



Facultad 5

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS.

***RED SOCIAL NEXO PARA LA TOMA DE
DECISIONES EN GRUPO BASADO EN
MAPAS COGNITIVOS DIFUSOS
UTILIZANDO FCM – FUSION.***

AUTORES:

YALENA AILETH VENEGA CAÑIZARES.
ALBERTO JESÚS PÉREZ GALBAN.

TUTORES:

DR. MAIKEL YELANDI LEYVA VÁZQUEZ.
DRA. KARINA PÉREZ TERUEL.

Co – TUTOR:

ING. JULIO CÉSAR ESPRONCEDA PÉREZ.

LA HABANA, 2015
"AÑO 57 DE LA REVOLUCIÓN"





Seamos realistas y hagamos lo imposible.

efe



DECLARACIÓN DE AUTORÍA.

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas (UCI) los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente: a los ____ días del mes de _____ del año 2015.

Yalena Aileth Venega Cañizares

Alberto Jesús Pérez Galban

Dr. Maikel Yelandi Leyva Vázquez

Tutor

Dra. Karina Pérez Teruel

Tutor

Ing. Julio César Espronceda Pérez

Co – Tutor



DATOS DE LOS TUTORES.

Generales del tutor:

Nombre y apellidos: Maikel Yelandi Leyva Vázquez

Especialidad: Bioinformática.

Síntesis del tutor: Ingeniero en Informática egresado del ISPJAE en el 2005. Máster en Ciencias en la especialidad de Bioinformática en el INSTEC año 2007. Doctor en Ciencias. Profesor Auxiliar del Departamento de Soluciones SOA, del Centro de Consultoría y Desarrollo de Arquitecturas Empresariales.

Correo electrónico: mleyvaz@uci.cu

Teléfono de contacto: (+53) 7 8372731

Generales del tutor:

Nombre y apellidos: Karina Pérez Teruel

Especialidad: Bioinformática.

Síntesis del tutor: Ingeniera en Informática egresado del ISPJAE en el 2005. Máster en Ciencias en la especialidad de Bioinformática en el INSTEC año 2007. Doctor en Ciencias. Profesora Auxiliar del departamento de Ingeniería, Gestión de software y Práctica Profesional de la Facultad 5 de la Universidad de las Ciencias Informáticas (UCI).

Correo electrónico: karinapt@uci.cu

Teléfono de contacto: (+53) 7 8372731

Generales del Co – Tutor:

Nombre y apellidos: Julio César Espronceda Pérez

Especialidad: Informática.

Síntesis del co – tutor: Graduado de Ingeniero en Ciencias Informáticas (2013). Ha participado en eventos nacionales e internacionales y ha publicado varios artículos. Miembro de la Sociedad Cubana de Matemática y Computación. Vicedecano Extensión y Residencia y profesor de la disciplina Ingeniería y Gestión de Software en la Facultad 5 de la Universidad de las Ciencias Informáticas (UCI).

Correo electrónico: jcespronceda@uci.cu

Teléfono de contacto: (+53) 7 8372745



AGRADECIMIENTOS

de Yalena

A mi padre por haberse sacrificado tanto por mí, por estar siempre a mi lado apoyándome en los buenos y malos momentos, dándome su amor y confianza.

A mis abuelas Umbelina y Nene, por darme lo mejor de ellas, por confiar en mí en que este día iba a llegar.

A mi madre por darme la vida y por estar conmigo en los momentos que la necesitaba.

A mis tíos Manuel y Ana por ayudarme a salir adelante en cada situación que la vida me ponía.

A Alberto por estar a mi lado desde 1er año dándome su apoyo incondicional a cada momento, por todas las cosas que vivimos.

A Martha, a Delfin y a Susset por ayudarme a salir adelante y convertirme en mejor persona, abriéndome las puertas de su hogar sin pedir nada a cambio.

A Idia por extenderme su mano y acogerme como una hija para ella.

A un gran amigo, Javier Chao por tantos momentos de risas y tristezas que vivimos juntos.

A mi hermana por compartir conmigo momentos felices.

A mi prima hermana por apoyarme en estos cinco años de la carrera.

A Karina, Maikel y a Julio César por ayudarme y hacer posible la realización de este trabajo. Son excelentes personas, fue un honor de trabajar con ustedes. Gracias.

A mis amistades y todos aquellos que siempre estuvieron ahí para darme el consejo adecuado. Gracias por haber formado parte de mi vida.

A todas las personas que de una forma u otra me apoyaron en la realización de este trabajo.



AGRADECIMIENTOS

de Alberto

*A mi madre por su lucha incansable en la construcción de mi bienestar,
por su amor casi divino, por ser la luz que guía cada uno de mis pasos,
por ser el regalo más grande y especial que me ha dado la vida.*

*A mi padre por ser mi guía, mi paradigma, mi ejemplo de integridad a
lo largo de mi vida, pilar fundamental de mi existencia.*

A mi hermana por su amor especial y su apoyo incondicional.

A mi tía Lourdes, mi eterna maestra y segunda madre.

A mi prima Moma, mi segunda hermana, mi confidente.

A mi madrina Miriam por su cariño, aprecio y constante apoyo.

*A Yalena, mi compañera en este viaje, por no dejarme retroceder jamás,
por alentarme siempre a dar un paso más, por aguantar mi mal humor
y por haber compartido conmigo, de cerquita, cada momento de estos
cinco años.*

A mis tutores y profesores Karina y Maikel por su apoyo y confianza.

*A mi co - tutor, vicedecano, profesor, alumno ayudante, dirigente,
amigo y hermano Julio César, por su inquebrantable apoyo desde que
entre en esta universidad.*

A mis más que amigos, hermanos El Dariel, El Robe y El Yuri.

*A El Micha, a Yadys, a Nene, a Aldo "El Jerry", a Yaniela, a Umbe, a
Mario, a Manolito, a Ana, a Yesenia mi familia de Sancti Spiritus.*

A la profe Zaida, una madre para mí en la UCI.

A todos los profesores que han contribuido en mi formación.

*A Orlando por su ayuda y colaboración en estos años y en especial en la
realización de este trabajo.*

*A El Yase, a Emilíto "El Baby", a Alfonso "Alf", a Cuevas, a mis compañero
de grupo por todos los buenos momentos que pasamos.*

*A todos los que de una forma y otra me han ayudado a convertir este
sueño en una realidad.*



DEDICATORIA

de Yalena

A mi padre que me ha dado su apoyo para seguir adelante en estos 5 años, a mi madre que con su perseverancia me ha dado fuerzas para seguir hasta lograr mi objetivo, a mis abuelas que sus consejos han estado presente en los momentos buenos y malos, confiando siempre en que puedo vencer los obstáculos que se presenten y a mis demás familiares que siempre confiaron en mí.

de Alberto

A mi madre y a mi padre, protagonistas de todos y cada uno de mis logros.



RESUMEN

La representación del conocimiento causal a través de mapas cognitivos difusos ha recibido una creciente atención por la posibilidad de representar modelos mentales tanto individuales como colectivos con cierta facilidad y luego de ser analizados poder llegar a tomar decisiones bajo condiciones de incertidumbre. En el presente trabajo se propone como objetivo el desarrollo de una aplicación web del tipo red social colaborativa usando la metodología de desarrollo Programación Extrema, que posibilite integrar en un mismo sistema: métodos de obtención de mapas, análisis estático, aprendizaje, análisis de escenarios y orientado a metas y basado en el uso de operadores de agregación para evaluar las distintas alternativas. La principal ventaja del sistema es la de posibilitar al usuario interpretar alternativas construidas con el uso del sistema de actividades que propone el modelo “Fusión de Mapas Cognitivos Difusos”, basándose en el conocimiento compartido por otros expertos de algún tema propuesto, siendo esto un hecho de gran utilidad para la toma de decisiones. Finalmente se lleva a cabo la validación de la propuesta a través de los métodos definidos en la investigación.

Palabras clave: análisis de escenarios, mapas cognitivos difusos, red social, toma de decisiones en grupo.



TABLA DE CONTENIDOS.

Introducción	13
Capítulo 1: Fundamentación teórica de la investigación.....	18
1.1 Modelos causales	18
1.2 Mapas cognitivos y mapas cognitivos difusos	19
1.3 Toma de decisiones	20
1.4 Métodos de decisión multicriterio.....	21
1.5 Toma de decisiones en grupo	22
1.6 Agregación de la información	22
1.6.1 Media Aritmética	22
1.6.2 Media Ponderada	23
1.6.3 Media Ponderada Ordenada.....	23
1.6.4 Integrales Difusas.....	23
1.7 Análisis de escenarios	24
1.8 Modelo FCM – Fusion.....	24
1.9 Razonamiento basado en casos.....	25
1.10 Sistemas basados en casos	25
1.11 Herramientas web existentes de ayuda a la toma de decisiones basadas en MCD	26
1.11.1 ESCEANALY	26
1.11.2 CAUSALITY TOOL.....	26
1.12 Redes sociales.....	27
1.13 Metodología de desarrollo de software	28
1.13.1 Proceso Racional Unificado (RUP).....	28
1.13.2 Programación Extrema (XP)	29
1.13.3 Selección de la metodología de desarrollo.....	31
1.14 Herramientas de desarrollo.....	31
1.14.1 Lenguajes de programación de lado del cliente	32
1.14.2 Lenguaje de programación de lado del servidor.....	33
1.14.3 Frameworks de desarrollo.....	33
1.14.5 Entorno de desarrollo integrado:.....	35



1.14.6	Sistema gestor de base de datos.....	36
	Conclusiones parciales.....	37
Capítulo 2: Red Social NEXO para la toma de decisiones en grupo basado en MCD utilizando FCM – Fusion..... 38		
2.1	Red Social NEXO.....	38
2.2	Exploración.....	38
2.2.1	Historias de Usuario.....	39
2.3	Planificación.....	41
2.3.1	Lista de reservas del producto.....	42
2.3.2	Plan de Iteraciones.....	45
2.3.3	Plan de Entrega.....	46
2.4	Iteraciones.....	46
2.4.1	Tarjetas CRC.....	47
2.5	Arquitectura de software.....	48
2.5.1	Patrón Modelo Vista Controlador (MVC).....	48
2.6	Patrones de diseño.....	49
2.6.1	Patrones GRASP.....	50
2.7	Modelo de datos.....	51
	Conclusiones parciales:.....	51
Capítulo 3: Implementación y prueba de la Red Social NEXO..... 53		
3.1	Tareas de Ingeniería.....	53
3.2	Estilos de programación.....	54
3.2.1	Definiciones de clases.....	55
3.2.2	Definición de métodos.....	55
3.2.3	Asignaciones a variables.....	55
3.2.4	Estructuras de control.....	56
3.2.5	Llamada a funciones.....	56
3.3	Diagrama de despliegue.....	57
3.4	Planificación de las Pruebas.....	58
3.5	Pruebas de Caja Blanca.....	61
3.5.1	Resultados.....	64



3.6 Pruebas de Caja Negra	64
3.6.1 Resultados	67
Conclusiones parciales:	68
Conclusiones Generales:	69
Recomendaciones:	69
Referencias Bibliográficas:	70
Anexos:	76

ÍNDICE DE FIGURAS

<i>Figura 1: Mapa cognitivo difuso y su correspondiente matriz de adyacencia.</i>	19
<i>Figura 2: Matriz de decisión.....</i>	20
<i>Figura 3: Operador de agregación.....</i>	22
<i>Figura 4: Funcionamiento de un SBC.....</i>	26
<i>Figura 5: Patrón Modelo Vista Controlador (MVC).</i>	48
<i>Figura 6: Diagrama de clases que representa el patrón Creador.....</i>	50
<i>Figura 7: Diagrama de clases que representa el patrón Controlador.....</i>	51
<i>Figura 8: Definición de clases.</i>	55
<i>Figura 9: Definición de métodos.</i>	55
<i>Figura 10: Asignación a variables.....</i>	56
<i>Figura 11: Estructuras de control.....</i>	56
<i>Figura 12: Llamada a funciones.....</i>	57
<i>Figura 13: Diagrama de Despliegue.....</i>	57
<i>Figura 14: Método que permite el autenticar a los usuarios.....</i>	61
<i>Figura 15: Grafo de flujo asociado al método actionLoguin.</i>	62
<i>Figura 16: Resultados de las PU.</i>	64
<i>Figura 17: Resultados de las PA.....</i>	68



ÍNDICE DE TABLAS

<i>Tabla 1: HU17 Mostrar grafo MCD.</i>	40
<i>Tabla 2: HU18 Mostrar matriz MCD.</i>	40
<i>Tabla 3: HU19 Obtener análisis dinámico.</i>	41
<i>Tabla 4: Lista de reservas del producto.</i>	43
<i>Tabla 5: Requisitos no funcionales.</i>	45
<i>Tabla 6: Plan de iteraciones.</i>	46
<i>Tabla 7: Plan de entregas.</i>	46
<i>Tabla 8: Tarjeta CRC Mapa_Control.</i>	47
<i>Tabla 9: Tarjeta CRC Proyecto_Control.</i>	47
<i>Tabla 10: Tarjeta CRC Vertice_Control.</i>	48
<i>Tabla 11: Tarea de ingeniería 1 HU1 Registrar usuario.</i>	53
<i>Tabla 12: Tarea de ingeniería 4 HU2 Modificar cuenta de usuario.</i>	54
<i>Tabla 13: Tarea de ingeniería 17 HU5 Insertar concepto.</i>	54
<i>Tabla 14: Plan de pruebas.</i>	59
<i>Tabla 15: Caso de Prueba para el Camino Básico #1.</i>	63
<i>Tabla 16: Caso de Prueba para el Camino Básico #2.</i>	64
<i>Tabla 17: Prueba de aceptación para la HU Modificar datos de usuario.</i>	66
<i>Tabla 18: Prueba de aceptación para la HU Adicionar vértice.</i>	66
<i>Tabla 19: Prueba de aceptación para la HU Eliminar notificación.</i>	67



INTRODUCCIÓN

El acelerado desarrollo de las tecnologías de la informática y las comunicaciones (TIC), juega un papel protagonista en muchas de las esferas de la vida cotidiana del hombre. Estas han transformado nuestra manera de trabajar y gestionar los recursos, convirtiéndose en un elemento clave para que nuestra labor sea más productiva, agilizando las comunicaciones y sustentando la colaboración entre los miembros de un determinado grupo de interés. Por lo cual, en la actualidad las TIC se han convertido en un instrumento imprescindible del desarrollo social, político y económico de los países, que ha introducido la necesidad de automatizar los procesos empresariales como punto de partida de la creación de una sociedad relacionada con su entorno.

En los últimos años, en medio de este perfeccionamiento tecnológico ha nacido un nuevo fenómeno de intercambio entre los usuarios de internet, *las redes sociales*, que desde sus inicios ha experimentado una creciente aceptación por los beneficios que ofrece. Estas han logrado integrar en un mismo sistema servicios como la comunicación continua entre personas o empresas, diluyendo las limitaciones de tiempo y espacio que las separan geográficamente. Posibilitando además de un intercambio afectivo, o de interés profesional, poder asentar las bases para la transacción de conocimiento, la colaboración en proyectos y la conexión con el mundo de la ciencia.

En medio de este intercambio de información, de la variedad de criterios y experiencias por parte de diferentes expertos, se ha hecho necesario tomar decisiones democráticas a partir de todo lo expuesto por cada uno de los miembros de un grupo cualquiera.

Para ello, los modelos causales son instrumentos empleados frecuentemente para comprender los sistemas complejos [1-3]. El razonamiento causal es útil en la toma de decisiones por dos razones fundamentales: primero, es natural y fácil de entender; segundo, es convincente debido a que explica el por qué se llega a una conclusión particular. En la toma de decisiones bajo incertidumbre [4], los modelos causales pueden ser empleados para realizar razonamiento evidencial [5, 6].

