

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



RED SOCIAL “NEXO”, PARA LA
TOMA DE DECISIONES EN GRUPO:
*MÓDULO PARA LA AUTOMATIZACIÓN DEL
PROCESO DE CONSENSO EN MAPAS
COGNITIVOS DIFUSOS.*

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

AUTOR: ORLANDO CRUZ TORRES

TUTOR: DRA. KARINA PÉREZ TERUEL.

DR. MAIKEL YELANDI LEYVA VÁZQUEZ.

LA HABANA 2015





"No es la inteligencia, recibida o casual lo que da al hombre honor: sino el modo con que la usa y la salva. No hay más que un modo de perdurar: y es servir."

José Martí.

DECLARACIÓN DE AUTORÍA

Declaro por este medio que yo Orlando Cruz Torres, con carnet de identidad 91031637784, soy el autor principal del trabajo final de tesis Red Social "NEXO", para la toma de decisiones en grupo: Módulo para la automatización del proceso de consenso en Mapas Cognitivos Difusos, y que autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Y para que así conste, firmo la presente declaración jurada de autoría en Ciudad de La Habana a los __días del mes de_____ del año_____.

Orlando Cruz Torres

Firma del Autor

Dra. Karina Pérez Teruel

Firma del Tutor

Dr. Maikel Yelandi Leyva Vázquez

Firma del Tutor

DATOS DE CONTACTO

Dra. Karina Pérez Teruel: Graduada de Ingeniería Informática en el ISPJAE en el año 2005, de la maestría de Bioinformática del INSTEC en el 2007, y Doctor en Ciencias Técnicas en el 2014. Es profesora auxiliar y labora en el 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

Dr. Maikel Yelandi Leyva Vázquez: Graduado de Ingeniería Informática en el ISPJAE en el año 2005, de la maestría de Bioinformática del INSTEC en el 2007, y Doctor en Ciencias Técnicas en el 2013. Es profesor auxiliar y labora en el Centro de Consultoría y Desarrollo de Arquitecturas Empresariales (CDAE) de la Facultad 5 de la Universidad de las Ciencias Informáticas (UCI).

Correo-Electrónico: mleyvaz@uci.cu

DEDICATORIA

Quiero dedicar este trabajo a mis padres, mi tía, mi hermano y demás familiares, por siempre confiar en mí y darlo todo para que me superara profesionalmente. En especial a mi madre, por todo su esfuerzo y sacrificio, que con su perseverancia me ha dado fuerzas para seguir adelante hasta lograr mi objetivo, a sus consejos que me han ayudado a enfrentar todas las dificultades que aparecieron a lo largo del camino, por su confianza y por su entrega, por su seguridad en que siempre puedo vencer los obstáculos que se me presenten.

AGRADECIMIENTOS

A mi madre Onilda Regina Torres Rodríguez por dedicarme su tiempo y sacrificio, por estar siempre pendiente de todo, apoyándome en los buenos y malos momentos, dándome la confianza necesaria para lograr lo que siempre me he propuesto. Pero por sobre todo, por darme la oportunidad de nacer.

A mis abuelas Zoila y Felicia por desde pequeño apoyarme en mis estudios, por cuidarme y saber dirigir mi vida hacia un buen camino. Gracias a las dos por cuidarme desde el cielo.

A mi tía Aracelia León Montes de Oca por apoyarme en todo desde el principio, por su entereza y dedicación, por quererme como un hijo y por brindarme su cariño.

A mi padre Orelvis Cruz Montes de Oca y a mi madrastra Beatriz por todo el apoyo que me han brindado.

A mi hermano Andy Niesther Noy Torres por su tiempo y por su preocupación.

A mi gran amiga Rosmery Pedraza Ceballo por aceptarme como un hermano más, por estar siempre ayudándome, por sus consejos y su tiempo. Por llorar junto a mí en los momentos más difíciles y celebrar también aquellos más felices.

A Samuel Ortiz Martin por brindarme su tiempo, por apoyarme en las circunstancias más difíciles y estar siempre a mi lado, por permitirme disfrutar tantos momentos de felicidad. Por dejarme conocer a mi otra familia, a los cuales quiero tanto como a la mía. A Margarita, a Mima, a todos...

A mi padrastro Lorenzo Rodríguez Rodríguez y a su familia por la ayuda brindada a lo largo de esta carrera. A Alicia, Mabety y Jorge, muchas gracias.

A Melva Rodríguez por su apoyo y ser como otra madre para mí desde la distancia.

A mi amigo Carlos Heriberto Cordoví García por su ayuda, por su tiempo y por demostrarme que la amistad es uno de los tesoros más preciados que puede tener el hombre. Gracias Carlitos.

A mis amigos Reitel, Sergio, Liuver, Raudel (el rojo), Manuel, Julio, Yurelys, Juan Emilio, Ofelia, Elizabeth, Vladimir, Ángel, Yaikel, Adolfo, a mis compañeros de grupo, gracias a todos por estos cinco años de mi vida, los cuales nunca olvidaré porque ustedes lograron que sea algo imborrable.

A mis compañeros de tesis Alberto y Yalena, por el gran equipo que hicimos demostrando así que todo es posible y que no hay tarea que no sea capaz de hacerse.

A mis tutores Karina Pérez Teruel y Maikel Yelandi Leyva Vázquez, por ayudarme y hacer posible la realización de este trabajo. Ambos son excelentes personas, y para mí fue un honor y un placer trabajar con ustedes. Gracias.

A mis profesores de la Universidad por haberme enseñado todos los conocimientos que hoy tengo.

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

A todos Muchas Gracias...

