionamiento de la capa de servicios web para la aplicación FCM-Decision, resolviéndose todas las no conformidades detectadas y terminando la presente investigación con resultados satisfactorios.

# RECOMENDACIONES



## RECOMENDACIONES

Después de lograr los objetivos planteados que se trazaron al principio del presente trabajo se plantea la siguiente recomendación:

- Incorporar un mecanismo de seguridad para el Servicio Web capa de servicios para la aplicación FCM-Decision implementado que permita hacer frente a posibles ataques o intrusiones malignas al sistema desarrollado.

# REFERENCIAS BIBLIOGRÁFICAS

## REFERENCIAS

1. **Morales, Mirian Untiveros.** *Lógica Difusa.*
2. *Técnicas para la representación del conocimiento causal: un estudio de caso en Informática Médica.* **Leyva Vázquez, Maikel Y. y Pérez Teruel, Karina.** 2013.
3. **Porras, Pablo Arturo Von Raesfeld.** *Mapas Cognitivos Difusos .* Cholula, Puebla, México : s.n., 19 de mayo de 2010.
4. **Márquez, Santiago.** *Introducción a la creación de servicios Web con .NET.* Madrid, España : Joaquin Medina Serrano, jueves, 23 de agosto de 2009.
5. **Zadeh, Lofty A.** *Hiperenciclopédica.* La Vanguardia, Barcelona : s.n., 23 noviembre 2012.
6. **Rosario.** tecnología educativa. [En línea] martes 17 de noviembre de 2009. [madelrosario123.blogspot.com/2009/11/tipos-de-mapas-cognitivos.html](http://madelrosario123.blogspot.com/2009/11/tipos-de-mapas-cognitivos.html).
7. **Web Service for PHP, Active Corporation.** Ecured. [En línea] 10 de Julio de 2009. [http://www.ecured.cu/index.php/Servicios\\_Web](http://www.ecured.cu/index.php/Servicios_Web).
8. **Mendoza, Yosbel Fonseca.** *FCM-Decision: MÓDULO DISEÑO AUTOMÁTICO DE MAPAS COGNITIVOS DIFUSOS.* .
9. **Metodología XP.** [En línea] <https://sites.google.com/site/xpmetodologia/marco-teorico>.
10. **Frias, Pablo.** Mundo Java. [En línea] domingo 8 de abril de 2009. <http://mundojava.blogspot.com/2007/04/aprendiendo-eclipse-32.html>.
11. —. Programando en Java. [En línea] 2009. <http://www.lenguajes-de-programacion.com/programacion-avanzada-en-java.shtml>.
12. **RFIDPOINT.** Middleware . [En línea] 2014 . <http://www.rfidpoint.com/fundamentos/middleware/>.
13. **Foundation, The Apache Software.** Apache Tomcat. [En línea] 23 de mayo de 2014. <http://tomcat.apache.org/>.
14. **AB, Autor MySQL.** MySQL Workbench. [En línea] <http://mysql-workbench.uptodown.com/>.
15. **Manual de Operación de Mysql Workbench.** [En línea] <http://doocument.com/pdf/manual-de-operacin-de-mysql-workbench-41328.pdf>.
16. **Community, Hibernate.** Hibernate. Everything data. [En línea] <http://hibernate.org/>.
17. **Norman, Donald.** Jummp. [En línea] 3 de junio de 2014. <http://jummp.wordpress.com/>.

## REFERENCIAS BIBLIOGRÁFICAS

18. Haw, Nedoww. Slideshare. [En línea] 28 de enero de 2013. <http://www.slideshare.net/nedowwhaw/diagrama-de-clases-16208245>.
19. Wilson, Jose Manuel Ayala. Blog sobre arquitectura. [En línea] <http://jmaw.blogspot.com/2011/04/h2-margin-bottom-0.html>.
20. —. Patrones de Arquitectura. [En línea] <http://jmaw.blogspot.com/2011/04/h2-margin-bottom-0.html>.
21. Pressman, Roger S. Ingeniería de Software. Un enfoque práctico.
22. [http://www.ecured.cu/index.php/Servicios\\_Web](http://www.ecured.cu/index.php/Servicios_Web). *Ecured*. [En línea]
23. ZAMORA, MSC. VIANNEY MORALES. *PROGRAMACIÓN LÓGICA Y FUNCIONAL*. 2013.
24. *Diccionario de informática*. Definición de Multiplataforma.
25. [En línea] <http://definicion.de/causalidad/>.
26. Pascal Chevrel. Definición de la Interoperabilidad. [En línea] <http://interoperability-definition.info/es/>.
27. Porras, Pablo Arturo Von Raesfeld. *Capítulo 2 mapas cognitivos difusos*. Cholula, Puebla, México : s.n., 19 de mayo de 2010 .