Una de las alternativas para el modelado causal son los Mapas Cognitivos Difusos (MCD). Estos proveen esquemas más realistas para la representación del



conocimiento, brindando la posibilidad de representar ciclos y modelar la vaguedad [7].

Los MCD han sido empleados en un creciente número de dominios [8-10]. Entre sus áreas de aplicación se reporta su empleo en la toma de decisiones [11-13] y la construcción de sistemas de soporte a la toma de decisiones [14-16]. La representación de conocimiento en forma de mapa permite sintetizar la información, seleccionando los principales conceptos que están vinculados en un problema. Además con esta técnica es posible estudiar la dinámica que hace posible pasar de un estado a otro en la situación que se analiza [17].

Por otra parte la utilización del modelo “Fusión de Mapas Cognitivos Difusos” (FCM – Fusion), el cual presenta la integración de distintas actividades relacionadas con la toma de decisiones apoyadas en MCD en un marco de trabajo único [18], incluyendo: métodos de obtención, análisis estático, aprendizaje, análisis de escenarios y orientado a metas y basado en el uso de operadores de agregación para evaluar las distintas alternativas, el cual permite elevar la aplicabilidad y fiabilidad de los MCD en la toma de decisiones.

En la actualidad esta tarea puede llegar a ser compleja, ya que las herramientas existentes según la bibliografía consultada, presentan la siguiente **situación problemática**:

1. Están orientadas a dominios específicos, como la medicina, la gestión de proyectos y la agricultura de precisión [44].
2. Aunque existen otras orientadas a entornos genéricos, ninguna de estas ha aprovechado las ventajas del uso de las tecnologías web, lo que trae como consecuencia que no se puedan utilizar los ambientes multiusuarios para la toma de decisiones en grupo.
3. Las que están orientadas a entornos genéricos que utilizan tecnología web no integran en un mismo sistema:
 - a. Métodos de gestión de MCD.
 - b. Más de un operador de agregación para evaluar diferentes alternativas.
 - c. Análisis estático.
 - d. Análisis dinámico de escenarios.
4. Los usuarios al desarrollar determinado proyecto no conocen quienes podrían ser expertos para colaborar con ellos, ni tiene la posibilidad de



formar equipos multidisciplinarios que den visiones especializadas desde su área de conocimiento.

5. Presentan una gestión rígida y centralizada, las principales funcionalidades son controladas por el administrador del sistema o por el autor del proyecto.
6. No reutilizan el conocimiento o las experiencias de otros expertos, trayendo consigo que haya que emplear mucho más tiempo y recursos en el estudio de la materia en cuestión y no siempre se obtienen las mejores alternativas, restándole efectividad al proceso.

Considerando la situación descrita, se ha definido el siguiente **diseño teórico de la investigación**.

- **Problema de investigación:** ¿Cómo facilitar la colaboración entre usuarios para tomar decisiones efectivas en grupo?
- **Objeto de estudio:** Sistemas informáticos para la toma de decisiones.
- **Objetivo general de la investigación:** Desarrollar una aplicación web del tipo red social colaborativa, para la toma de decisiones en grupo basado en mapas cognitivos difusos, utilizando el modelo FCM – Fusión.
- **Campo de acción:** Aplicaciones web del tipo Red Social, para la toma de decisiones en grupo basado en MCD que utilizan FCM – Fusion.
- **Idea a defender:** Se plantea que con el desarrollo de una Red Social para la toma de decisiones en grupo usando MCD se pueden hacer efectivas dichas decisiones.

Objetivos específicos:

1. Elaborar el Marco Teórico de la investigación a partir del estado del arte existente actualmente sobre el tema.
2. Realizar el análisis y diseño de la aplicación.
3. Implementar la herramienta informática para la toma de decisiones en grupo basado en mapas cognitivos difusos, utilizando el modelo FCM – Fusion.
4. Validar la herramienta propuesta a través de los métodos definidos en la investigación.



En la investigación se destaca la utilización de los siguientes **métodos teóricos**:

- **Análítico – sintético:** Empleado en el momento de buscar la esencia del tema en cuestión, también permite realizar análisis de teorías, de documentos, entre otros; realizando una síntesis de la misma, posibilitando así encontrar los elementos más importantes que se relacionan con el proceso de toma de decisiones.
- **Análisis histórico – lógico:** Utilizado para conocer con mayor profundidad cómo ha evolucionado el proceso de toma de decisiones usando MCD, estudiando la lógica de cómo se realizaba anteriormente y en la actualidad, con el objetivo de detectar las deficiencias y aspectos positivos a tener en cuenta en la solución que se propone y cumplir con las exigencias del cliente.
- **Inductivo – deductivo:** El cual permite reflejar los elementos comunes entre los fenómenos estudiados, establecer generalizaciones y analizar cada uno de los detalles hasta establecer las relaciones existentes entre los fenómenos estudiados.
- **Modelación:** Utilizado para representar cómo ocurren los procesos que se desean automatizar, además para elaborar los diferentes modelos definidos en la metodología escogida y que sirven de guía durante el desarrollo de la solución.

Los **métodos empíricos** utilizados para obtener información sobre el objeto de estudio fueron:

- **Consulta bibliográfica:** Permite la elaboración del marco teórico de la investigación fundamentada por la información consultada.
- **Consulta de especialistas:** Permite orientar el trabajo teórico – metodológico en la realización de la investigación.
- **Observación científica:** Utilizado con el objetivo de conocer cómo se comportaba la utilización de los mapas cognitivos difusos a la hora de tomar decisiones en grupo en los diferentes dominios de aplicación.
- **Estudio de casos:** Utilizado con el objetivo de valorar el resultado científico de las investigaciones, validar los beneficios y la aplicabilidad de la propuesta.
- **Pruebas de validación:** Confirma la veracidad y utilidad de la propuesta elaborada.



El trabajo de diploma se estructura de la siguiente forma:

Capítulo 1: Fundamentación teórica de la investigación: En este capítulo se hace referencia a los conceptos y definiciones de la investigación. Además de llevar a cabo el análisis de las herramientas y tecnologías a usar, así como la definición de la metodología de desarrollo de software.

Capítulo 2: Red Social NEXO para la toma de decisiones en grupo basado en MCD utilizando FCM – Fusion: En dicho capítulo además de una descripción de la solución propuesta, se detalla el análisis y diseño de la misma, definiendo los requerimientos identificados con el cliente. Además se especifican los distintos patrones de diseño y el patrón arquitectónico a utilizar.

Capítulo 3: Implementación y prueba de la Red Social NEXO: El capítulo hace alusión en una primera parte al proceso de implementación de la herramienta, reflejando el cumplimiento de los requerimientos funcionales propuestos, para luego describir el proceso de prueba de cada una de las funcionalidades implementadas a partir del diseño de casos de pruebas.

Por último, se exponen las **Conclusiones** y **Recomendaciones** derivadas de la investigación, las **Referencias bibliográficas**, así como los **Anexos** que apoyan la comprensión y dan información adicional sobre el trabajo realizado.



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN.

Introducción:

En el presente capítulo se precisan un conjunto de elementos que conforman la fundamentación teórica de la investigación. Se abordan los principales conceptos y definiciones de los modelos mentales, los mapas cognitivos difusos, la toma de decisiones en grupo, los operadores de agregación y el análisis estático y dinámico de escenario. Además de describir las principales características del modelo utilizado, las herramientas existentes respecto a la temática analizada, así como lenguajes, tecnologías y metodología de desarrollo utilizadas en el desarrollo de la propuesta de solución.

1.1 Modelos causales

La causalidad es un tipo de relación entre dos entidades, causa y efecto. Es un proceso directo cuando A causa B y B es el efecto directo de A, o indirecto cuando A causa C a través de B y C es un efecto indirecto de A [19]. A pesar de la dificultad de desarrollar una definición de la causalidad los humanos poseen una comprensión de esta que permite elaborar modelos mentales de la interacción entre los fenómenos existentes a su alrededor [3]. En las ciencias de la computación existen esfuerzos para hacer que la computadora tenga la capacidad de aprendizaje y comprensión de los fenómenos causales, innata en los niños pequeños [20].

En este trabajo se toma la definición sistémica de causalidad. Definida como: las interacciones que involucran un conjunto de elementos (las entradas o causas) afectando el estado futuro de otros (las salidas o efectos) [3].

Los modelos causales constituyen instrumentos prácticos que se emplean frecuentemente para comprender los sistemas complejos [21]. A partir de los modelos causales se pueden establecer las causas de los eventos y predecir sus efectos. La representación y análisis del conocimiento causal facilita el proceso de toma de decisiones [22].

Cada modelo causal se puede representar por un grafo dirigido, denominado grafo causal [23].



La causalidad se ve generalmente como una relación precisa: la misma causa provoca siempre el mismo efecto. Sin embargo en el mundo cotidiano los enlaces entre causa y efecto son frecuentemente imprecisos o imperfectos por naturaleza [24]. Este tipo de causalidad, denominada causalidad imperfecta, desempeña un papel importante en la toma de decisiones [23].

1.2 Mapas cognitivos y mapas cognitivos difusos

Los Mapas Cognitivos (MC) representan el conocimiento causal sin indicar el grado de fortaleza de la relación causa-efecto [2]. En muchos problemas prácticos es necesario representar el grado de influencia causal entre conceptos. La lógica difusa resulta entonces una alternativa para representar la causalidad [18].

La teoría de los conjuntos difusos o borrosos fue introducida por Zadeh [25] en el año 1965. La teoría de los conjuntos difusos o borrosos parte de la teoría clásica de conjuntos, añadiendo una función de pertenencia [26].

La lógica difusa ofrece un marco adecuado para tratar con la causalidad imperfecta. Para expresar el grado de causalidad entre conceptos se pueden emplear expresiones lingüísticas como "negativamente fuerte", "positivamente fuerte", "negativamente débil", "positivamente débil", etc. [27].

Los Mapas Cognitivos Difusos (MCD) son una técnica introducida por Kosko [28] como una extensión de los MC [29]. Mejoran los MC describiendo la fortaleza de la relación mediante el empleo de valores borrosos en el intervalo $[-1,1]$. Los nodos son conceptos causales y pueden modelar eventos, acciones, valores, metas o procesos [12]. Constituyen una estructura de grafo difuso con retroalimentación para representar causalidad [30]. Ofrecen un marco de trabajo más potente y flexible para representar el conocimiento humano y para el razonamiento de los sistemas expertos tradicionales que implementan explícitamente reglas de tipo "SI/ENTONCES" [31].

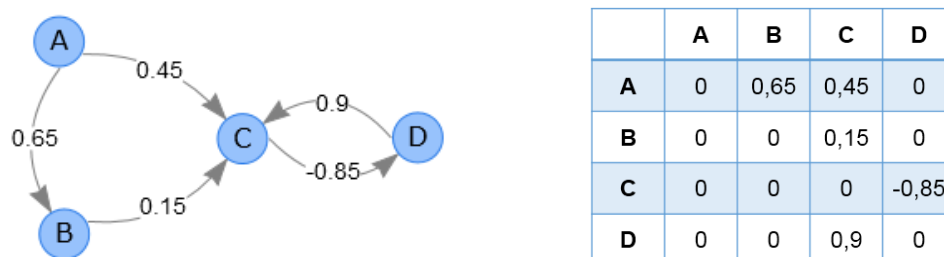


Figura 1: Mapa cognitivo difuso y su correspondiente matriz de adyacencia.



Un MCD se puede representar a través de un grafo dirigido ponderado en el cual los nodos representan conceptos y los arcos indican relación causal [32]. La matriz de adyacencia se obtiene a partir de los valores asignados a los arcos.

Existen tres enfoques fundamentales para el aprendizaje en MCD [33]: la obtención de la matriz de pesos a partir de datos históricos, la adaptación de la relación causa-efecto sobre la base de la intervención de los expertos, y la obtención de los pesos de la matriz combinando la opinión de los expertos y los datos. Los principales métodos para el aprendizaje se pueden clasificar en: aprendizaje hebbiano [34], basados en poblaciones [35] y enfoques mixtos que combinan los métodos anteriores [36].

1.3 Toma de decisiones

La toma de decisiones hace referencia al conjunto de operaciones que comprenden desde el momento en que se detecta una situación que hace necesaria la toma de decisiones hasta que esta es adoptada y ejecutada [37]. Herrera y otros [38] presentan un procedimiento tipo para la toma de decisiones, el cual consta de las siguientes actividades:

1. Definir el problema de toma de decisiones.
2. Analizar el problema y la información.
3. Identificar las alternativas de solución.
4. Establecer criterios o expertos por los cuales ellas puedan ser evaluadas.
5. Evaluar alternativas y seleccionar la mejor.

En el proceso de toma de decisiones se parte de establecer lo que se conoce como matriz de decisión, cuya forma general se muestra en la Figura 2. La información existente en esta matriz puede ser precisa, imprecisa, borrosa, lingüística, etc. [39].

$$\begin{array}{c} E_1 \\ \vdots \\ \vdots \\ E_m \end{array} \begin{pmatrix} N_1 & \dots & N_n \\ a_{11} & \dots & a_{1n} \\ \vdots & \dots & \vdots \\ \vdots & \dots & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix}$$

Figura 2: Matriz de decisión.



Los elementos presentes en la matriz de decisión son los siguientes:

N_1, \dots, N_n : los diferentes estados de la naturaleza que se pueden presentar.

E_1, \dots, E_m : las diferentes decisiones o estrategias que puede adoptar el decisor.

a_{mn}, \dots, a_{mn} : los diferentes resultados que surgen de combinar una estrategia con los diferentes estados de la naturaleza. Estos resultados pueden ser exactos o imprecisos según la información disponible para el problema.

La matriz puede ser diseñada también desde la fijación de criterios. Este caso es conocido en la literatura como métodos de decisión multicriterio (MDMC) [40], o desde la utilización de múltiples expertos (decisiones multiexperto) [41].

Los ambientes de decisión por su parte se pueden clasificar de la siguiente forma [39]:

- Certidumbre: la decisión es inmediata y se conoce cuál es la mejor decisión posible.
- Riesgo: no se conoce cuál es la mejor solución posible, pero sí se puede asociar un grado de probabilidad a las posibles situaciones futuras.
- Incertidumbre: se conoce cuáles son los posibles resultados futuros pero se desconoce el grado de probabilidad asociada a ellos.

1.4 Métodos de decisión multicriterio

Los métodos de decisión multicriterio (MDMC) [42], también conocidos como métodos de ayuda a la decisión multicriterio [43], constituyen un conjunto de metodologías que brindan soporte a las situaciones donde criterios en conflicto, metas, objetivos y puntos de vista deben ser tomados en consideración. Estos métodos se componen generalmente de las siguientes actividades: estructuración del problema, modelado de las preferencias, construcción y caracterización de diferentes funciones de agregación de criterios y construcción de herramientas que le brinden soporte a los modelos. En la actualidad la decisión multicriterio se considera un área fundamental de la teoría de la decisión [43].



1.5 Toma de decisiones en grupo

La toma de decisiones en grupo proporciona gran variedad de experiencias y perspectivas al proceso de decisión, lo que no sucede cuando se trata de un único decisor o individuo. Esta forma de tomar decisiones permite una información más completa del análisis que se desee realizar. Los grupos pueden identificar más alternativas debido a que tienen mayor cantidad y diversidad de información, lo que resulta más evidente cuando los miembros del grupo representan diferentes especialidades. Además, la decisión en grupo incrementa la aceptación de la solución.

El proceso de toma de decisiones en grupo es consistente con los ideales democráticos, de ahí que las decisiones tomadas en grupos puedan percibirse y ser más legítimas que las decisiones tomadas por una persona [44].