RESUMEN

Uno de los avances significativos en el proceso de la toma de decisiones es el uso de los mapas cognitivos difusos. Sin embargo, en las redes sociales existentes en la actualidad el proceso de toma de decisiones no hace uso de los mapas cognitivos difusos, por lo que se imposibilita realizar el proceso de consenso mediante el uso de los mismos. Por tanto, el presente trabajo tiene como finalidad la automatización del proceso de consenso mediante el uso de mapas cognitivos difusos en la red social NEXO (una red social que permite la toma de decisiones en grupos). Mediante la aplicación del modelo Ágora (modelo para el proceso de logro de consenso) a los mapas cognitivos difusos individuales de los usuarios de NEXO, se logró automatizar el proceso de consenso mediante la satisfacción del 100% de los casos de prueba de aceptación realizados por el cliente.

Palabras clave: Ágora, consenso, mapas cognitivos difusos, NEXO, red social, toma de decisiones

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	5
1.1 Toma de decisiones en grupo	5
1.2 Consenso	6
1.2.1 <i>Proceso de logro de consenso (PLC)</i>	6
1.3 Mapas Cognitivos Difusos (MCD)	8
1.3.1 <i>MCD en la toma de decisiones en grupos</i>	8
1.3.2 <i>Consenso en MCD</i>	9
1.4 Modelos PLC para la toma de decisiones en grupo	9
1.4.1 <i>Métodos usados en PLC para la toma de decisiones en grupo mediante el uso de MCD</i>	9
1.4.2 <i>Modelo Ágora</i>	11
1.5 Metodologías para el desarrollo del software	16
1.5.1 <i>Programación Extrema (XP)</i>	16
1.6 Herramientas de Lenguaje y Modelado	17
1.6.1 <i>Herramienta CASE Visual Paradigm</i>	17
1.6.2 <i>Lenguajes de programación del lado del cliente</i>	17
1.6.3 <i>Lenguajes de programación del lado del servidor</i>	18
1.6.4 <i>Framework de desarrollo Yii</i>	19
1.6.5 <i>Framework de CSS Bootstrap</i>	20
1.6.6 <i>Sistema Gestor de Base de Datos PostgreSQL</i>	21
1.7 Entorno Integrado de Desarrollo	21
CONSIDERACIONES PARCIALES	22
CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA APLICACIÓN	23
2.1 Fase de exploración	23
2.1.1 <i>Historias de Usuario</i>	23
2.2 Fase de Planificación	28
2.2.1 <i>Lista de Reservas del Producto</i>	28

2.2.1.1. Requisitos Funcionales.....	28
2.2.1.2. Requisitos no Funcionales.....	29
2.2.2. Estimación del esfuerzo por Historia de Usuario.....	31
2.2.3. Plan de Iteraciones	32
2.2.4. Plan de Entrega	32
2.3 Fase de Iteraciones.....	33
2.3.1 Tarjetas CRC.....	33
2.4 Arquitectura de Software.....	34
2.4.1. Patrón Modelo Vista Controlador (MVC).....	34
2.5.1. Patrones GRASP.....	37
2.5.2 Patrones GOF.....	39
CONSIDERACIONES PARCIALES	39
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA.....	40
3.1 Tareas de Ingeniería	40
3.2 Estilos de Programación.....	43
3.2.1 Definiciones de Clases	44
3.2.2 Definiciones de Métodos	44
3.2.3 Llamadas a funciones y asignación de variables	44
3.2.4 Estructuras de Control.....	45
3.3 Diagrama de Despliegue	45
3.4 Planificación de las Pruebas.....	47
3.5 Pruebas de Caja Blanca	48
3.5.1 Resultados.....	53
3.6 Pruebas de Caja Negra.....	54
3.6.1 Resultados.....	59
3.7 Resultados Esperados	60
CONSIDERACIONES PARCIALES	60
CONCLUSIONES GENERALES.....	61

ÍNDICE DE FIGURAS

Fig. 1: Proceso de consenso y selección en problemas de toma de decisiones en grupo (6).....	7
Fig. 2: Mapa cognitivo difuso y su correspondiente matriz de adyacencia.	8
Fig. 3: Estructura, componentes, cualidades y principios del modelo Ágora.	12
Fig. 4: Modelo PLC Ágora.....	12
Fig. 5: Funcionamiento del Patrón Modelo Vista Controlador de Yii. (MVC).....	34
Fig. 6: Aplicación de la arquitectura en la capa Modelo.....	35
Fig. 7: Aplicación de la arquitectura en la capa Vista.	36
Fig. 8: Aplicación de la arquitectura en la capa Controlador.....	36
Fig. 9: Diagrama de clases que representa el Patrón Creador.	37
Fig. 10: Diagrama de clases que representa el Patrón Controlador.	38
Fig. 11: Definición de Clases.	44
Fig. 12: Definición de Métodos.	44
Fig. 13: Llamadas a funciones y asignación de variables.	45
Fig. 14: Estructura de control.....	45
Fig. 15: Diagrama de Despliegue.	46
Fig. 16: Método que permite el registro de parámetros sobre un consenso.	49
Fig. 17: Grafo de flujo asociado al método actionCreate.	50
Fig. 18: Resultados de las pruebas de caja blanca.	53
Fig. 19: Resultados de las pruebas de caja negra.....	60

ÍNDICE DE TABLAS

Tabla 1: Tabla comparativa entre los modelos de PLC Delphi y Bryson (29).....	11
Tabla 2: HU1 Establecer Parámetros del Consenso.	23
Tabla 3: Listar Parámetros del Consenso.....	24