# BIBLIOGRAFÍA

## BIBLIOGRAFÍA

1. **Morales, Mirian Untiveros.** *Lógica Difusa.*
2. *Técnicas para la representación del conocimiento causal: un estudio de caso en Informática Médica.* **Leyva Vázquez, Maikel Y. y Pérez Teruel, Karina.** 2013.
3. **Porras, Pablo Arturo Von Raesfeld.** *Mapas Cognitivos Difusos .* Cholula, Puebla, México : s.n., 19 de mayo de 2010.
4. **Márquez, Santiago.** *Introducción a la creación de servicios Web con .NET.* Madrid, España : Joaquin Medina Serrano, jueves, 23 de agosto de 2009.
5. **Zadeh, Lofty A.** *Hiperenciclopédica.* La Vanguardia, Barcelona : s.n., 23 noviembre 2012.
6. **Rosario.** tecnología educativa. [En línea] martes 17 de noviembre de 2009. [madelrosario123.blogspot.com/2009/11/tipos-de-mapas-cognitivos.html](http://madelrosario123.blogspot.com/2009/11/tipos-de-mapas-cognitivos.html).
7. **Web Service for PHP, Active Corporation.** Ecured. [En línea] 10 de Julio de 2009. [http://www.ecured.cu/index.php/Servicios\\_Web](http://www.ecured.cu/index.php/Servicios_Web).
8. **Mendoza, Yosbel Fonseca.** *FCM-Decision: MÓDULO DISEÑO AUTOMÁTICO DE MAPAS COGNITIVOS DIFUSOS.* .
9. **Metodología XP.** [En línea] <https://sites.google.com/site/xpmetodologia/marco-teorico>.
10. **Frias, Pablo.** Mundo Java. [En línea] domingo 8 de abril de 2009. <http://mundojava.blogspot.com/2007/04/aprendiendo-eclipse-32.html>.
11. —. Programando en Java. [En línea] 2009. <http://www.lenguajes-de-programacion.com/programacion-avanzada-en-java.shtml>.
12. **RFIDPOINT.** Middleware . [En línea] 2014 . <http://www.rfidpoint.com/fundamentos/middleware/>.
13. **Foundation, The Apache Software.** Apache Tomcat. [En línea] 23 de mayo de 2014. <http://tomcat.apache.org/>.
14. **AB, Autor MySQL.** MySQL Workbench. [En línea] <http://mysql-workbench.uptodown.com/>.
15. **Manual de Operación de Mysql Workbench.** [En línea] <http://doocument.com/pdf/manual-de-operacin-de-mysql-workbench-41328.pdf>.
16. **Community, Hibernate.** Hibernate. Everything data. [En línea] <http://hibernate.org/>.
17. **Norman, Donald.** Jummp. [En línea] 3 de junio de 2014. <http://jummp.wordpress.com/>.

## BIBLIOGRAFÍA

18. Haw, Nedoww. Slideshare. [En línea] 28 de enero de 2013. <http://www.slideshare.net/nedowwhaw/diagrama-de-clases-16208245>.
19. Wilson, Jose Manuel Ayala. Blog sobre arquitectura. [En línea] <http://jmaw.blogspot.com/2011/04/h2-margin-bottom-0.html>.
20. —. Patrones de Arquitectura. [En línea] <http://jmaw.blogspot.com/2011/04/h2-margin-bottom-0.html>.
21. Pressman, Roger S. Ingeniería de Software. Un enfoque práctico.
22. [http://www.ecured.cu/index.php/Servicios\\_Web](http://www.ecured.cu/index.php/Servicios_Web). *Ecured*. [En línea]
23. ZAMORA, MSC. VIANNEY MORALES. *PROGRAMACIÓN LÓGICA Y FUNCIONAL*. 2013.
24. *Diccionario de informática*. Definición de Multiplataforma.
25. [En línea] <http://definicion.de/causalidad/>.
26. Pascal Chevrel. Definición de la Interoperabilidad. [En línea] <http://interoperability-definition.info/es/>.
27. Porras, Pablo Arturo Von Raesfeld. *Capítulo 2 mapas cognitivos difusos*. Cholula, Puebla, México : s.n., 19 de mayo de 2010 .

# GLOSARIO DE TÉRMINOS



## GLOSARIO DE TÉRMINOS

**Causalidad:** La causalidad es el principio o el origen de algo. El concepto se utiliza para nombrar a la relación entre una causa y su efecto y puede utilizarse en el ámbito de la física, la estadística y la filosofía. De acuerdo al principio de causalidad, todo efecto siempre tiene una causa. El principio de uniformidad agrega que, en idénticas circunstancias, una causa siempre produce el mismo efecto.

**Centralidad de Factores:** Es un resultado que se obtiene de la matriz de adyacencia, son cálculos que se obtiene de esta matriz devolviendo resultado que ayudan a la toma de decisiones influyendo en el peso de cada nodo.

**Simulación:** La simulación es una técnica para analizar y estudiar sistemas complejos. Nos permite reunir información pertinente sobre el comportamiento del sistema porque ejecuta un modelo computarizado.

**Middleware:** Es un software de conectividad que ofrece un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas.

**Emulen:** Contender, luchar, superar, rivalizar.

**Retroalimentación:** Proceso de un sistema que es mantenido por uno de sus productos finales.

**Heterogeneidad:** Cualidad de una cosa heterogénea o formada por elementos de distinta clase o naturaleza.