1.6 Agregación de la información

Los operadores de agregación son un tipo de función matemática empleada para la fusión de la información. Combinan n valores en un dominio D y devuelven un valor en ese mismo dominio [45]. Denominando esas funciones C [45], los operadores de agregación son funciones de forma:

$$C: N^n \rightarrow N$$

Figura 3: Operador de agregación.

Los operadores de agregación presentan múltiples aplicaciones en diversos dominios [46]. En la toma de decisiones su papel fundamental está en la evaluación y en la construcción de alternativas [45]. Su empleo se enmarca fundamentalmente en la toma de decisiones multicriterio.

Algunos de los operadores de agregación más empleados son los siguientes:

1.6.1 Media Aritmética

La media aritmética es el operador de agregación más utilizado, y se obtiene a partir de la suma de todos sus valores dividida entre el número de sumandos.



1.6.2 Media Ponderada

La media ponderada (WA por sus siglas en inglés) es también uno de los operadores más empleados en la literatura, el cual incorpora un factor de peso.

Definición 1: El operador WA de dimensión n tiene asociado un vector de pesos V , donde: $V_i \in [0; 1]$ y $\sum_1^n V_i = 1$

$$WA(a_1, a_2, \dots, a_n) = \sum_{i=1}^n V_i a_i$$

1.6.3 Media Ponderada Ordenada

La media ponderada ordenada (OWA, siglas del inglés Ordered Weighted Averaging): Este método unifica los criterios clásicos de decisión con incertidumbre en un solo modelo. Es decir, esta unificación abarca los criterios optimista, el pesimista, el de Laplace y el de Hurwicz en una sola expresión [47].

Definición 2: El operador OWA de dimensión n tiene asociado un vector de pesos W , donde: $W_j \in [0; 1]$ y $\sum_1^n W_j = 1$

$$OWA(a_1, a_2, \dots, a_n) = \sum_{j=1}^n W_j b_j$$

Donde b_j es el j – ésimo más grande de los a_j .

1.6.4 Integrales Difusas

Parten de la obtención de medidas difusas (μ) [48]. Las integrales difusas más empleadas son la integral de Sugeno y la integral de Choquet [49].

Cada una de estas familias de operadores presenta características que les permiten modelar determinadas situaciones. La WA posibilita asignar peso a las fuentes de información lo que permite su empleo para representar fiabilidad o importancia/preferencia. Por su parte el operador OWA posibilita la compensación o dar peso a los datos en dependencia de sus valores. Las integrales difusas permiten modelar redundancia, complementariedad e interacciones entre criterios. Sin embargo estas últimas presentan la limitante de que sus coeficientes crecen exponencialmente con el número de criterios a ser agregados [50, 51], lo que dificulta su aplicabilidad.



1.7 Análisis de escenarios

La construcción de escenarios es una de las herramientas más utilizadas en el análisis político prospectivo. Los escenarios suelen ser utilizados con dos objetivos interrelacionados. En primer lugar, para someter a prueba diferentes estrategias en el marco de escenarios de futuro posibles, debiendo estos diferentes “futuros” ser contruidos en base a evidencias robustas y a un análisis de los actores, procesos e incertidumbres a los cuales están expuestos. En segundo lugar, la “reflexión sobre el futuro” mediante la construcción de escenarios suele ser también un medio o instrumento para construir interpretaciones, maneras de entender y opciones que puedan ser compartidas por el grupo de actores involucrados en tal proceso. El resultado final de los escenarios no es lograr proyecciones precisas de lo que pasará en el futuro, sino sugerencias [44].

1.8 Modelo FCM – Fusion

FCM – Fusion presenta como principios y características la integración de distintas actividades relacionadas con la toma de decisiones apoyadas en MCD en un marco de trabajo único. El modelo se orienta al incremento de la aplicabilidad y fiabilidad de la toma de decisiones empleando MCD. El enfoque sistémico está presente en FCM-Fusion con el objetivo de obtener resultados sinérgicos a partir de la interacción e integración de sus componentes [18].

Las premisas fundamentales del modelo están relacionadas con la capacidad del centro decisor para obtener modelos causales en forma de MCD y la existencia del nivel de madurez básico requerido para implantar un proceso de toma de decisiones. La capacidad de modelar las preferencias mediante operadores de agregación constituye otra premisa del modelo [18].

FCM – Fusion integra elementos relacionados con los MCD que generalmente se ven de forma aislada. Estos se pudieran resumir en:

- a) Agregación de MCD de múltiples expertos en un solo modelo.
- b) Determinación de los conceptos más importantes del modelo.
- c) Aprendizaje a partir de entradas y salidas conocidas.
- d) Análisis orientado a metas en MCD.



e) Ordenamiento de alternativas.

1.9 Razonamiento basado en casos

El Razonamiento Basado en Casos (RCB) es el proceso de solucionar nuevos problemas basándose en las soluciones de problemas anteriores. Un mecánico de automóviles que repara un motor porque recordó que otro auto presentaba los mismos síntomas está usando razonamiento basado en casos. Un abogado que apela a precedentes legales para defender alguna causa está usando razonamiento basado en casos. También un ingeniero cuando copia elementos de la naturaleza, está tratando a esta como una “base de datos de soluciones”. El Razonamiento basado en casos es una manera de razonar haciendo analogías. Se ha argumentado que el razonamiento basado en casos no sólo es un método poderoso para el razonamiento de computadoras, sino que es usado por las personas para solucionar problemas cotidianos. Más radicalmente se ha sostenido que todo razonamiento es basado en casos porque está basado en la experiencia previa [79].

El razonamiento basado en casos cuenta con las siguientes características:

- La búsqueda de la solución a un problema no se inicia a partir de los datos o el objetivo, por lo que el camino se acorta considerablemente.
- El razonamiento incluye un algoritmo para determinar la medida de semejanza entre dos objetos.
- Memoria indexada.
- Memoria dinámica.

1.10 Sistemas basados en casos

Los Sistemas Basados en Casos (SBC) basan su funcionamiento en experiencias anteriormente vividas, ya sea por el propio sistema o bien por la persona experta, y a partir de este conocimiento de vivencias realizar una asociación con estas experiencias para extraer una solución de esto. Este sistema de los que se basan en RBC es el que más se asemeja al modo de pensar que tenemos los seres humanos. Si nos ponemos a pensar en los niños pequeños, el tipo de razonamiento que utilizan es el basado en las experiencias que le proporciona su trato con el medio [79].

Para desarrollar un sistema basado en casos (SBC) se deben de definir los siguientes módulos:



- Módulo recuperador de casos.
- Módulo de adaptación.
- Módulo de evaluación de soluciones.
- Módulo de almacenamiento.

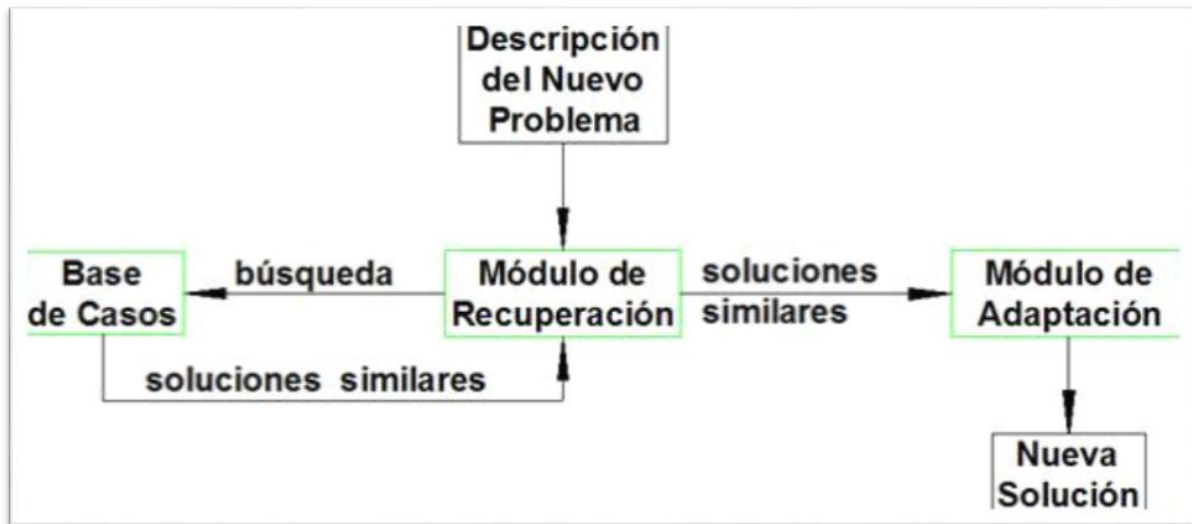


Figura 4: Funcionamiento de un SBC.

1.11 Herramientas web existentes de ayuda a la toma de decisiones basadas en MCD

En la literatura consultada se reportan las siguientes herramientas web de ayuda a la toma de decisiones basadas en el Modelo FCM – Fusion.

1.11.1 ESCEANALY

Herramienta web, que permite la simulación de escenarios previamente establecidos a través de un análisis dinámico. Esta no parte para el análisis de MCD o matrices de decisión, sino que el usuario es el encargado de introducir los escenarios para su posterior simulación. Aunque hace persistentes los resultados de la simulación, no pueden ser compartidos o reutilizados por otros usuarios.

1.11.2 CAUSALITY TOOL

Herramienta web, que permite la construcción de MCD a partir de la gestión de conceptos y etiquetas, luego de creados, el usuario puede obtenerlo en forma de matriz o grafo dirigido y ponderado con los valores de dichas etiquetas. El sistema también permite obtener la agregación de los MCD – aunque solo el administrador puede hacerlo – mediante el operador *Media aritmética*.



Es importante destacar que estas herramientas no logran ni por si solas ni complementándose, el enfoque sistémico que propone el modelo FCM – Fusion, por lo cual no se logra la sinergia en el proceso de integración de sus componentes.

Entre otras de las características que presentan las herramientas antes mencionada, se puede mencionar que los grupos de trabajo son gestionados y conformados por un administrador, lo cual imposibilita que los usuarios puedan compartir de una forma directa experiencias o conocimientos obtenidos con el uso de las mismas, además de no tener la posibilidad de conformar equipos multidisciplinarios que den perspectivas particulares desde sus áreas de investigación y en muchas ocasiones se hace casi imposible contar con los expertos mejor preparados en dicha área, restándole factibilidad y eficiencia al modelo utilizado lo cual hace sumamente complicado el proceso de construir alternativas efectivas que ayuden a la elección de una decisión correcta.

1.12 Redes sociales

Las redes sociales han cambiado y revolucionado las comunicaciones personales en la Red. Si hace un par de años la comunicación por Internet se limitaba a la mensajería instantánea Messenger o Chat online ahora prolongan su vida social, afectiva, profesional y escolar a plataformas en red [52].

Las redes sociales nacieron hace unos años con el fin de establecer redes de contactos en el entorno profesional. Con el paso del tiempo, han adquirido la dimensión de "punto de encuentro" para usuarios que comparten aficiones, gustos o intereses. Existen cientos de redes sociales con perfiles de usuarios diferentes y temáticas variadas, profesionales, de ocio, de consumo. Estas plataformas han conseguido situarse a la cabeza del ranking de páginas con más accesos y volumen de tráfico de información [52].

Los usuarios de las redes sociales profesionales no solo proporcionan datos de contacto, de formación, experiencia laboral. Si no también datos vinculados a su vida profesional [52].

Las Redes Sociales se han convertido en el mayor canal de comunicación existente, por lo cual se convierte en una herramienta poderosa para la colaboración entre profesionales distanciados geográficamente.



1.13 Metodología de desarrollo de software

Una metodología es el conjunto de fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de un sistema [53].

En ingeniería de software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información [54].

El desarrollo de software es un proceso difícil y lleno de riesgos donde la metodología define quién debe hacer qué, cuándo y cómo debe hacerlo. El fracaso en el desarrollo de un sistema está estrechamente relacionado con una mala selección de la misma [55], por lo cual este proceso debe llevar un análisis previo y exhaustivo por parte del equipo de desarrollo. Estas se clasifican en dos grandes grupos:

- **Pesadas/Tradicionales:** Son las más tradicionales, se centran en la definición detallada de los procesos y tareas a realizar, herramientas a utilizar, y requiere una extensa documentación, ya que pretende prever todo de antemano. Este tipo de metodologías son más eficaces y necesarias cuanto mayor es el proyecto que se pretende realizar respecto a tiempo y recursos que son necesarios emplear, donde una gran organización es requerida [56].
- **Ligeras/Ágiles:** Orientadas a la interacción con el cliente y el desarrollo incremental del software, mostrando versiones parcialmente funcionales del producto al cliente en intervalos cortos de tiempo, para que pueda evaluar y sugerir cambios según se va desarrollando [57].

1.13.1 Proceso Racional Unificado (RUP)

Es una metodología de desarrollo de software que intenta integrar los aspectos a tener en cuenta durante todo el ciclo de vida del sistema, con el propósito de abarcar grandes o pequeños proyectos de software. Proporciona un acercamiento a la asignación de tareas y responsabilidades en una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad, que se ajuste a las necesidades de sus usuarios finales. Provee herramientas para los pasos del desarrollo y una vasta documentación de ayuda para sus clientes [58].



RUP como metodología que tiene tres características importantes que la hacen única, está dirigida por casos de uso, centrada en la arquitectura, iterativa e incremental [59]. Otras de sus características más importantes son:

- Propone un flujo de trabajo específicamente para el Análisis y Diseño del sistema, donde se especifican los requisitos identificados y se describen en términos de diseño sobre cómo se va a implementar el sistema.
- Provee vasta documentación al equipo de trabajo.
- Describe como obtener, organizar y documentar todas las funcionalidades y restricciones del sistema.
- En cada una de sus fases obtiene una serie de artefactos que sirven para comprender mejor tanto el Análisis como el Diseño del sistema.

Ventajas:

- Coste del riesgo a un solo incremento.
- Reduce el riesgo de no sacar el producto en el calendario previsto.
- Acelera el ritmo de desarrollo.
- Se adapta mejor a las necesidades del cliente.

Metodología de desarrollo de software que está basada en componentes e interfaces bien definidas, y junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

1.13.2 Programación Extrema (XP)

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. XP se basa en realimentación 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. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy



cambiantes, y donde existe un alto riesgo técnico [60]. Donde lo fundamental en este tipo de metodología es:

- La comunicación, entre los usuarios y los desarrolladores.
- La simplicidad, al desarrollar y codificar los módulos del sistema.
- La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales [61].

El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega, Iteraciones, Producción, Mantenimiento y Muerte del Proyecto [61].

Fase I: Exploración: En esta fase, los clientes plantean a grandes rasgos las Historias de Usuario (HU) que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología [61].

Fase II: Planificación de la Entrega: En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días [61].

Fase III: Iteraciones: Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. Al final de la última iteración el sistema estará listo para entrar en producción. Los elementos que deben tomarse en cuenta durante la elaboración del Plan de la Iteración son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior [61].

Fase IV: Producción: La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del



cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase [61].

Fase V: Mantenimiento: Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura [61].

Fase VI: Muerte del Proyecto: Es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo [61].

1.13.3 Selección de la metodología de desarrollo

Luego de analizar las características de las metodologías mencionadas anteriormente, se selecciona XP para que guíe el proceso de desarrollo, debido a que:

- Es la que mejor se ajusta a las necesidades del proyecto en cuanto a recursos técnicos, humanos y tiempo de desarrollo.
- El equipo de trabajo es pequeño y el cliente forma parte de mismo, lo cual mitiga los riesgos y aumenta la probabilidad de éxito.
- Es adecuada para proyectos con requisitos imprecisos y cambiantes.
- Permite que en cualquier momento del proceso de desarrollo de la aplicación, se pueda regresar a operaciones anteriores sin afectar el ciclo de vida del software.