Tabla 4: HU4 Calcular Consenso.	25
Tabla 5: HU5 Generar Recomendaciones.	26
Tabla 6: HU6 Notificar Recomendación.	27
Tabla 7: Estimación de esfuerzo por HU	31
Tabla 8: Plan de Iteraciones.	32
Tabla 9: Plan de Entrega.	32
Tabla 10: Tarjeta CRC ConsensoController.	33
Tabla 11: Tarjeta CRC RecomendacionController.	34
Tabla 12: Tarea de Ingeniería 1 de la HU7	40
Tabla 13: Tarea de Ingeniería 1 de la HU1	41
Tabla 14: Tarea de Ingeniería 1 de la HU2	41
Tabla 15: Tarea de Ingeniería 1 de la HU3	42
Tabla 16: Tarea de Ingeniería 1 de la HU5	42
Tabla 17: Tarea de Ingeniería 1 de la HU6	43
Tabla 18: Tarea de Ingeniería 1 de la HU4	43
Tabla 19: Caso de Prueba para el Camino Básico #1.	51
Tabla 20: Caso de Prueba para el Camino Básico #2.	51
Tabla 21: Caso de Prueba para el Camino Básico #3.	52
Tabla 22: Prueba de aceptación para la HU Establecer Parámetros del Consenso.	54
Tabla 23: Prueba de aceptación para la HU Modificar Parámetros del Consenso.	55
Tabla 24: Prueba de aceptación para la HU Calcular Consenso.	56
Tabla 25: Prueba de aceptación para la HU Generar Recomendaciones.	57
Tabla 26: Prueba de aceptación para la HU Notificar Recomendaciones.	58
Tabla 27: Prueba de aceptación para la HU Monitorear Recomendaciones.	59



INTRODUCCIÓN

La toma de decisiones es el proceso por medio del cual se obtiene como resultado una o más decisiones con el propósito de dar solución a una situación. Pueden participar uno o más actores y se elige entre varias alternativas. Nadie se encuentra exento de tomar decisiones. Todos en algún momento han tenido que recurrir a tomar una de ellas tanto en el entorno personal o laboral (60).

Diariamente vemos como en algunas organizaciones empresariales, educativas o comités, las decisiones son tomadas en grupos, otra de las vías comunes donde se toman decisiones de gran importancia en cualquier entorno. La toma de decisiones en grupo mezcla a individuos relacionados entre sí, que aportan propuestas y alternativas para lograr una interacción e intercambiar ideas. Los grupos generan información y conocimientos más completos al agregar recursos de diversos individuos que contribuyen más al proceso de decisión ofreciendo una incrementada diversidad de puntos de vista que representa mayor oportunidad de considerar más conocimiento para tomar una decisión. Estas decisiones pueden estar dirigidas para obtener un fin común, tanto para lograr el consenso, tomar una decisión conjunta o solucionar un problema, manteniendo en sí la efectividad, la eficiencia y la calidad necesaria.

La toma de decisiones por consenso es un proceso de decisión que busca no solamente el acuerdo de la mayoría de los participantes, sino también persigue el objetivo de resolver o atenuar las objeciones de la minoría para alcanzar la decisión más satisfactoria (41). Más que una lista de alternativas conocidas, debate en corto tiempo, voto, y aceptación o rechazo por algún porcentaje de mayoría (por ejemplo, más del 50% o 2/3 partes), un proceso de decisión por consenso implica, identificar y discutir las inquietudes, generar nuevas alternativas, combinar elementos de múltiples alternativas y confirmar que las personas entienden una propuesta o un argumento. Lograr tomar una decisión en grupo mediante el consenso puede implicar un arduo trabajo, por tal razón, algunas veces se utiliza la representación gráfica mediante grafos dirigidos. En este sentido existen técnicas de lógica difusa dentro de las cuales se encuentran los Mapas Cognitivos Difusos (MCD).

Los MCD han resultado útiles para la representación de modelos mentales tanto individuales como colectivos. Su capacidad para ser agregados y construir MCD grupales a partir de MCD individuales los hace apropiados como técnica para asistir a la toma de decisiones en grupo. Una actividad significativa para el aprendizaje organizacional lo constituye la construcción del consenso en los modelos mentales de



los integrantes de las organizaciones. La obtención del MCD que representa el conocimiento causal de los expertos sobre el dominio mediante la agregación de los mapas individuales, la hace menos susceptible a creencias potencialmente erróneas de un único experto y sirviendo como base de conocimiento colectivo (38).

En organizaciones de gran tamaño o con equipos distribuidos geográficamente se hace necesario automatizar este proceso. Actualmente no se reportan en la literatura herramientas que automaticen el proceso de consenso en MCD (39; 21).

Aunque pese a la existencia de un grupo de herramientas para el trabajo con MCD, éstas no son adecuadas para ayudar a la toma de decisiones de forma general (40, 18). Tampoco han sido aprovechadas las ventajas que ofrece la tecnología web para desarrollar aplicaciones multiusuarios que permitan a la sociedad desde cualquier punto del planeta realizar modelos colectivos, plantear sus ideas y tomar decisiones en grupo, además de poder llegar a lograr un proceso de consenso mediante un sistema de recomendaciones en MCD que brinde la posibilidad de realizar una mejor comunicación entre los usuario, facilitando así un entendimiento por ambas partes para lograr el consenso.

Después de un análisis de las necesidades existentes y con el fin de solucionar la situación anterior, se plantea el siguiente **problema de la investigación**: ¿Cómo realizar el proceso de consenso en MCD a través del modelo Ágora en una red social?

Se define como **objetivo general**: Desarrollar un módulo que automatice el proceso de consenso en MCD mediante el uso del modelo Ágora en la Red Social NEXO.

Lo que determina como **objeto de estudio**: toma de decisiones en grupo y redes sociales y se precisa como **campo de acción**: proceso de consenso en redes sociales para la toma de decisiones en grupo.

Para dar cumplimiento al objetivo propuesto, se han definido las siguientes tareas de investigación:

1. Elaboración del marco teórico conceptual relacionado con los aspectos teóricos que sustentan la investigación.
2. Realización del análisis y diseño de la herramienta.
3. Desarrollo de un módulo informático para la red social NEXO.
4. Realización de pruebas funcionales al módulo.
5. Validación de la solución propuesta.