1.14 Herramientas de desarrollo

Una herramienta de desarrollo de software es un programa informático que usa un programador para crear, depurar, gestionar o mantener un programa [62].



1.14.1 Lenguajes de programación de lado del cliente

HTML5:

HTML5 (las siglas de HTML significan Hyper Text Markup Language) es un lenguaje usado para estructurar y presentar el contenido para la web. Es uno de los aspectos fundamentales para el funcionamiento de los sitios, pero no es el primero. Es de hecho es la quinta revisión del estándar que fue creado en 1990. A fines del año pasado, la W3C la recomendó para transformarse en el estándar a ser usado en el desarrollo de proyectos venideros. Por así decirlo, qué es HTML5 está relacionado también con la entrada en decadencia del viejo estándar HTML4, que se combinaba con otros lenguajes para producir los sitios que podemos ver hoy en día. Con HTML5, tenemos otras posibilidades para explotar usando menos recursos. Con HTML5, también entra en desuso el formato XHTML, dado que ya no sería necesaria su implementación [63].

JavaScript:

JavaScript (JS) es un lenguaje de programación interpretado. Se define como orientado a objetos [64], basado en prototipos, imperativo, versátil, potente y dinámico. Se utiliza principalmente en su forma del lado del cliente – de esta forma se evita la sobre carga del lado del servidor – implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas [65].

En la actualidad todos los navegadores son capaces de interpretar JS, que para interactuar con una página web se provee de una implementación del Document Object Model (DOM).

CSS3:

CSS es un lenguaje de hojas de estilos creado para controlar la presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para la creación de páginas web complejas [66].

La separación de los contenidos y su presentación presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados “documentos semánticos”). Además, mejora la



accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes [66].

1.14.2 Lenguaje de programación de lado del servidor

PHP5:

Hypertext Preprocessor (PHP) es un lenguaje interpretado de alto nivel embebido ejecutado en el servidor, originalmente diseñado para el desarrollo web de contenido dinámico. PHP fue uno de los primeros lenguajes de programación de su tipo y es uno de los más extendidos de la web. Permite realizar determinadas acciones de una forma fácil y eficaz sin tener que implementados en otro lenguaje distinto al HTML. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

Se considera uno de los lenguajes más flexibles, potentes y de alto rendimiento conocidos hasta el día de hoy. Lo que ha atraído el interés de múltiples sitios con gran demanda de tráfico como Facebook, para optar por PHP como tecnología de servidor.

Este lenguaje forma parte del software libre publicado bajo la licencia PHP, que es incompatible con la Licencia Pública General de GNU debido a las restricciones del uso del término PHP.

Una de las características más potentes y destacables de PHP es su soporte para una gran cantidad de bases de datos, como PostgreSQL, MySQL, Oracle, Solid, entre otros [67].

1.14.3 Frameworks de desarrollo

Un framework simplifica el desarrollo de una aplicación a través de la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, un framework proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último un framework facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas [68].



Bootstrap:

Es un framework o conjunto de herramientas de software libre para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales. Es compatible con la mayoría de los navegadores web. La información básica de compatibilidad de sitios web o aplicaciones está disponible para todos los dispositivos y navegadores. Además es de código abierto y está disponible en GitHub [69]. Los desarrolladores están motivados a participar en el proyecto y a hacer sus propias contribuciones a la plataforma.

jQuery:

jQuery es una biblioteca JavaScript, rápida, pequeña y rica en funciones. Que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Con una combinación de versatilidad y capacidad de ampliación, jQuery ha cambiado la forma en que millones de personas escriben JavaScript, siendo hoy la más utilizada.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados [70].

Yii:

Es un framework orientado a objetos, software libre, de alto rendimiento basado en componentes, PHP y para aplicaciones web, con características [71]:

- Patrón de diseño Modelo Vista Controlador (MVC).
- Integración con jQuery.
- Entradas de Formulario y validación.
- Widgets de Ajax, como autocompletado de campos de texto y demás.
- Soporte de Autenticación incorporado.
- Personalización de aspectos y temas.



- El manejo de errores y logging.
- Las medidas de seguridad incluyen la prevención cross-site scripting (XSS).
- Generación automática de código para el esqueleto de la aplicación, aplicaciones CRUD, etc.

1.14.4 Herramienta CASE

Visual Paradigm v8.0:

Es una herramienta CASE que utiliza UML como lenguaje de modelado. Es libre y multiplataforma y se caracteriza principalmente por su robustez, usabilidad y portabilidad. Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Se integra fácilmente con varios IDEs¹¹ y permite la generación semiautomática de código a partir de los diagramas construidos. Soporta aplicaciones web y existe compatibilidad entre sus versiones. Acelera el desarrollo de software obteniendo productos de calidad reduciendo costos y riesgos.

Esta herramienta permite realizar ingeniería tanto directa como inversa, a partir de un modelo relacional en SQL Server, MySQL, PostgreSQL, es capaz de desplegar todas las clases asociadas a las tablas (siguiendo el patrón de diseño Una Clase-Una Tabla). Soporta múltiples usuarios trabajando sobre el mismo proyecto, permite control de versiones y genera la documentación automáticamente en formatos como web o .pdf.

1.14.5 Entorno de desarrollo integrado:

Un entorno de desarrollo integrado, llamado también IDE (sigla en inglés de integrated development environment), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios.

NetBeans v8.0:

NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE [72] es un producto libre y gratuito sin restricciones de uso.



NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos. Algunas de las características de la aplicación son:

- Gestión de la interfaz de usuario (menús y barras de herramientas).
- Gestión de configuración de usuario.
- Gestión de almacenamiento (guardar o cargar algún tipo de dato).
- Gestión de ventana.
- Librería visual de Netbeans.

1.14.6 Sistema gestor de base de datos

Un SGBD o Sistema Gestor de Base de Datos (DBMS, siglas del inglés Data Base Management System) es una colección de datos relacionados entre sí, estructurados y organizados, y un conjunto de programas que acceden y gestionan esos datos. La colección de esos datos se denomina Base de Datos (BD). Los sistemas gestores de base de datos están diseñados para gestionar grandes bloques de información, que implica tanto la definición de estructuras para el almacenamiento como de mecanismos para la gestión de la información. El SGBD es una aplicación que permite a los usuarios definir, crear y mantener la DB y proporciona un acceso controlado a la misma [44].

PostgreSQL v9.4:

PostgreSQL es un Sistema de gestión de bases de datos relacional orientado a objetos y libre, publicado bajo la licencia BSD.

Es una alternativa a otros sistemas de bases de datos de código abierto como MySQL y propietarios como Oracle. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema y entre sus principales características aparecen:

- Ofrece funcionalidades de estándar SQL92 y SQL99.
- Sistema multiplataforma (Windows, Linux y Unix).



- Incluye subconsultas, valores por defecto, disparadores, vistas
- Permite la gestión de diferentes usuarios con permisos asociados.
- Claves ajenas también denominadas llaves foráneas.
- Incluye características de la programación orientada a objetos.
- Capacidad de albergar programas en el servidor en varios lenguajes.
- Múltiples métodos de autenticación.

Conclusiones parciales

- Las herramientas existentes no integran los principios y características necesarios para llevar a cabo las actividades relacionadas con las toma de decisiones, necesarias para la construcción de alternativas eficientes. Además de no cumplir con el carácter sistémico que presenta el modelo FCM – Fusion.
- El desarrollo de la propuesta de solución requiere el uso de un enfoque de desarrollo ágil, en este caso XP es la metodología que se ha adoptado, dada su flexibilidad ante el cambio de requisitos y el manejo de los riesgos técnicos. Además XP incluye la ejecución de buenas prácticas de ingeniería tales como: el desarrollo guiado por pruebas, la programación en parejas y la refactorización.



CAPÍTULO 2: RED SOCIAL NEXO PARA LA TOMA DE DECISIONES EN GRUPO BASADO EN MCD UTILIZANDO FCM – FUSION.

Introducción:

En este capítulo además de describir la solución propuesta, se abordan las fases de Exploración, Planificación e Iteraciones – que incluye el diseño del sistema –, todas estas etapas propias de la metodología de desarrollo utilizada. Se describen los artefactos obtenidos según la misma, se realiza el modelo de datos y se especifican los requisitos funcionales y no funcionales con los que contara el sistema.

2.1 Red Social NEXO

NEXO tiene como principal objetivo agrupar opiniones y preferencias de un grupo de usuarios en temas concretos, para de esta manera construir un conocimiento colectivo que sirva como base de análisis para la toma de decisiones en grupo.

Dicho sistema unifica en un entorno de red social colaborativa, además de las funcionalidades propias de este tipo de aplicaciones tales como sugerencias de proyectos y de usuarios afines en áreas de conocimiento, las potencialidades de los MCD para la toma de decisiones, basándose en el sistema actividades que propone el modelo FCM – Fusion. Se logra en la misma un tratamiento tanto desde la perspectiva estructural de los MCD, dígase su inserción, agregación, visualización de las matrices de adyacencia y análisis estáticos, como facilidades para el análisis dinámico de escenarios.

NEXO incluye además un SBC, el cual se encarga de recomendar a los usuarios quienes podrían ser colaboradores expertos para trabajar en algún proyecto en específico. Este SBC se basa en experiencias propias de cada usuario, a la hora de seleccionar sus colaboradores, con la posibilidad de aprender y mantenerse en constante actualización con el fin de lograr la eficiencia en cada una de las recomendaciones que realiza.

2.2 Exploración

La metodología de desarrollo XP comienza con la fase de exploración. Durante esta etapa se realiza el proceso de identificación de las Historias de Usuario (HU), estas



constituyen uno de los artefactos más importante que se generan en la metodología, pues son la forma en que se especifican los requisitos del sistema.

2.2.1 Historias de Usuario

Las Historias de Usuario (HU) son la técnica utilizada para representar y especificar los requisitos del software. Se trata de un conjunto de tablas en las cuales el cliente describe brevemente las características que el sistema debe poseer. Su tamaño no excede de unas pocas líneas de texto. El tratamiento de las HU es muy dinámico y flexible. Tienen el mismo propósito que los casos de uso, las escriben los propios clientes tal y como ven ellos las necesidades del sistema. Las historias de usuario son descompuestas en tareas de programación y asignadas a los programadores para ser implementadas durante una iteración [73].

En el presente trabajo de se obtuvieron un total de 39 HU, cada una de ellas respondiendo a las diferentes funcionalidades solicitadas por el cliente, las cuales se realizarán en 4 iteraciones. Teniendo en cuenta la dimensión del sistema propuesto, se realiza una selección de las HU de mayor importancia. A continuación se describen 3 de estas, otras 3 se encuentran en el Anexo 1.

Historia de Usuario	
Número: 17	Nombre de Historia de Usuario: Mostrar grafo MCD
Modificación de Historia de Usuario Número:	
Usuario: Usuario_Registrado	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 6
Riesgo en Desarrollo: Alto	Puntos Reales: 7
Descripción: Inicia cuando el usuario decide visualizar el grafo del MCD de un proyecto determinado. Este grafo es insertado en la página donde se encuentra.	
Observaciones:	
<ul style="list-style-type: none">• Si el usuario no tiene el mapa creado, el sistema muestra un mensaje comunicándoselo.	



- Si el mapa no tiene vértices, se muestra un mapa, vacío.

Prototipo de Interfaz:

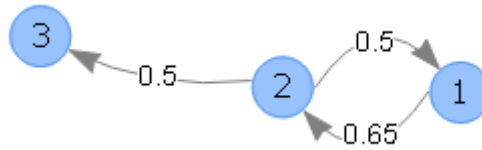


Tabla 1: HU17 Mostrar grafo MCD.

Historia de Usuario

Número: 14 **Nombre de Historia de Usuario:** Mostrar matriz MCD.

Modificación de Historia de Usuario Número:

Usuario: Usuario_Registrado **Iteración Asignada:** 2

Prioridad en Negocio: Alta **Puntos Estimados:** 5

Riesgo en Desarrollo: Alto **Puntos Reales:** 3

Descripción: Inicia cuando el usuario decide visualizar el matriz del MCD de un proyecto determinado. Dicha matriz se muestra en la página donde se encuentra.

Observaciones:

- Si el usuario no tiene el mapa creado, el sistema muestra un mensaje comunicándoselo.
- Si el mapa no tiene vértices insertados el sistema muestra una matriz con relación “ninguno” entre los conceptos.

Prototipo de Interfaz:

Conceptos	Concepto X	Concepto Y	Concepto Z	Concepto W
Concepto X	Ninguno	Muy_Muy_Fuerte	Medio	Fuerte
Concepto Y	Medio	Ninguno	Fuerte	Bajo
Concepto Z	Muy_Muy_Fuerte	Muy_Bajo	Ninguno	Medio
Concepto W	Medio	Fuerte	Muy_Muy_Fuerte	Ninguno

Tabla 2: HU18 Mostrar matriz MCD.



Historia de Usuario	
Número: 20	Nombre de Historia de Usuario: Obtener análisis dinámico.
Modificación de Historia de Usuario Número: 0	
Usuario: Usuario_Registrado	Iteración Asignada: 3
Prioridad en Negocio: Media	Puntos Estimados: 6
Riesgo en Desarrollo: Medio	Puntos Reales: 8
Descripción: Inicia cuando el usuario decide iniciar la simulación de escenarios, luego de introducido los datos necesarios por el usuarios y dar clic en el botón “Obtener análisis dinámico”, se muestra una gráfica con los escenarios obtenidos en cada iteración.	
Observaciones: <ul style="list-style-type: none">• Si el proyecto que escoge el usuario tiene solo un MCD insertado, el sistema muestra un mensaje comunicándole al usuario que no se puede llevar a cabo el análisis dinámico.• Si el usuario entra de forma incorrecta los escenarios el sistema le muestra un mensaje comunicándole el error.	
Prototipo de Interfaz: <div style="text-align: center;"><input type="text" value="Proyecto1"/> <input type="text" value="0.9;0.2;0.75"/> <input type="text" value="0.88"/> <input type="text" value="5"/> <input type="button" value="GetDinamicAnalisys!"/></div>	

Tabla 3: HU19 Obtener análisis dinámico.

2.3 Planificación

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente [73].



2.3.1 Lista de reservas del producto

La lista de reserva del producto es una tabla que recoge los requisitos funcionales y no funcionales de una aplicación, el tiempo de duración estimado en semanas para el desarrollo de cada funcionalidad y el rol que realiza la estimación. A continuación se muestra la Lista de reservas del producto:

Item *	Descripción	Estimación	Estimado por
Prioridad: Muy Alta.			
1	Registrar usuario.	3	Analista.
2	Modificar dato de usuario.	2	Analista.
3	Autenticar usuario.	2	Analista.
4, 5, 6, 7	Mostrar/ Crear/ Modificar/ Eliminar proyecto.	6	Analista.
8	Insertar concepto.	1	Analista.
9	Seleccionar grupo de etiquetas.	3	Analista.
10, 11	Crear/ Eliminar MCD.	4	Analista.
12	Mostrar información del MCD.	1	Analista.
Prioridad: Alta.			
13	Mostrar grafo MCD.	6	Analista.
14	Mostrar matriz MCD.	5	Analista.
15, 16	Adicionar/ Eliminar vértice.	2	Analista.
17	Obtener agregación por media aritmética.	2	Analista.
18	Obtener agregación por media ponderada.	4	Analista.
19	Obtener análisis estático.	3	Analista.
Prioridad: Media.			
20	Obtener análisis dinámico.	6	Analista.
21, 22, 23, 24	Crear/ Mostrar/ Modificar/ Eliminar notificación.	4	Analista.



25	Mostrar perfil de usuario.	1	Analista.
26, 27, 28	Mostrar/ Adicionar/ Eliminar área de interés.	3	Analista.
29, 30, 31	Mostrar/ Adicionar/ Eliminar amigos.	3	Analista.
Prioridad: Baja.			
32	Buscar proyectos.	3	Analista.
33	Buscar personas.	3	Analista.
34	Mostrar “Mis Proyectos”	4	Analista.
35	Mostrar “Colaboraciones”	1	Analista.
36, 37, 38	Adicionar/ Mostrar/ Eliminar Colaboradores	3	Analista.
39	Sugerir expertos.	7	Analista.

Tabla 4: Lista de reservas del producto.

Requisitos No Funcionales:

1	Usabilidad: Permitir una navegación sencilla, esto se logrará a partir de una estructura de la información correcta, en todo momento los usuarios tendrán conocimiento del lugar donde se encuentra en el sistema a través del uso de títulos, subtítulos. Los enlaces y botones serán fáciles de asociar con las operaciones que realizan.	Analista.
2	Apariencia o interfaz externa: La interfaz gráfica uniforme incluyendo pantallas, menús y opciones. La consistencia de la interacción entre usuario y componente estará determinada por el diseño de la interfaces de usuario que mantendrán los elementos como menús y zonas de trabajo, en posiciones fijas, además de la mayor uniformidad posible entre cuadros de texto y botones. Tanto los títulos de los componentes de la interfaz, como los mensajes para interactuar con los usuarios, así como los mensajes de error, deberán ser en idioma español y tener una apariencia uniforme, también deberán cumplir con los estándares de interfaz de usuario presentes en el centro de la Universidad.	Analista.
3	Confiabilidad: Ante cualquier evento, ya sea correcto o incorrecto, se mostrará	Analista.



	<p>un mensaje informativo. Frente a la falta de fluido eléctrico el sistema reinicia el proceso que desarrollaba en momentos del incidente.</p>	
4	<p>Portabilidad: Permitir que la aplicación se ejecute en Sistemas Operativos como GNU/Linux y Windows, garantizando un producto multiplataforma.</p>	Analista.
5	<p>Seguridad: Permitir el acceso a la información de acuerdo a los permisos y rol del usuario, teniendo en cuenta en todo momentos los objetivos y principios de la seguridad informática. Los usuarios deben autenticarse antes de interactuar con el sistema.</p> <p>El sistema deberá garantizar el acceso controlado a la información. Este debe influir sobre cómo se presentan las interfaces para cada usuario dependiendo del nivel de acceso a la información.</p>	Analista.
6	<p>Software:</p> <p>Para el cliente:</p> <ul style="list-style-type: none">• Utilizar navegador (Mozilla Firefox o Google Chrome).• Sistema operativo Windows o GNU/Linux. <p>Para el servidor:</p> <ul style="list-style-type: none">• Sistema operativo Windows o GNU/Linux en cualquiera de sus distribuciones.• Servidor de base de datos MySQL.• Servidor con módulo PHP disponible (WAMP, XAMPP, u otros).	Analista.
7	<p>Hardware:</p> <p>Para el servidor:</p> <ul style="list-style-type: none">• Capacidad de disco duro superior a 80 GB. Se requiere un mínimo de 2 GB de RAM y 2.1 GHz de velocidad de procesamiento. <p>Para el cliente:</p> <ul style="list-style-type: none">• Utilizar Capacidad de disco duro superior a 50 Giga bytes. Se requiere un	Analista.



	mínimo de 256 MB de RAM y 1.6 GHz de velocidad de procesamiento.	
8	Eficiencia: Los tiempos de respuesta y velocidad de procesamiento de la información que no sean mayores de 3 segundos para las actualizaciones y 10 para las recuperaciones.	Analista.
9	Restricciones en el diseño y la implementación: El análisis y diseño de la aplicación estará basado en la metodología de desarrollo XP. Definir un estilo de código para la implementación. Utilizar la herramienta CASE Visual Paradigm para la modelación del sistema mediante el lenguaje UML. Desarrollar la aplicación utilizando el marco de trabajo Yii y el IDE NetBeans.	Analista.

Tabla 5: Requisitos no funcionales.

2.3.2 Plan de Iteraciones

Luego de identificar las HU y la estimación del esfuerzo por cada una de ellas, se procede a realizar el plan de iteraciones, en el cual estarán contenidas las HU en el orden a realizar por cada iteración según su orden de relevancia, así como la descripción y el total de semanas que durarán cada una de estas. A continuación se muestra el plan de iteraciones:

Iteración	Descripción	Orden de la HU a implementar	Duración total
1	En esta primera iteración se tiene como principal objetivo implementar las HU que poseen prioridad <i>Muy Alta</i> , las cuales representan funcionalidades sensibles del sistema que inciden críticamente en el funcionamiento del mismo, las cuales son más que indispensables para cumplir con las necesidades del cliente	1 – 12.	19.0
2	En la siguiente iteración nos concentraremos en implementar las HU que tienen prioridad <i>Alta</i> , las cuales no son menos importantes pero no inciden de la críticamente en el funcionamiento del sistema.	13 – 19.	21.0



3	En esta iteración se implementaran las HU que cuentan con una prioridad <i>Media</i> , al igual que las anteriores son de gran importancia en cuanto a su incidencia en el funcionamiento del producto final pero muchas dependen de la implementación de las HU anteriores.	20 – 31	17.0
4	En esta última iteración se implementaran las HU que cuentan con una prioridad <i>Baja</i> , al igual que las anteriores son de gran importancia en cuanto a su incidencia en el funcionamiento del producto final pero otras HU no dependen de ellas para su implementación.	32 – 39	21.0

Tabla 6: Plan de iteraciones.

2.3.3 Plan de Entrega

A partir del plan de iteraciones descritas anteriormente se procede a realizar el plan de entrega, el cual tiene como objetivo dar a conocer las fechas de culminación de las iteraciones y sus correspondientes historias de usuario.

Iteración	1	2	3	4
Cantidad de HU	12	7	11	8
Fecha de Entrega	19-01-2015	13-02-2015	13-03-2015	10-04-2015

Tabla 7: Plan de entregas.

2.4 Iteraciones

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. Al final de la última iteración el sistema estará listo para entrar en producción [73].

Diseño del sistema:

La metodología XP no requiere de la representación del sistema mediante diagramas de clases utilizando UML para diseñar las aplicaciones, en este caso



utiliza otras técnicas como las tarjetas CRC (Cargo o clase, Responsabilidad, Colaboración).

2.4.1 Tarjetas CRC

Las tarjetas CRC permiten que el equipo completo contribuya en la tarea del diseño. Como su nombre lo indica se dividen en tres secciones: el nombre de la clase, sus responsabilidades y sus colaboradores. El nombre de la clase se coloca a modo de título en la tarjeta, las responsabilidades se colocan a la izquierda, y las clases que se implican en cada responsabilidad a la derecha, en la misma línea que su requerimiento correspondiente. Una clase describe cualquier objeto o evento, mediante los atributos y los métodos, las responsabilidades son las tareas que realizan o los métodos correspondientes a la clase y los colaboradores son las demás clase con las que trabaja conjuntamente para cumplir con sus responsabilidades. A continuación se describen 3 de estas, 3 se encuentran en el Anexo 2.

Mapa_Control	
Responsabilidades	Colaboraciones
Crear mapa.	Proyecto_Control.
Eliminar mapa.	Usuario_Control.
Mostrar grafo MCD.	Vértice_Control.
Mostrar matriz MCD.	

Tabla 8: Tarjeta CRC Mapa_Control.

Proyecto_Control	
Responsabilidades	Colaboraciones
Crear proyecto.	Usuario_Control.
Modificar datos del proyecto.	
Eliminar proyecto.	

Tabla 9: Tarjeta CRC Proyecto_Control.



Vértice_Control	
Responsabilidades	Colaboraciones
Insertar vértice.	Mapa_Control.
Eliminar vértice.	Concepto_Control.
	Etiqueta_Control.

Tabla 10: Tarjeta CRC Vertice_Control.

2.5 Arquitectura de software

Según el estándar IEEE (Institute of Electrical and Electronics Engineers): “La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución”.

2.5.1 Patrón Modelo Vista Controlador (MVC)

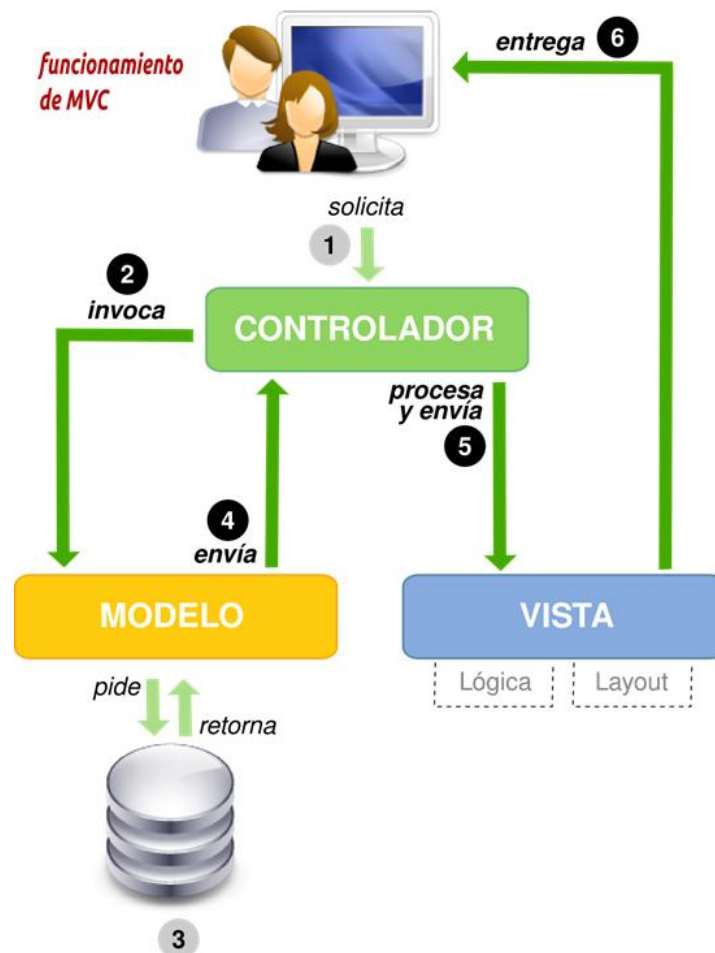


Figura 5: Patrón Modelo Vista Controlador (MVC).



El Modelo es la representación de los datos que maneja la aplicación. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos, es el responsable de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Define reglas de negocio (la funcionalidad del sistema).
- Notificar a las vistas los cambios que en los datos pueda producir un agente externo si se está ante un modelo [74].

La Vista es el elemento que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa preferentemente con el Controlador. Es responsable de:

- Recibir datos procesados por el controlador o del modelo y mostrarlos al usuario.
- Tener un registro de su controlador asociado [74].

El Controlador es el elemento que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo, centra toda la interacción entre la Vista y el Modelo. Maneja y entrega las solicitudes del usuario, procesando la información necesaria y modificando el Modelo en caso de ser necesario. Es responsable de:

- Recibir los eventos de entrada.
- Implementar la lógica del negocio, es decir la funcionalidad de la aplicación.
- Permitir la validación de los datos [74].

2.6 Patrones de diseño

Un patrón de diseño es una solución repetible a un problema recurrente en el diseño de software. Esta solución no es un diseño terminado que puede traducirse directamente a código, sino más bien una descripción sobre cómo resolver el problema, la cual puede ser utilizada en diversas situaciones. Los patrones de diseño reflejan todo el rediseño y recodificación que los desarrolladores han ido haciendo a medida que intentaban conseguir mayor reutilización y flexibilidad en su



software. Los patrones documentan y explican problemas de diseño, y luego discuten una buena solución a dicho problema. Con el tiempo, los patrones comienzan a incorporarse al conocimiento y experiencia colectiva de la industria del software, lo que demuestra que el origen de los mismos radica en la práctica misma más que en la teoría [73].

2.6.1 Patrones GRASP

Los Patrones de Software para la Asignación General de Responsabilidad GRASP (General Responsibility Assignment Software Patterns) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones. En la implementación de la aplicación se utilizaron los siguientes:

Creador: Ayuda a identificar quién debe ser el responsable de la creación o instanciación de nuevos objetos o clases. Tiene la información necesaria para realizar la creación del objeto que es una de las actividades más comunes en un sistema orientado a objetos. También se encarga de almacenar o manejar varias instancias de la clase.

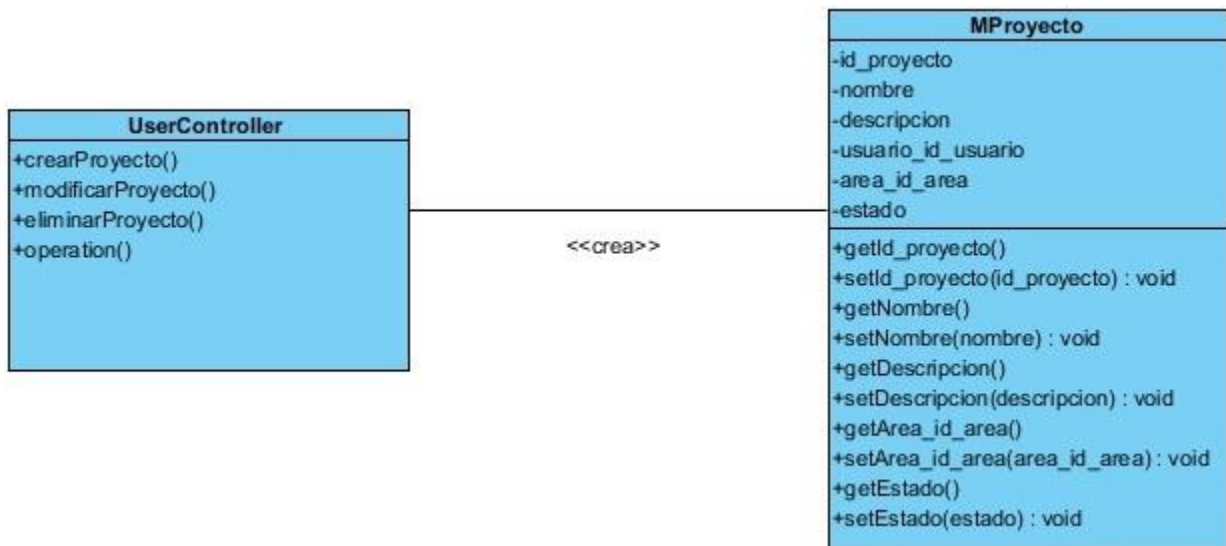


Figura 6: Diagrama de clases que representa el patrón Creador.

Controlador: Es el encargado de asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. La mayor parte de los sistemas reciben eventos de entrada externa, en cualquiera de los casos que puedan existir, si se recurre a un diseño orientado a objetos, hay que elegir los controladores que manejen esos eventos de entrada.



Figura 7: Diagrama de clases que representa el patrón Controlador.

Experto: Es un patrón que se usa más que cualquier otro al asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos. Expresa que siempre se debe asignar una responsabilidad al experto en información, o sea, a la clase que cuenta con la información necesaria para llevar a cabo la funcionalidad [76].

La evidencia del uso de este patrón se puede observar en la mayoría de las clases controladoras del sistema. Un ejemplo de ella podría ser *ProyectoController* que es el encargado de controlar la información respecto a los proyectos y a todos los datos relacionados con los mismos, y de esta forma, optimiza y organiza las responsabilidades sobre la información sensible de los proyectos registrados en el sistema.

2.7 Modelo de datos

El Modelo de Datos (MD) es una definición lógica, independiente y abstracta de objetos, operadores que en conjunto ayudan a diseñar aplicaciones cuyo objetivo es permitir describir y manipular información almacenada en la base de datos. El MD estudia los datos independientemente del procesamiento que los transforma. Se compone de tres piezas de información interrelacionadas: el objeto de datos, los atributos que describen el objeto de datos y la relación que conecta objetos de datos entre sí [75]. En el Anexo 4, se visualiza el MD perteneciente a la solución propuesta.