A lo largo de esta investigación se utilizarán los siguientes métodos científicos:

Métodos teóricos:

- ✓ **Análisis-síntesis:** Para el estudio de los conceptos empleados relacionados con MCD.
- ✓ **Histórico-lógico:** Para analizar, con mayor profundidad los antecedentes y las tendencias actuales referidas a las herramientas que sirven como soporte a la toma de decisiones y al uso del consenso empleando MCD.
- ✓ **Modelación:** Para la creación y elaboración de diagramas que modelan la estructura, relaciones internas y características del objeto de investigación.

Métodos empíricos:

- ✓ **Experimental:** Se utiliza para efectuar la ejecución de pruebas, creando las condiciones que permiten validar el correcto funcionamiento del software desarrollado en la presente investigación.
- ✓ **Observación:** Se utilizó para ver apreciar el comportamiento de diferentes redes sociales como LinkedIn y Twitter, para poder enfocar mejor el diseño y evitar errores.
- ✓ **Consultas de fuentes de Información:** Este método se utilizó para la consulta de las fuentes bibliográficas durante la investigación, lo cual fue esencial para la elaboración del marco teórico.

El presente documento se encuentra estructurado en 3 capítulos:

Capítulo 1 Fundamentación teórica

En este capítulo se abordan los conceptos principales relacionados con la investigación para el desarrollo de la solución al problema planteado. Se realiza además un análisis de las redes sociales. Y se explican la metodología y herramientas utilizadas para dar solución a la problemática planteada.

Capítulo 2 Análisis y diseño de la aplicación

En este capítulo se identifican todos los requerimientos funcionales y no funcionales del módulo a implementar. Se muestran los principales procesos a través de casos de uso del sistema, los actores que en ella intervienen, sus relaciones y la descripción de cada uno de ellos. Además, se elabora el modelo de diseño con sus respectivos diagramas.



Capítulo 3 Implementación y validación de la aplicación

Se tratan los aspectos relacionados con la implementación de la solución propuesta. Además, se valida la misma a través de las pruebas que se le realizan al sistema.



CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se abordan los conceptos principales relacionados con la investigación para el desarrollo de la solución al problema planteado. Se realiza además un análisis de las redes sociales. Y se explican la metodología y herramientas utilizadas para dar solución a la problemática planteada.

1.1 Toma de decisiones en grupo

Los grupos, además de resolver tareas y proporcionar una base afectiva a sus miembros, toman decisiones. En los estudios que se han dedicado al análisis de este proceso se ha prestado especial atención a:

1. los mecanismos a través de los cuales el grupo toma decisiones
2. la calidad de la decisión

Y se ha visto que existen diferencias entre las decisiones individuales y en grupo. Como resultado de los estudios, se concluyó que el grupo es superior al individuo en la toma de decisiones y en la solución de problemas; generalmente los grupos (1):

1. aprenden más rápido
2. recuerdan mejor
3. toman mejores decisiones
4. cometen menos errores
5. son más productivos (en cantidad y calidad)

Tomar decisiones en grupos incluye actividades típicas tales como (54):

1. Definir el problema de toma de decisiones.
2. Analizar el problema e identificar alternativas de solución:

$$X = \{x_1, x_2, \dots, x_n\} (n \geq 2).$$

3. Establecer criterio(s) de evaluación.
4. Seleccionar experto(s).



5. Evaluar alternativas.
6. Ordenar y seleccionar la mejor alternativa.
7. Implementarla y hacer el seguimiento.

1.2 Consenso

La Real Academia de la Lengua (RAE 2002) define el término consenso como:

“Acuerdo producido por consentimiento entre todos los miembros de un grupo o entre varios grupos”

A fines del presente trabajo se define consenso como un acuerdo entre dos o más personas en torno a un tema. La toma de decisiones por consenso implica plantear varias opciones y se elige la opción más popular en el grupo, por lo que todos los miembros del grupo se reúnen para discutir las propuestas y buscan llegar a un acuerdo que todos acepten. No quiere esto decir que todos estén de acuerdo en la totalidad de los puntos, sino que existe un criterio general unificado, a modo de común denominador.

1.2.1 Proceso de logro de consenso (PLC)

El consenso es una importante área de investigación con múltiples aplicaciones, entre ellas la toma de decisiones en grupo y el aprendizaje en equipo (3; 4). El proceso de construcción del consenso se ha definido como un proceso iterativo y dinámico consistente en varias rondas en las que los expertos presentan sus puntos de vista, coordinados por un moderador, quien calcula el consenso entre los expertos en cada ronda de acuerdo a métricas de consenso y ayuda a acercar las preferencias de los expertos en cada ronda (3; 5).

Según Mata (6) el proceso de consenso puede representarse según muestra la Figura 1. En este proceso se representa al consenso previo a la toma de decisiones en grupo teniendo como entrada el problema en cuestión y las alternativas de solución aportadas por los expertos, un proceso de exposición de preferencias sobre el conjunto de alternativas por parte de los individuos involucrados sujetas a cambios a partir de las recomendaciones emitidas por el moderador en cada ronda a fin de acercar las preferencias a un acuerdo común, y finalmente como salidas las opiniones consensuadas que servirán como base estable a la toma de decisiones o selección de la alternativa(s) solución.