Conclusiones parciales:

- La definición de las HU nos brinda una guía descriptiva de cómo se comportara la implementación del sistema.
- Los requisitos del sistema y la arquitectura propuesta permitieron obtener un diseño robusto de la aplicación.



- Las definiciones de las Tarjetas CRC aportaron en la identificación de las responsabilidades de las clases y las colaboraciones de las mismas.
- Se utilizaron patrones en el diseño que optimizaran y harán más flexibles la implementación de la aplicación, así como la seguridad en el acceso a datos de los datos.



CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA DE LA RED SOCIAL NEXO

Introducción:

Este capítulo estará dedicado a la implementación del sistema referente a la fase de *Iteraciones* y a la realización de pruebas realizadas perteneciente a la fase de *Producción*. Por otra parte se identifican y describen todas y cada una de las tareas de ingenierías llevadas a cabo por el equipo de desarrollo referentes a cada una de las HU. Por último se presenta el diagrama de despliegue y se llevan a cabo las pruebas de aceptación, con el objetivo de certificar que el sistema propuesto funcione de forma correcta.

3.1 Tareas de Ingeniería

Las Tareas de Ingeniería (TI) definen cada una de las actividades necesarias para dar cumplimiento a cada HU y se describen de forma clara; de forma tal que se entienda lo que el sistema tiene que hacer y facilite su construcción. Para el desarrollo de la aplicación se determinaron un total de 65. A continuación se describen 3 de estas, otras 3 se muestran en el Anexo 3.

Tarea de Ingeniería	
Número de tarea: 1	Número de Historia de Usuario: 1
Nombre de la tarea: Investigar información necesaria de usuarios.	
Tipo de tarea: Investigación	Puntos estimado: 0.5
Fecha inicio: 27-10-2014	Fecha fin: 27-10-2014
Descripción: Inicia cuando se hace necesario conocer cuál es la información necesaria que se debe de almacenar en la base datos respecto a los usuarios, para el buen funcionamiento del producto. En este sistema la información de los usuarios es de mucha importancia, ya que mucho del valor agrado del producto depende en gran medida de ello.	

Tabla 11: Tarea de ingeniería 1 HU1 Registrar usuario.



Tarea de Ingeniería	
Número de tarea: 4	Número de Historia de Usuario: 2
Nombre de la tarea: Desarrollo de la interfaz para la modificación de la cuenta de usuario.	
Tipo de tarea: Desarrollo	Puntos estimado: 1.5
Fecha inicio: 3-11-2014	Fecha fin: 5-11-2014
Descripción: Inicia llevando a cabo el diseño e implementación de la interfaz de modificación de la cuenta de usuario, adaptando los componentes visuales específicos para este caso. Se debe de llevar a cabo una correspondencia entre los datos entrados y los soportados en la base de datos.	

Tabla 12: Tarea de ingeniería 4 HU2 Modificar cuenta de usuario.

Tarea de Ingeniería	
Número de tarea: 17	Número de Historia de Usuario: 5
Nombre de la tarea: Validar los datos de los conceptos a insertar.	
Tipo de tarea: Desarrollo	Puntos estimado: 0.5
Fecha inicio: 13-11-2014	Fecha fin: 13-11-2014
Descripción: Se implementan los métodos encargados de verificar que los datos del nuevo concepto posean el formato correcto y que el mismo no esté ya registrado en el proyecto en el que se desea utilizar.	

Tabla 13: Tarea de ingeniería 17 HU5 Insertar concepto.

3.2 Estilos de programación

Los estilos de programación definen la estructura y apariencia física del código, lo que facilita su comprensión, mantenimiento y lectura. En la implementación de la aplicación se utilizaron diferentes estilos que se describen de la manera siguiente:



3.2.1 Definiciones de clases

Las declaraciones de clases tienen su llave de apertura una línea más abajo de la declaración y el nombre de la clase comienza siempre con letra minúscula como se muestra en el siguiente ejemplo:

```
class usuario
{
    private $id_usuario;
    private $contrasena;
    private $nombre;
    private $s_nombre;
    private $apellidos;
    private $grado;
    private $area_id_area;
    private $estado;

    public $cnx;
```

Figura 8: Definición de clases.

3.2.2 Definición de métodos

Los métodos de las clases están precedidos por la palabra reservada que define el encapsulamiento del método (public, protected o private), seguido de la palabra function y después el nombre del método. El nombre comienza con una letra minúscula, en caso de ser un nombre compuesto por dos o más palabras, estos comenzarán con letra mayúscula como se muestra en la siguiente figura:

```
public function getEstado()
{
    return $this->estado;
}
```

Figura 9: Definición de métodos.

3.2.3 Asignaciones a variables

La asignación a variables se realiza siempre dejando un espacio antes y después del signo de igualdad (=) como se puede observar en la Figura 12 mostrada a continuación.



```
$id_proyecto = $_POST['id_proyecto'];
$indegree = array();
$outdegree = array();
$centrality = array();
$indegreeN = array();
$outdegreeN = array();
$centralityN = array();
$indegreeS = (double)0;
$outdegreeS = (double)0;
$centralityS = (double)0;
$indegreeN[$i] = $indegree[$i]/$indegreeS;
$outdegreeN[$i] = $outdegree[$i]/$outdegreeS;
$centralityN[$i] = $centrality[$i]/$centralityS;
```

Figura 10: Asignación a variables.

3.2.4 Estructuras de control

Las estructuras de control tales como if(), for(), while(), switch(), foreach(), entre otras se utilizaron colocando la llave de apertura en la misma línea de código y la de cierre en la última línea del módulo, como se puede observar en la siguiente figura:

```
for ($j = 0; $j < count($rConceptsProject); $j++){
    for ($k = 0; $k < count($rConceptsProject); $k++){
        $sumValuesVertexMaps[$j][$k] = (double)0;
        for ($i = 0; $i < count($rMapsProject); $i++){
            $rValueConcept = $cnxGetStaticAnalysis->recuperarDatos($sqlValueConcept);
            if(count($rValueConcept)) {
                $sumValuesVertexMaps[$j][$k] += $rValueConcept[0][0];
            }
        }
        $sumValuesVertexMaps[$j][$k] /= count($rMapsProject);
    }
}
```

Figura 11: Estructuras de control.

3.2.5 Llamada a funciones

La llamada a funciones se efectuara sin dejar espacios entre el nombre de la función, los paréntesis y los parámetros. Estos últimos separados por coma en el caso de existir más de uno y dejando un espacio después de la misma, como se puede apreciar en la siguiente figura:



```
//Select all maps.
$sqlMapsProject = "select * from mapa where proyecto_id_proyecto=". $id_proyecto."";

$rMapsProject = $cnxGetAggregation->recuperarDatos($sqlMapsProject);

//Select all concepts.
$sqlConceptsProject = "select * from concepto where proyecto_id_proyecto=". $id_proyecto."";

$rConceptsProject = $cnxGetAggregation->recuperarDatos($sqlConceptsProject);

//Select all labels.
$sqlLabelsProject = "select * from etiqueta where proyecto_id_proyecto=". $id_proyecto."";

$rLabelsProject = $cnxGetAggregation->recuperarDatos($sqlLabelsProject);
```

Figura 12: Llamada a funciones.

3.3 Diagrama de despliegue

Los diagramas de despliegue son los complementos de los diagramas de componentes que, unidos, proveen la vista de implementación del sistema. Describen la topología del sistema la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos. Los diagramas de despliegue representan a los nodos y sus relaciones. Un nodo es un elemento físico que existe en tiempo de ejecución y representa un recurso computacional, como son los procesadores o dispositivos de hardware, existen relaciones que representan medios de comunicación entre ellos.



Figura 13: Diagrama de Despliegue.

- **PC – cliente:** Está definida por las estaciones de trabajo que el usuario utilizará para acceder a la aplicación Web.
- **Servidor web:** El servidor de aplicación es utilizado para la publicación de la aplicación. Es la herramienta principal para ejecutar la lógica de negocio en el lado del servidor y es el encargado de ejecutar el código de las páginas



servidor. En el diagrama de despliegue se utiliza el servidor de aplicación Xamp.

- **Protocolos de comunicación:** Un protocolo de comunicación es un conjunto de reglas establecidas entre dos dispositivos para permitir la comunicación entre ambos:
 - **Conexión HTTP:** Es el protocolo utilizado entre los navegadores de los clientes y el servidor Web. Este elemento de la arquitectura representa un tipo de comunicación no orientado a la conexión entre clientes y servidor.
 - **Conexión TCP/IP:** Es el protocolo utilizado entre el Servidor Web y el Servidor de Base de Datos. Este protocolo garantiza que los datos serán entregados en su destino sin errores y en el mismo orden en que se transmitieron.
- **Servidor de Base de Datos:** Se hace referencia al gestor de bases de datos donde se encuentran los datos necesarios para el trabajo con el sistema. El servidor de base de datos elegido es PostgreSQL, que es un gestor de licencia gratuita y muy potente.

3.4 Planificación de las Pruebas

El proceso de pruebas de XP permite aumentar la calidad del sistema reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones.

XP divide las pruebas del sistema en dos grupos: pruebas unitarias, encargadas de verificar el código, las cuales están diseñada por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida y están diseñadas por el cliente final [78].

A continuación se presenta el cronograma de ejecución de las pruebas.



#	Tarea	Fecha	Responsables	Participantes	Observaciones
1	Planificación de las pruebas.	13-04-2015	Maikel Y. Leyva Vázquez, Karina Pérez Teruel, Julio C. Espronceda Pérez.	Yalena A. Venega Cañizares, Alberto J. Pérez Galban.	Caracterizar los contenidos necesarios y planificar las pruebas enmarcando el producto y sus especificaciones.
2	Diseño de los casos de prueba.	15-04-2015	Maikel Y. Leyva Vázquez, Karina Pérez Teruel, Julio C. Espronceda Pérez	Yalena A. Venega Cañizares, Alberto J. Pérez Galban.	Identificar las HU críticas, y elaborar los casos de prueba aplicables según sus respectivas descripciones.
3	Ejecución de los casos de prueba.	20-04-2015	Maikel Y. Leyva Vázquez, Karina Pérez Teruel, Julio C. Espronceda Pérez	Yalena A. Venega Cañizares, Alberto J. Pérez Galban.	Aplicar los casos de pruebas al sistema, y registrar los resultados de cada uno de ellos.

Tabla 14: Plan de pruebas.

- **Pruebas Unitarias**

El objetivo de las pruebas unitarias está enfocado en probar los elementos más pequeños del software. Es aplicable a componentes representados en el modelo de implementación para verificar que los flujos de control y de datos están cubiertos, y que ellos funcionen como se espera. La prueba de unidad siempre está orientada a caja blanca.

Las pruebas de caja blanca se centran en el minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del software proponiendo



casos de prueba que examinen que están correctas todas las condiciones y/o bucles para determinar si el estado real coincide con el esperado o afirmado. Esto genera gran cantidad de caminos posibles por lo que hay que dedicar esfuerzos a la determinación de las condiciones de prueba que se van a verificar.

Este tipo de prueba será aplicada a la estructura procedimental (código fuente) de las funcionalidades que implementa cada historia de usuario, a través de la *técnica del camino básico*.

La prueba del camino básico es una técnica de prueba de Caja Blanca propuesta por Tom McCabe. Esta técnica permite obtener una medida de la complejidad lógica de un diseño y usar esta medida como guía para la definición de un conjunto básico [78].

La idea es derivar casos de prueba a partir de un conjunto dado de caminos independientes por los cuales puede circular el flujo de control. Para obtener dicho conjunto de caminos independientes se construye el Grafo de Flujo asociado y se calcula su *complejidad ciclomática*. Los pasos que se siguen para aplicar esta técnica son:

- 1 A partir del diseño o del código fuente, se dibuja el grafo de flujo asociado.
- 2 Se calcula la complejidad ciclomática del grafo.
- 3 Se determina un conjunto básico de caminos independientes.
- 4 Se preparan los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico.

Los casos de prueba derivados del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa.

- **Pruebas de Aceptación**

Las pruebas de aceptación son creadas a partir de las HU. Durante una iteración la HU seleccionada en la planificación de iteraciones se convertirá en una prueba de aceptación. El cliente o usuario especifica los aspectos a probar cuando una HU ha sido correctamente implementada.

Una prueba de aceptación es como una caja negra. Cada una de ellas representa una salida esperada del sistema. Es responsabilidad del cliente verificar la corrección de las pruebas de aceptación y tomar decisiones acerca de las mismas [78].



Para implementar este nivel de prueba se utilizarán pruebas de caja negra, creando para cada HU uno o más casos de prueba en dependencia de las funcionalidades que involucre.

Cada caso de prueba debe contener un código, la HU a la que pertenece, el nombre, una breve descripción, la acción a probar, los datos de entrada, los resultados esperados y la evaluación de la prueba.

3.5 Pruebas de Caja Blanca

Al sistema desarrollado se le aplicó un grupo de 5 pruebas de caja blanca, haciendo uso de la técnica del camino básico, con el objetivo de evaluar la complejidad lógica de un diseño procedimental y usar esta medida como guía para la definición de un conjunto básico de caminos de ejecución [78]. Esta prueba permite garantizar que en los casos de prueba obtenidos a través del camino básico se ejecute cada sentencia del programa por lo menos una vez.

A continuación se analizan y enumeran las sentencias de código del método **actionLogin** contenidos en la clase **UsuarioController**. Este método permite la autenticación de los usuarios para acceder al sistema. En la *Figura 16* se detalla el código del mismo.

```
public function actionLogin() {
    $model = new LoginForm; //1

    // if it is ajax validation request
    if (isset($_POST['ajax']) && $_POST['ajax'] === 'login-form') { //2
        echo CActiveForm::validate($model); //3
        Yii::app()->end(); //4
    }

    // collect user input data
    if (isset($_POST['LoginForm'])) { //5
        $model->attributes = $_POST['LoginForm']; //6
        // validate user input and redirect to the previous page if valid
        if ($model->validate() && $model->login()) { //7
            $this->redirect(Yii::app()->homeUrl . '?r=usuario/index'); //8
        }
    }

    // display the login form
    $this->render('login', array('model' => $model)); //9
}
```

Figura 14: Método que permite el autenticar a los usuarios.

Luego, es necesario representar el grafo de flujo asociado al código antes presentado a través de nodos, aristas y regiones, ver *Figura 17*.

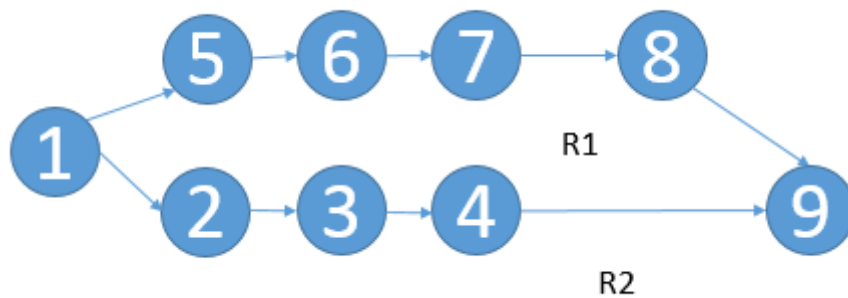


Figura 15: Grafo de flujo asociado al método actionLoguin.

Una vez construido el grafo de flujo asociado al procedimiento anterior, se determina la complejidad ciclomática, la cual es una métrica de software muy útil, pues proporciona una medición cuantitativa de la complejidad lógica de un programa. El valor calculado como complejidad ciclomática define el número de caminos independientes del conjunto básico de un programa y da un límite superior para el número de pruebas que se deben realizar.