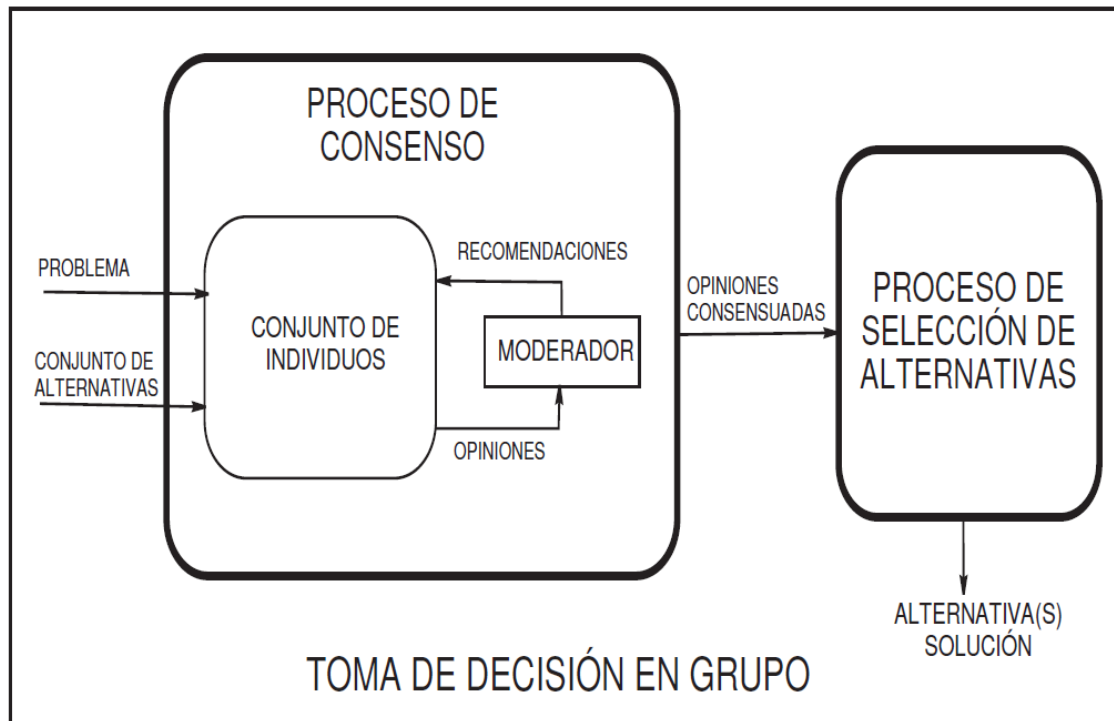


Fig. 1: Proceso de consenso y selección en problemas de toma de decisiones en grupo (6).

Se han identificado una serie de actividades que contribuyen al logro del consenso en cada ronda. Entre ellas se incluye: obtención de preferencias de los expertos, cálculo del grado de consenso, control del PLC, reducción de la búsqueda de las preferencias y producción de recomendaciones (5).

Para llegar al consenso, debemos considerar una serie de pasos (2):

1. Considerar todos los aspectos del problema.
2. Analizar las objeciones a las diversas alternativas de solución.
3. Aprovechar las diferencias de opinión bien intencionadas para:
 - Obtener información adicional.
 - Aclarar dudas.
 - Obligar al grupo a buscar mejores soluciones.



1.3 Mapas Cognitivos Difusos (MCD)

Los MCD (Figura 2) son una técnica introducida por Kosko (7) como una extensión de los mapas cognitivos (8). Mejoran los mapas cognitivos 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 (9). Constituyen una estructura de grafo difuso con retroalimentación para representar causalidad (10). En los MCD existen tres tipos posibles de relaciones causales entre los conceptos: causalidad positiva, causalidad negativa o la no existencia de relaciones.

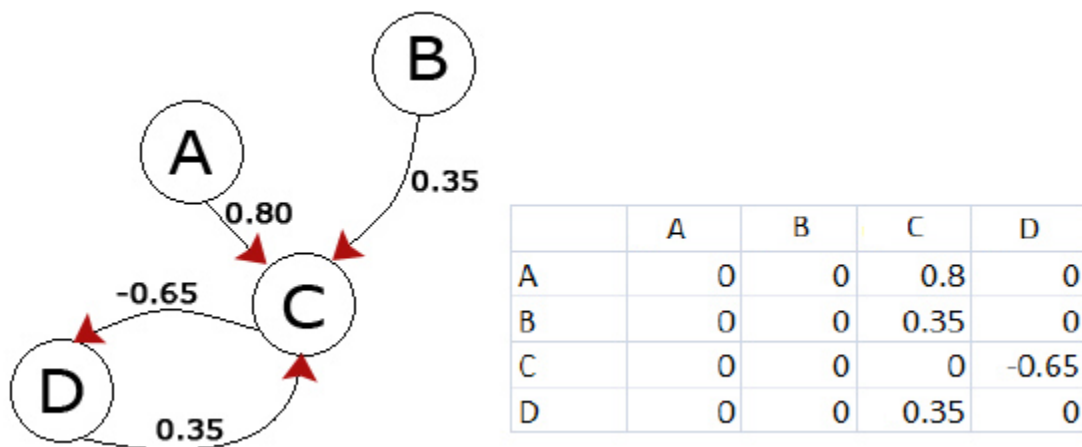


Fig. 2: Mapa cognitivo difuso y su correspondiente matriz de adyacencia.

1.3.1 MCD en la toma de decisiones en grupos

Los MCD tienen múltiples aplicaciones. Entre sus áreas de aplicación se destaca su utilización en la toma de decisiones (11; 12; 13; 14) y la construcción de sistemas de ayuda a la toma de decisiones (15; 16; 17).

Salmerón junto a otros colaboradores ha propuesto varios modelos para la toma de decisiones como es el caso de la selección de proyectos de tecnologías de la información (26) y el análisis de escenarios (20). Sin embargo están enfocadas a dominios específicos y/o no incluyen de forma sistémica todas las actividades en las cuales los MCD soportan la toma de decisiones.

Los MCD han sido empleados para la toma de decisión en grupo debido a las facilidades que brinda para la agregación de modelos causales provenientes de múltiples expertos (18; 19).



1.3.2 Consenso en MCD

Para problemas tales como el análisis de sistemas complejos y la toma de decisiones usualmente se hace necesario un proceso de consenso que permita lograr en el grupo un estado de acuerdo mutuo entre sus miembros (21; 5; 22).

A pesar del amplio uso de los MCD en dominios tan variados como las ciencias sociales y políticas, ingeniería, tecnologías de la información, robótica, medicina, entre otras, y el creciente número de publicaciones científicas en la pasada década dedicadas al tema, aún persisten insuficiencias en esta técnica (17).

Una de las limitaciones para su uso en el tratamiento de modelos mentales aplicado a la toma de decisiones, es el escaso tratamiento del proceso de consenso en MCD así como el inadecuado uso de valores numéricos para describir las relaciones entre conceptos, siendo más natural para las personas el uso de términos lingüísticos (23).

1.4 Modelos PLC para la toma de decisiones en grupo

En la presente sección se plantean los modelos usados en el Proceso de Logro del Consenso, mediante los MCD para la toma de decisiones en grupo, así como otros modelos propuestos.

1.4.1 Métodos usados en PLC para la toma de decisiones en grupo mediante el uso de MCD

En la literatura científica se reporta el uso de métodos de consenso en MCD fundamentalmente en el proceso de soporte a la toma de decisiones en grupo (24; 25; 26; 20). En estos trabajos identificados son aplicados el método Delphi (27) y el método propuesto por Bryson (21).

Método Delphi

El método Delphi fue desarrollado por Olaf Helmer y asociados a la Rand Corporation a principios de los 50 en investigaciones del departamento de defensa de los Estados Unidos. Luego amplió su uso como método de trabajo científico en ámbitos tan variados como la administración de empresas, planificación urbana y regional, educación, política, psicología, sociología entre otros. El proceso consta de seis pasos. Primeramente se identifican los miembros del grupo de expertos. Luego se le aplican cuatro cuestionarios, en cada uno de los cuales se trabaja en la convergencia de las preferencias de los expertos, y finalmente



se tabulan los resultados del cuarto cuestionario dando como resultado el consenso logrado por el grupo (27). Aunque no fue diseñado para ello, ha sido usado para la toma de decisiones en grupo mediante MCD (25; 26; 20).

En los estudios de la literatura y bibliografía consultada se reporta el uso del método Delphi en los MCD, pero no así los resultados obtenidos ni los conjuntos de datos utilizados. El método Delphi soporta la obtención de las preferencias y el cálculo del consenso. No ocurre así con la identificación automática de áreas de desacuerdo de manera que facilite la reducción de la búsqueda de las preferencias en próximas rondas y la producción de recomendaciones. Esto supone la imposibilidad de automatizar la ejecución de las actividades del PLC basado en el Método Delphi (29). En la Tabla 1 se compara con el método propuesto por Bryson.

Método propuesto por Bryson

Es un método más reciente, desarrollado específicamente para la toma de decisiones en grupo mediante MCD (21). Solo se identifica un trabajo que propone su utilización para lograr el consenso en la toma de decisiones en grupo mediante MCD (20) aunque no se reportan los resultados de su uso en la literatura.

El método de Bryson incluye actividades para la obtención de las preferencias y el cálculo del consenso. No ocurre así con la identificación automática de áreas de desacuerdo de manera que facilite la reducción de la búsqueda de las preferencias en próximas rondas y la producción automática de recomendaciones. Esto supone la imposibilidad de automatizar la mayor parte de las actividades del PLC basado en este método (29). En la Tabla 1 se compara con el método Delphi.



Tabla 1: Tabla comparativa entre los modelos de PLC Delphi y Bryson (29).

Modelos	Obtención	Cálculo cg	Reducción búsqueda	Producción recomendaciones	Uso en MCD para TDG*	Herramienta
Delphi	Si, aunque no adaptado a las características de los MCD para la TDG* ni hace uso de CWW**	Si, aunque no adaptado a las características de los MCD para la TDG* ni hace uso de CWW**	No	No	Se reporta su utilización no así <u>resultados</u> obtenidos ni conjuntos de datos utilizados	Parcialmente
Bryson	Si, aunque no hace uso de CWW**	Si, aunque no hace uso de CWW**	No	No	Se reporta su utilización no así resultados obtenidos ni conjuntos de datos utilizados	No

* Toma de decisiones en grupo

** Computación con palabras

1.4.2 Modelo Ágora

Según Pérez-Teruel (29) a partir de las fortalezas e insuficiencias detectadas, se desarrolló un modelo de proceso para el logro de consenso adaptado a las características de los MCD a fin de soportar con mayor fiabilidad la toma de decisiones en grupo y capaz de buscar el consenso de manera automática, sin la necesidad de la intervención de la figura humana en el rol de moderador en cada una de las rondas del proceso. Se denominó de forma resumida al modelo como Ágora, en alusión a las plazas de las ciudades-estado griegas, centro de la vida cultural, política y social de los antiguos griegos.

A tal razón Pérez-Teruel propone el Modelo Ágora que cuenta con estructura, componentes, cualidades que emergen de él y principios que lo guían. En la Figura 2 se representan gráficamente los elementos



anteriormente mencionados en relación con el modelo Ágora. En la Figura 3 se presenta su modelo de PLC. Luego se expone la descripción textual de cada una de sus actividades detallando el modelo matemático que soporta el proceso.