Cálculo de la complejidad ciclomática para el grafo de flujo de la Figura 17

$$V(g) = (a - n) + 2 \text{ (I)} \quad V(g) = (p + 1) \text{ (II)} \quad V(g) = r \text{ (III)}$$

Siendo “ $V(g)$ ” el valor de la complejidad ciclomática “ a ” la cantidad total de aristas, “ n ” la cantidad total de nodos, “ p ” la cantidad total de nodos predicados (nodos de los cuales parten dos o más aristas) y “ r ” la cantidad total de regiones, se incluye el área exterior del grafo como una región más.

$$V(g) = (9 - 9) + 2 = 2 \text{ (I)} \quad V(g) = 1 + 1 = 2 \text{ (II)} \quad V(g) = 2 \text{ (III)}$$

La evaluación de las fórmulas I, II y III arroja un valor de complejidad ciclomática igual a 2, de manera que existen 2 posibles caminos por donde el flujo puede circular. Este valor representa el número mínimo de casos de pruebas para el procedimiento tratado. Seguidamente, es necesario especificar los caminos básicos que puede tomar el algoritmo durante su ejecución.

- Camino básico #1: 1 – 2 – 3 – 4 – 9
- Camino básico #2: 1 – 5 – 6 – 7 – 8 – 9

Se procede a ejecutar los casos de pruebas para cada uno de los caminos básicos determinados en el grafo de flujo, ver *Tablas 18, 19*. Para definir los casos de prueba es necesario tener en cuenta:

- **Descripción:** Se describe el caso de prueba y se especifican los aspectos fundamentales de los datos de entrada.



- **Condición de ejecución:** Se verifica que cada parámetro cumpla las condiciones de ejecución.
- **Entrada:** Se muestran los parámetros de entrada del procedimiento.
- **Resultados Esperados:** Se especifica el resultado que debe arrojar el procedimiento.

Caso de Prueba Camino Básico #1
<p>Descripción: La Variable “model” contiene los datos definidos para el formulario de autenticación en la aplicación.</p> <ul style="list-style-type: none">• Usuario: debe ser un usuario que no esté registrado en la aplicación.• Contraseña: cualquiera.
<p>Condiciones: El usuario debe acceder a la página principal del sistema, donde se encuentra el formulario de autenticación.</p>
<p>Entrada:</p> <ul style="list-style-type: none">• Usuario: qwerty.• Contraseña: asdfgh.
<p>Resultado Esperado: El sistema debe de emitir un mensaje de error comunicándole al usuario que el usuario o la contraseña son incorrectos.</p>
<p>Resultado: Satisfactorio.</p>

Tabla 15: Caso de Prueba para el Camino Básico #1.

Caso de Prueba Camino Básico #1
<p>Descripción: La Variable “model” contiene los datos definidos para el formulario de autenticación en la aplicación.</p> <ul style="list-style-type: none">• Usuario: debe ser un usuario que este registrado en la aplicación.• Contraseña: debe de ser la contraseña correspondiente al usuario que intenta autenticarse.
<p>Condiciones: El usuario debe acceder a la página principal del sistema,</p>



donde se encuentra el formulario de autenticación.

Entrada:

- Usuario: jacinto.
- Contraseña: j4c1nt0.

Resultado Esperado: El sistema autentica correctamente al usuario “jacinto” y muestra la página principal de este usuario.

Resultado: Satisfactorio.

Tabla 16: Caso de Prueba para el Camino Básico #2.

3.5.1 Resultados

Se realizaron un total de 5 casos de pruebas de caja blanca o Pruebas Unitarias (PU) como también se le conocen, a las funcionalidades más críticas del sistema. Los 10 casos de PU obtuvieron resultados satisfactorios representando el 100% de los realizados.



Figura 16: Resultados de las PU.

3.6 Pruebas de Caja Negra

Las Pruebas de Caja Negra o también llamadas Pruebas de Aceptación (PA) son creadas a partir de las historias de usuario. Especifican, desde la perspectiva del cliente, los escenarios para probar que una HU ha sido implementada correctamente. Para asegurar el correcto funcionamiento de una HU se realizan las PA que sea necesarias. Su propósito es garantizar que cada HU cumpla con lo



solicitado por el cliente. Las mismas no se consideran terminada hasta que no haya pasado sus PA,

Para realizar dichas pruebas se empleó la técnica Partición de Equivalencia. Esta permite examinar los valores válidos e inválidos de las entradas existentes en el software [76].

Para el desarrollo de la aplicación se determinaron un total de 39 casos de PA, a continuación se describen 3 estos.

Caso de Prueba de Aceptación	
Código: HU2_P1	Historia de Usuario: 2
Nombre: Modificar datos de usuario.	
Descripción: Esta prueba se encarga de validar la HU “Modificar datos de usuario”, modificando en la BD la nueva información que el usuario introduzca.	
Condiciones de Ejecución: El usuario debe de estar autenticado en el sistema.	
Entradas/Pasos de Ejecución: <ol style="list-style-type: none">1. Autenticarse con el usuario “pepe” y la contraseña “pepe.123,,,”.2. Ir al botón opciones, luego seleccionar preferencias.3. Luego de que el sistema muestre la página de preferencias, hacer clic ir al botón Modificar datos.4. Luego de que el sistema muestre los antiguos datos del usuario “pepe”, modificar los apellidos poniendo “Pérez Venega”, poner la antigua contraseña “pepe.123,,,” y en el campo de nueva contraseña y repetir nueva contraseña poner “p3p3... 123,,,”.5. Hacer clic en el botón Modificar.	
Resultado Esperado: El sistema debe almacenar en la BD los nuevos datos del usuario “pepe”.	
Resultado Obtenido: Los datos fueron modificados correctamente en la BD, el sistema muestra los nuevos datos del usuario “pepe”.	



Evaluación de la Prueba: Satisfactoria.

Tabla 17: Prueba de aceptación para la HU Modificar datos de usuario.

Caso de Prueba de Aceptación	
Código: HU15_P1	Historia de Usuario: 15
Nombre: Adicionar vértice.	
Descripción: Esta prueba se encarga de validar la HU “Adicionar vértice”, la cual se encarga de agregar un vértice en el mapa del proyecto en que el usuario está trabajando.	
Condiciones de Ejecución: El usuario debe de autenticarse en el sistema y debe de tener o un proyecto propio o estar colaborando en alguno.	
Entradas/Pasos de Ejecución: <ol style="list-style-type: none">1. Autenticarse con el usuario “yavenega” y la contraseña “Y4v3n3g4**+”.2. Ir a la opción “Mis proyectos”, luego hacer clic en la opción “Ver” del proyecto “ProyectoX”.3. El sistema muestra la página del proyecto “ProyectoX”.4. El usuario selecciona la opción “Mi MCD”, si no se ha creado el mapa, debe de hacer clic en la opción “Crear mapa...”, si ha se encuentra el mapa creado, debe de ir directamente a la opción “Adicionar vértice”.5. Seleccionar como concepto 1 “ConceptoXX1”, como concepto 2 “ConceptoYY2” y como etiqueta “muy_fuerte”.6. Hacer clic en el botón “Añadir”.	
Resultado Esperado: El sistema muestra a la derecha del nuevo vértice insertado con los que ya han sido insertados anteriormente y lo inserta en la base de datos.	
Resultado Obtenido: Los datos fueron insertados correctamente en BD y mostrado en el cuadro de información donde se muestran los vértices que están insertados en el mapa.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 18: Prueba de aceptación para la HU Adicionar vértice.



Caso de Prueba de Aceptación	
Código: HU24_P1	Historia de Usuario: 24
Nombre: Eliminar notificación.	
Descripción: Esta prueba se encarga de validar la HU “Eliminar notificación”, la cual elimina de la BD la notificación seleccionada.	
Condiciones de Ejecución: El usuario debe de autenticarse en el sistema y debe tener alguna notificación.	
Entradas/Pasos de Ejecución: <ol style="list-style-type: none">1. Autenticarse con el usuario “ajperez” y la contraseña “4lg0p\$\$”.2. El sistema en la página principal muestra las notificaciones, que posee el usuario “ajperez”.3. El usuario hace clic en la notificación con descripción “El proyecto: “Marketing Viral”, ha sido eliminado.”.	
Resultado Esperado: El sistema muestra elimina la notificación de la BD y de la lista de notificaciones mostradas al usuario.	
Resultado Obtenido: El sistema esconde las notificaciones y luego las muestra sin la notificación que ha sido eliminada.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 19: Prueba de aceptación para la HU Eliminar notificación.

3.6.1 Resultados

De un total de 39 casos de PA, 3 de ellos resultaron no satisfactorios, lo cual representa el 8% del total de casos de pruebas realizadas. Mientras que los 36 restantes resultaron satisfactorios para un 92% (Figura 18). Los errores detectados por los casos de pruebas no satisfactorios fueron mitigados después de 3 iteraciones de prueba.



Figura 17: Resultados de las PA.

Conclusiones parciales:

- Se definieron las TI, las cuales tuvieron como objetivo evitar la sobrecarga de trabajo en el equipo de desarrollo y agilizar aún más el proceso.
- Fueron detectados 3 casos de PA con resultados no satisfactorios, estos fueron mitigados tras tres iteraciones de prueba.
- La aplicación de las PU permitieron validar las funcionalidades críticas del sistema, mientras que las PA posibilitaron comprobar la correcta implementación de las historias de usuarios definidas con anterioridad.



CONCLUSIONES GENERALES:

Al concluir el presente trabajo se ha arribado a las conclusiones siguientes:

- Se desarrolló una aplicación web del tipo red social colaborativa, para la toma de decisiones en grupo basado en mapas cognitivos difusos, utilizando el modelo FCM – Fusión.
- La investigación sobre el nivel de eficiencia que tienen las herramientas web que utilizan el modelo FCM – Fusion, permitió obtener la información necesaria que ha servido de base para ejecutar la solución que se propone.
- Se validó el correcto funcionamiento del sistema propuesto a través de las pruebas unitarias y de aceptación.

RECOMENDACIONES:

Teniendo en cuenta los resultados obtenidos en esta investigación y basado en la experiencia adquirida, los autores recomiendan:

- Implementar el Módulo de Aprendizaje que propone el modelo FCM – Fusion basado en algoritmos genéticos.
- Implementar las medidas de intermediación y cercanía para integrarlos al Módulo de Análisis Estático, los cuales indican la importancia de un nodo en el flujo de información y cuán rápido difunde la información un nodo por la red, respectivamente.
- El despliegue de la aplicación en la red de la universidad para que pueda ser utilizada en la toma de decisiones por los usuarios de la misma.



REFERENCIAS BIBLIOGRÁFICAS:

- [1] A. M. Sharif and Z. Irani, "Applying a fuzzy-morphological approach to complexity within management decision making," vol. 44, ed: Emerald Group Publishing Limited, 2006, pp. 930-961.
- [2] M. Glykas, *Fuzzy Cognitive Maps: Advances in Theory, Methodologies, Tools and Applications*: Springer Verlag, 2010.
- [3] J. Ross, "Assessing Understanding of Complex Causal Networks Using an Interactive Game," Doctor of Philosophy in Information and Computer Science, University of California, Irvine, 2013.
- [4] J. M. Tacnet and J. Dezert, "Cautious OWA and evidential reasoning for decision making under uncertainty," 2011, pp. 1-8.
- [5] R. Srivastava, et al., "Belief Function Approach to Evidential Reasoning in Causal Maps," in *Causal Mapping for Research in Information Technology*, V. K. Narayanan and D. J. Armstrong, Eds., ed: Idea Group Pub., 2005.
- [6] J. Pearl, "Bayesian and belief-functions formalisms for evidential reasoning: a conceptual analysis," in *Readings in uncertain reasoning*, ed: Morgan Kaufmann Publishers Inc., 1990, pp. 540-574.
- [7] L. I. U. Zhi-Qiang, "Causation, bayesian networks, and cognitive maps," *ACTA AUTOMATICA SINICA*, vol. 27, pp. 552-566, 2001.
- [8] G. Pajares, et al., "Fuzzy Cognitive Maps Applied to Synthetic Aperture Radar Image Classifications Advances Concepts for Intelligent Vision Systems." vol. 6915, J. Blanc-Talon, et al., Eds., ed: Springer Berlin / Heidelberg, 2011, pp. 103-114.
- [9] J. Carvalho, "Rule Based Fuzzy Cognitive Maps in Humanities, Social Sciences and Economics Soft Computing in Humanities and SocialSciences." vol. 273, R. Seising and V. Sanz González, Eds., ed: Springer Berlin / Heidelberg, 2012, pp. 289-300.
- [10] J. L. Salmeron, "Fuzzy cognitive maps for artificial emotions forecasting," *Applied Soft Computing*, 2012.
- [11] J. L. Salmeron, et al., "Ranking fuzzy cognitive map based scenarios with TOPSIS," *Expert Systems with Applications*, vol. 39, pp. 2443-2450, 2012.
- [12] D. K. Iakovidis and E. Papageorgiou, "Intuitionistic Fuzzy Cognitive Maps for Medical Decision Making," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 15, pp. 100-107, 2011.



- [13] A. Altay and G. Kayakutlu, "Fuzzy cognitive mapping in factor elimination: A case study for innovative power and risks," *Procedia Computer Science*, vol. 3, pp. 1111-1119, 2011.
- [14] S. Lee, et al., "Development of a decision making system for selection of dental implant abutments based on the fuzzy cognitive map," *Expert Systems with Applications*, vol. 39, pp. 11564-11575, 2012.
- [15] J. L. Salmeron and E. I. Papageorgiou, "A Fuzzy Grey Cognitive Maps-based Decision Support System for radiotherapy treatment planning," *Knowledge-Based Systems*, vol. 30, pp. 151-160, June 2012 2012.
- [16] V. C. Georgopoulos and C. D. Stylios, "Fuzzy Cognitive Map Decision Support System for Successful Triage to Reduce Unnecessary Emergency Room Admissions for the Elderly," in *Fuzziness and Medicine: Philosophical Reflections and Application Systems in Health Care*, ed: Springer, 2013, pp. 415-436.
- [17] L. Curia and A. Lavalle, "Estrategias de decisión en sistemas dinámicos: aplicando mapas cognitivos difusos aplicación a un ejemplo socio – económico.," *Revista de Gestão da Tecnologia e Sistemas de Informação*, vol. 8, pp. 663-680, 2011.
- [18] M. Leyva-Vázquez, "MODELO DE AYUDA A LA TOMA DE DECISIONES BASADO EN MAPAS COGNITIVOS DIFUSOS", 2013.
- [19] C. Puente Águeda, et al., "Estudio de las relaciones causales: de un marco teórico a una aplicación práctica," in *Anales de mecánica y electricidad*, 2010, pp. 54-59.
- [20] J. B. Tenenbaum, et al., "How to grow a mind: Statistics, structure, and abstraction," *Science*, vol. 331, p. 1279, 2011.
- [21] S. Iamratanakul, et al., "Improving Project Portfolio Management with Strategic Alignment," in *PICMET 2009, Portland, Oregon USA*, 2009.
- [22] R. Bradley, et al. (2006, April 18). Aggregating causal judgements. Available: <http://eprints.lse.ac.uk/20088/>
- [23] A. Sobrino, "Imperfect Causality: Combining Experimentation and Theory." vol. 271, E. Trillas, et al., Eds., ed: Springer Berlin / Heidelberg, 2012, pp. 371-389.
- [24] C. Puente Águeda, et al., "Estudio de las relaciones causales," *Anales de mecánica y electricidad*, vol. 87, pp. 54-59, 2010.
- [25] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338-353, 1965.