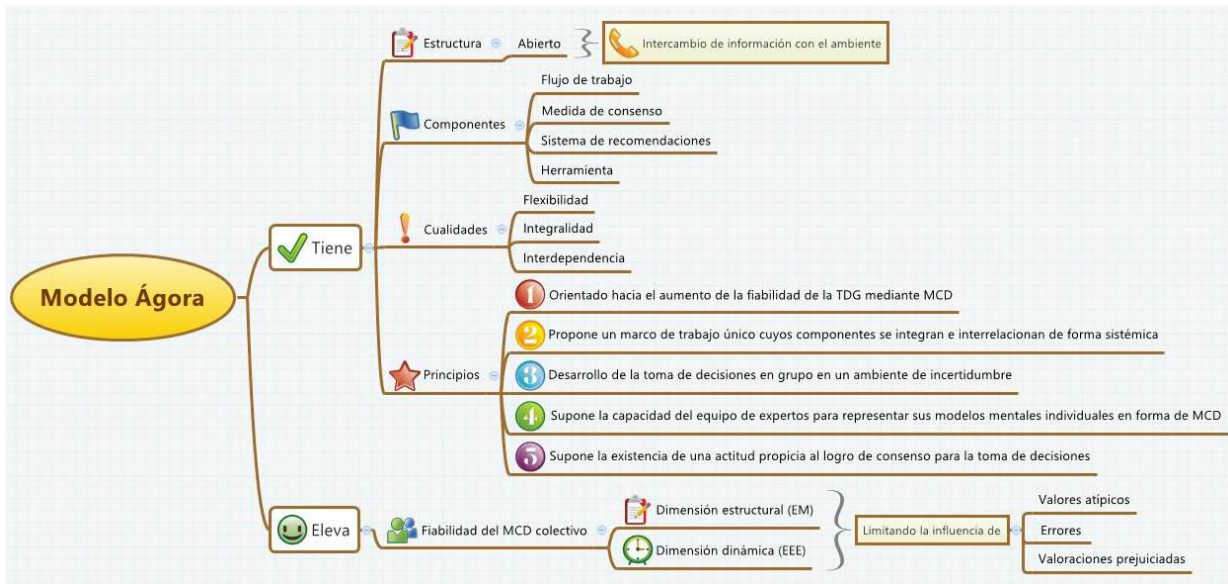


Fig. 3: Estructura, componentes, cualidades y principios del modelo Ágora.

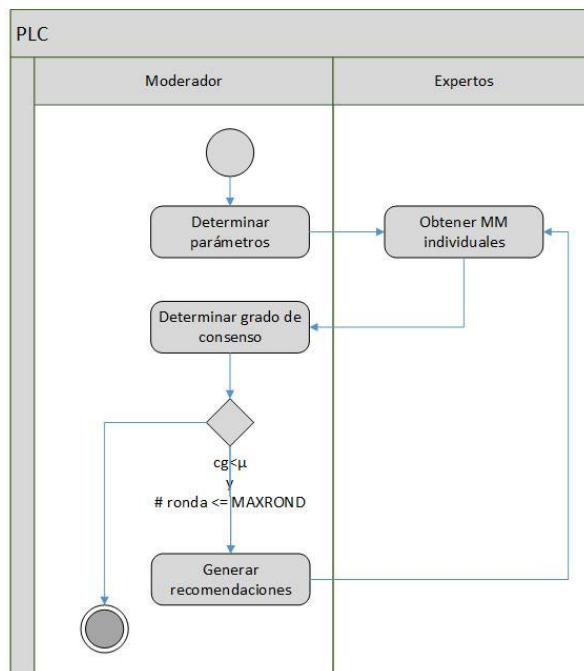


Fig. 4: Modelo PLC Ágora



Actividades detalladas del modelo matemático que soporta PLC Ágora (61):

1. Determinar parámetros: El moderador del PLC determina los nodos que desean ser analizados, los expertos que participarán y la granularidad del conjunto de términos lingüísticos. Determina además los parámetros para el control del consenso. Estos son: umbral de consenso $\mu \in [0,1]$, número máximo de rondas $MAXROND \in \mathbb{N}$ y el umbral de aceptabilidad $\varepsilon \geq 0$.
2. Determinar los modelos mentales individuales: Cada experto expresa sus modelos mentales haciendo uso de los términos lingüísticos definidos. El peso de la conexión que va del concepto C_i al concepto C_j , dado por el experto k, es representado mediante 2-tuplas lingüísticas del siguiente modo:

$$w_{ij}^k = (s_u, \alpha)_{ij}^k$$

3. Determinar grado de consenso: El grado de consenso es determinado en el intervalo [0,1]. Por cada relación causal definida por los expertos, su correspondiente valor β , el cual denotaremos como β_{ij}^k , es calculado de la siguiente forma:

$$\beta_{ij}^k = \Delta^{-1}(s_u, \alpha)_{ij}^k = u$$

siendo Δ^{-1} la función de transformación mostrada en (2).

Por cada par de expertos $e_k, e_t, (k < t)$ se calcula una matriz de similitud $SM_{kt} = (sm_{ij}^{kt})^{n \times n}$, $sm_{ij}^{kt} \in [0,1]$ calculada de la siguiente forma:

$$sm_{ij}^{kt} = 1 - \frac{|\beta_{ij}^k - \beta_{ij}^t|}{g}$$

La matriz de consenso $CM = (cm_{ij})^{n \times n}$, que representa la similitud entre todas las relaciones causales dadas por los expertos y contiene la información a partir de la cual se va a calcular el grado de consenso, es obtenida de la siguiente forma:

$$cm_{ij} = OAG_1(SIM_{ij})$$



donde OAG_1 es un operador de agregación $SIM_{ij} = \{sm_{ij}^{12}, \dots, sm_{ij}^{1m}, \dots, sm_{ij}^{(m-1)m}\}$ y representa la similitud en las opiniones de los expertos con respecto a la relación causal existente entre (C_i, C_j) y cm_{ij} es el grado de consenso alcanzado por el grupo de expertos con respecto a la relación causal existente entre (C_i, C_j) .