- [26] B. M. d. Brio and A. S. Molina, *Redes Neuronales y Sistemas Borosos*, Segunda Edición ed.: Alfaomega, 2001.
- [27] I. Y. Sokar, et al., "KPIs Target Adjustment Based on Trade-off Evaluation Using Fuzzy Cognitive Maps," *Australian Journal of Basic and Applied Sciences*, vol. 5, pp. 2048-2053, 2011.
- [28] J. L. Salmeron, "Supporting decision makers with Fuzzy Cognitive Maps," vol. 52, ed: Industrial Research Institute, Inc, 2009, pp. 53-59.
- [29] B. Kosko, "Fuzzy cognitive maps," *International Journal of Man-Machine Studies*, vol. 24, pp. 65-75, 1986.
- [30] C. W. Ping, "A Methodology for Constructing Causal Knowledge Model from Fuzzy Cognitive Map to Bayesian Belief Network," PhD Thesis, Department of Computer Science, Chonnam National University, 2009.
- [31] E. I. Papageorgiou, "Learning Algorithms for Fuzzy Cognitive Maps---A Review Study," *Systems, Man, and Cybernetics, Part C: Applications and Reviews*, IEEE Transactions on, vol. PP, pp. 1-14, 2011.
- [32] B. Kosko, "Fuzzy engineering," ed: Prentice-Hall, Inc., 1997.
- [33] E. I. Papageorgiou and J. L. Salmeron., "A Review of Fuzzy Cognitive Maps research during the last decade," *IEEE Transactions on Fuzzy Systems.*, 2012.
- [34] E. I. Papageorgiou, et al., "Active Hebbian learning algorithm to train fuzzy cognitive maps," *International Journal of Approximate Reasoning*, vol. 37, pp. 219-249, 2004.
- [35] D. E. K. I. E. D. D. M. Emiris, "Learning Fuzzy Cognitive Maps using Evolution Strategies: a Novel Schema for Modeling and Simulating High-Level Behavior," 2001, p. 364.
- [36] E. I. Papageorgiou and P. P. Groumpos, "A new hybrid method using evolutionary algorithms to train Fuzzy Cognitive Maps," *Applied Soft Computing*, vol. 5, pp. 409-431, 2005.
- [37] N. Munier, *A Strategy for Using Multicriteria Analysis in Decision-making: A Guide for Simple and Complex Environmental Projects*: Springer, 2011.
- [38] F. Herrera, et al., "Computing with words in decision making: foundations, trends and prospects," *Fuzzy Optimization and Decision Making*, vol. 8, pp. 337-364, 2009.
- [39] J. Merigó, "New extensions to the OWA operators and its application in decision making," PhD Thesis, Department of Business Administration, University of Barcelona, 2008.



- [40] E. K. Zavadskas and Z. Turskis, "Multiple criteria decision making (MCDM) methods in economics: an overview," *Technological and Economic Development of Economy*, vol. 17, pp. 397-427, 2011/06/01 2011.
- [41] H.-B. Yan, et al., "A probabilistic model for linguistic multi-expert decision making involving semantic overlapping," *Expert Systems with Applications*, vol. 38, pp. 8901-8912, 2011.
- [42] M. Doumpos and C. Zopounidis, "Preference disaggregation and statistical learning for multicriteria decision support: A review," *European Journal of Operational Research*, vol. 209, pp. 203-214, 2010.
- [43] G. F. Barberis and M. C. E. Ródenas, "La Ayuda a la Decisión Multicriterio: orígenes, evolución y situación actual," presented at the VI Congreso Internacional de Historia de la Estadística y de la Probabilidad. , Valencia, 2011.
- [44] L. M. Varela-Pérez, "SISTEMA WEB PARA EL ANÁLISIS DE ESCENARIOS BASADO EN MAPAS COGNITIVOS DIFUSOS", 2014.
- [45] V. Torra and Y. Narukawa, *Modeling decisions: information fusion and aggregation operators*: Springer, 2007.
- [46] G. Beliakov, et al., *Aggregation functions: a guide for practitioners*: Springer, 2007.
- [47] J. Merigó, "New extensions to the OWA operators and its application in decision making," PhD Thesis, Department of Business Administration, University of Barcelona, 2008.
- [48] G. Arenas-Díaz, "Medidas difusas e integrales difusas," *Universitas Scientiarum*, vol. 18, pp. 7-32, 2013.
- [49] J. L. Marichal, "On Choquet and Sugeno Integrals as Aggregation Functions," *Fuzzy measures and integrals: theory and applications*, vol. 40, p. 247, 2000.
- [50] R. R. Yager, et al., *Recent Developments in the Ordered Weighted Averaging Operators: Theory and Practice*: Springer, 2011.
- [51] M. Grabisch, "The application of fuzzy integrals in multicriteria decision making," *European Journal of Operational Research*, vol. 89, pp. 445-456, 1996.
- [52] López, Natalia. "Redes Sociales y la privacidad", Observatorio Tecnológico. Instituto Nacional de Tecnologías Educativas y Formación del Profesorado. Ministerio de Educación, Cultura y Deporte. [Consultado 14-01-15] [En línea] <http://recursostic.educacion.es/observatorio/web/es/internet/web-20/705-las-redes-sociales-y-la-privacidad>



[53] Schenone, Hernán Marcelo. Diseño de una Metodología Ágil de Desarrollo de Software. 2004.

[54] SELECTING A DEVELOPMENT APPROACH. Revalidated: March 27, 2008. Consultado: 27 Oct 2008. [En línea] <https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf>

[55] Rodriguez-Perez, JJ. "SISTEMA DE RECUPERACIÓN DE INFORMACIÓN DE MERCADOTECNIA PROVENIENTE DE INTERNET PARA ALBET S.A" 2008.

[56] Ecured. [En línea] http://www.ecured.cu/index.php/Metodolog%C3%ADas_de_desarrollo_de_software

[57] ROBERTH G. FIGUEROA, CAMILO J. SOLÍS, ARMANDO A. CABRERA. Metodologías tradicionales vs. Metodologías ágiles. [En línea] 2008. [Citado el: 20 de enero de 2014.] <http://adonisnet.wordpress.com/2008/06/18/metodologias-tradicionales-vs-metodologias-agiles>.

[58] IVAR JACOBSON, G. B., JAMES RUMBAUGH. El Proceso Unificado de Desarrollo de Software. Edtion ed. Madrid, 2000. [En línea] ISBN 84-7829-036-2.

[59] MARTÍNEZ, A. M. R. Guía a Rational Unified Process. [En línea] 2003. [Citado el: 5 de 12 de 2013.] pp. 1, 2.

[60] Loaiza, Douglas Alfredo. [En línea] <http://pnfiingenieriadesoftwaregrupocuatro.blogspot.com/2012/07/bienvenidos-al-blog.html>

[61] Patricio Letelier, M^a Carmen Penadés. Metodologías ágiles para el desarrollo de software:eXtreme . Programming (XP). [En línea] [Citado el: 18 de 11 de 2013.] <http://eva.uci.cu>

[62] Brian Kernighan; P. J. Plauger (1976), Software Tools, Addison-Wesley, pp. 352, ISBN 020103669X

[63] Pavan, Bárbara. "ENTENDIENDO HTML5: GUÍA PARA PRINCIPIANTES" [Consultado en línea: 15-01-15] <http://bitelia.com/2013/05/entendiendo-html5-guia-para-principiantes>

[64] <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-262.pdf>

[65] M. Domínguez-Dorado, Todo Programación. Nº 12. Págs. 48-51. Editorial Iberprensa(Madrid). DL M-13679-2004. Septiembre, 2005. Bases de datos en el cliente con JavaScript DB.



- [66] Eguíluz Pérez, Javier. "Introduccion a CSS" [Disponible en:] <http://www.librosweb.es/css>
- [67] D., C. Manual de PHP. s.l. : Edtion ed., 2012.
- [68] Gerner, Jason y et al. Professional LAMP: Linux, Apache, MySQL and PHP 5 Web Development. Indianapolis: Wiley Publishing, 2006. 978-0-7645-9723-7.
- [69] GitHub: Popular Watched Repositorie. [Consultado en línea 15-01-15] <https://github.com/popular/watched>
- [70] Sitio web Oficial jQuery [Citado en línea 15-01-2015] <http://jquery.com/>
- [71] About Yii. [Consultado en línea: 15-01-15] <http://www.yiiframework.com/about/>
- [72] M. Domínguez-Dorado,. Todo Programación. Nº 13. Págs. 32-34. Editorial Iberprensa (Madrid). DL M-13679-2004. Noviembre, 2005. NetBeans IDE 4.1. La alternativa a Eclipse.
- [73] JOSÉ H. CANÓS, P. L., M^a CARMEN PENADÉS. Metodologías Ágiles en el Desarrollo de Software 2002.
- [74] GONZÁLEZ, Y. Patrón Modelo-Vista-Controlador. In Revista Telemática. 2012, p. 9.
- [75] DATE, C. J. Introducción a los sistemas de bases de datos. Edition ed.: José Luis Vázquez, 2001. 960 p. ISBN 968-444-419-2.
- [76] Larman, Craig. UML y Patrones Introducción al análisis y diseño orientado a objetos y al proceso unificado. 2. s.l. : Prentice Hall, 2003. págs.162-177.
- [77] Diagrama de Despliegue. [En línea] <http://www.diadspg.blogspot.com/>
- [78] MAITE RODRÍGUEZ CORBEA y MEYLIN ORDÓÑEZ PÉREZ. « LA METODOLOGÍA XP APLICABLE AL DESARROLLO DEL SOFTWARE EDUCATIVO EN CUBA». [En línea]. Disponible en: http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_0837_07/1/TD_0837_07.pdf.
- [79] Aamodt, Agnar, and Enric Plaza. "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches" Artificial Intelligence Communications 7, no. 1 (1994): 39-52.



ANEXOS:

Anexo 1: Historias de usuario.

Historia de Usuario	
Número: 1	Nombre de Historia de Usuario: Registrar usuario.
Modificación de Historia de Usuario Número: 0	
Usuario: Usuario_Sistema	Iteración Asignada: 1
Prioridad en Negocio: Muy Alta	Puntos Estimados: 2
Riesgo en Desarrollo: Medio	Puntos Reales: 4
Descripción: El sistema debe permitir que nuevos usuarios puedan registrarse en el sistema, insertar su información personal y de esta forma poder acceder a las funcionalidades de los usuarios registrados.	
Observaciones: En caso de que el nuevo usuario intente registrarse con algún "nombre de usuario" que se encuentra ya en uso, debe de enviar un mensaje alertándole que dicho nombre no está disponible.	
Prototipo de Interfaz:	
<div style="text-align: center;"><input type="text" value="Usuario"/> <input type="text" value="Contraseña"/> <input type="text" value="Repetir contraseña"/> <input type="text" value="Nombre"/> <input type="text" value="Apellidos"/> <input type="text" value="Elija un grado científico..."/> ▼ <input type="text" value="Elija un área de investigación..."/> ▼ <input type="button" value="Registrar"/></div>	

Anexo 1.1: HU1 Registrar usuario.



Historia de Usuario

Número: 17 **Nombre de Historia de Usuario:** Obtener agregación por media aritmética.

Modificación de Historia de Usuario Número: 0

Usuario: Usuario_Registrado **Iteración Asignada:** 2

Prioridad en Negocio: Alta **Puntos Estimados:** 2

Riesgo en Desarrollo: Medio **Puntos Reales:** 3

Descripción: El sistema debe permitir obtener agregación de los mapas de un proyecto mediante el operador de agregación media aritmética.

Observaciones: Si en el proyecto existen menos de 2 mapas, el sistema debe comunicarle al usuario que no se puede obtener la agregación.

Prototipo de Interfaz:

Agregación de mapas del proyecto **Consenso**, usando el operador: **Media**.

Conceptos	Forma del número	Cantidad que representa	Filosofía del número
Forma del número	0.52	0.62	0.36
Cantidad que representa	0.3	0.28	0.32
Filosofía del número	0.44	0.14	0.36

Anexo 1.2: HU17 Obtener agregación por media aritmética.



Historia de Usuario

Número: 19

Nombre de Historia de Usuario: Obtener análisis estático.

Modificación de Historia de Usuario Número: 0

Usuario: Usuario_Registrado

Iteración Asignada: 2

Prioridad en Negocio: Alta

Puntos Estimados: 3

Riesgo en Desarrollo: Medio

Puntos Reales: 4

Descripción: Inicia cuando el usuario decide obtener la tabla con los datos de análisis estático de un proyecto en específico.

Observaciones: Si en el proyecto existen menos de 2 mapas, el sistema debe comunicarle al usuario que no se puede obtener el análisis estático.

Prototipo de Interfaz:

Conceptos	Indegree	Outdegree	Centrality	CentralityN	IndegreeN	OutdegreeN
1. Forma del número	1.5	1.26	2.76	0.44910179640719	0.37724550898204	0.41317365269461
2. Cantidad que representa	0.9	1.04	1.94	0.26946107784431	0.31137724550898	0.29041916167665
3. Filosofía del número	0.94	1.04	1.98	0.2814371257485	0.31137724550898	0.29640718562874

Anexo 1.3: HU17 Obtener análisis estático.



Anexo 2: Tarjetas CRC.

P_Abierto_Control	
Responsabilidades	Colaboraciones
Listar proyectos abiertos. Insertar proyecto abierto. Modificar proyecto abierto. Eliminar proyecto abierto.	Proyecto_Control.

Anexo 2.1: Tarjeta CRC P_Abierto_Control.

Notificación_Control	
Responsabilidades	Colaboraciones
Listar notificaciones. Insertar notificaciones. Modificar notificaciones. Eliminar notificaciones.	Usuario_Control.

Anexo 2.2: Tarjeta CRC Notificación_Control.

P_Abierto_Control	
Responsabilidades	Colaboraciones
Listar conceptos. Insertar concepto.	Proyecto_Control.

Anexo 2.3: Tarjeta CRC Concepto_Control.

Anexo 3: Tareas de ingeniería.

Tarea de Ingeniería	
Número de tarea: 19	Número de Historia de Usuario: 9
Nombre de la tarea: Investigar información necesaria de etiquetas.	
Tipo de tarea: Investigación	Puntos estimado: 1



Fecha inicio: 15-11-2014	Fecha fin: 16-11-2014
Descripción: Inicia cuando se hace necesario definir los grupos de etiquetas que se van a utilizar en la construcción de los MCD. Estas deben de tener diferentes niveles de granularidad.	

Anexo 3.1: Tarea de ingeniería 9 HU9 Registrar usuario.

Tarea de Ingeniería	
Número de tarea: 24	Número de Historia de Usuario: 12
Nombre de la tarea: Desarrollo de la interfaz para mostrar los datos de los MCD.	
Tipo de tarea: Desarrollo	Puntos estimado: 1.5
Fecha inicio: 3-11-2014	Fecha fin: 5-11-2014
Descripción: Inicia llevando a cabo el diseño e implementación de la interfaz para mostrar los datos de un MCD.	

Anexo 3.2: Tarea de ingeniería 24 HU12 Mostrar información del MCD.

Tarea de Ingeniería	
Número de tarea: 30	Número de Historia de Usuario: 15
Nombre de la tarea: Validar los datos de los vértices a insertar.	
Tipo de tarea: Desarrollo	Puntos estimado: 0.5
Fecha inicio: 13-11-2014	Fecha fin: 13-11-2014
Descripción: Se implementan los métodos encargados de verificar que los datos del nuevo vértice que se desea insertar sean correctos, los conceptos deben de pertenecer al proyecto al cual pertenece el mapa al que se le desea adicionar el vértice, así mismo como la etiqueta que se seleccione para el mismo.	

Anexo 3.2: Tarea de ingeniería 30 HU15 Adicionar vértice.



Anexo 4: Modelo de Datos – Red Social NEXO.