El grado de consenso cn_i con respecto a la influencia que ejerce el nodo C_i (en caso de no permitirse la autoconexión) es calculado de la siguiente forma:

$$cn_i = \frac{\sum_{j=1, j \neq i}^n cm_{ij}}{n-1}$$

Finalmente el grado de consenso general es calculado

$$cg = \frac{\sum_{i=1}^n cn_i}{n}$$

4. Control del consenso: El grado de consenso general es comparado con el umbral de consenso (μ). Si $cg < \mu$ el proceso de consenso finaliza. Si no es así indica que se requiere mayor discusión adicional. El parámetro MAXROND también limita el número máximo de rondas de discusión posible.
5. Generación de recomendaciones: En esta fase se emplea un mecanismo automático capaz de realizar el asesoramiento a los expertos con el fin de ayudarles a cambiar sus preferencias en la dirección adecuada. Se inicia calculando las medidas de proximidad a partir del MCD colectivo $W^c = (w_{ij}^c)^{n \times n}$, $w_{ij}^c \in S \times [-0.5, 0.5]$. Este valor es calculado agregando los MCD individuales:

$$w_{ij}^c = (s_u, \alpha)_{ij}^k = OAG_2((s_u, \alpha)_{ij}^1, \dots, (s_u, \alpha)_{ij}^m)$$

donde $u \in S$, y OAG_2 es un operador de agregación sobre 2-tuplas lingüísticas. Posteriormente se calcula la matriz de proximidad PP_k , entre cada uno de e_k expertos y W^c . Los valores de proximidad $pp_{ij}^k \in [0,1]$ son calculados de la siguiente forma:

$$pp_{ij}^k = 1 - \frac{|\beta_{ij}^k - \beta_{ij}^c|}{g}$$



donde $\beta_{ij}^c = \Delta^{-1}((s, \alpha)_{ij}^c)$.

Se identifican las relaciones causales a cambiar (CC). Para esto se identifican las relaciones en los conceptos C_i y C_j cuyo grado de consenso sea menor al umbral definido (μ):

$$CC = \{w_{ij}^c | cn_i < \mu \wedge cm_{ij} < \mu\}$$

Una vez determinadas las relaciones que deben ser cambiadas se determinan los expertos que deben actualizar las relaciones. Con este fin se agregan las medidas de proximidad:

$$pp_{ij}^A = OAG_2(pp_{ij}^1, \dots, pp_{ij}^m)$$

OAG_2 es un operador de agregación sobre 2-tuplas lingüísticas.

A los expertos e_k , cuyas $pp_{ij}^k < pp_{ij}^A$ se les recomienda cambiar las relaciones causales w_{ij}^k .

Finalmente se recomienda la dirección en que se debe cambiar la valoración. Para ello se emplea el umbral de aceptabilidad ε para evitar generar un excesivo número de recomendaciones innecesarias.

R 1: Si $\beta_{ij}^k - \beta_{ij}^c < -\varepsilon \wedge w_{ij}^k \neq s_{g/2}$ entonces e_k debe variar la etiqueta lingüística S_u asignada a la relación causal w_{ij} por otra S_w de manera que $u < w$.

R 2: Si $\beta_{ij}^k - \beta_{ij}^c > \varepsilon \wedge w_{ij}^k \neq s_{g/2}$ entonces e_k debe variar la etiqueta lingüística S_u asignada a la relación causal w_{ij} por otra S_w de manera que $u > w$.

R 3: Si $\beta_{ij}^k - \beta_{ij}^c < -\varepsilon \wedge w_{ij}^k = s_{g/2}$ entonces e_k debe crear una conexión con causalidad positiva.

R 4: Si $\beta_{ij}^k - \beta_{ij}^c > \varepsilon \wedge w_{ij}^k = s_{g/2}$ entonces e_k debe crear una conexión con causalidad negativa.



R 5: Si $-\varepsilon \leq \beta_{ij}^k - \beta_{ij}^c \leq \varepsilon$ entonces e_k no debe cambiar el valor de la relación causal W_{ij} .

1.5 Metodologías para el desarrollo del software

El desarrollo y la implementación de software incluyen un reto para los desarrolladores, pues crear un software en el menor tiempo posible y con calidad resulta difícil, sino se tiene en cuenta algún proceso que ayude a agilizar el desarrollo de los mismos. Para resolver este problema y lograr la obtención de un producto de calidad que a su vez cumpla con los requerimientos de los clientes, se hace necesario el uso de las metodologías de desarrollo de software. Estas imponen un proceso disciplinado en el que se ve paso a paso y detalladamente las tareas y actividades que se deben realizar para obtener un producto informático con excelente calidad.

1.5.1 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. (30).

La selección de esta metodología está basada en los siguientes parámetros:

- ✓ Es la que mejor se ajusta a las necesidades del proyecto en cuanto a recursos técnicos, tiempo de desarrollo y personal.
- ✓ El equipo de desarrollo es pequeño y el cliente forma parte constante de él, mitiga el riesgo, aumenta la probabilidad de éxito.
- ✓ Es adecuada para proyectos con requisitos imprecisos y que pueden cambiar constantemente.
- ✓ Permite que en cualquier fase del proceso de desarrollo de la aplicación, se pueda regresar a operaciones ya realizadas sin afectar el ciclo de vida del software.



1.6 Herramientas de Lenguaje y Modelado

1.6.1 Herramienta CASE Visual Paradigm

Las herramientas CASE (Ingeniería de Software Asistida por Computadora) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero. (32)

Visual Paradigm es una herramienta CASE que utiliza “UML”: como lenguaje de modelado. Se caracteriza principalmente por su robustez, usabilidad y portabilidad.

Esta herramienta permite realizar ingeniería tanto directa como inversa, a partir de un modelo relacional en SQLServer, 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. Es libre, multiplataforma y se tiene su licencia en nuestra universidad. Además soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue.

1.6.2 Lenguajes de programación del 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 (El World Wide Web Consortium) 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 (55).



JavaScript:

JavaScript (JS) es un lenguaje de programación interpretado. Se define como orientado a objetos (56), 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 (57).

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 (58).

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 (58).

1.6.3 Lenguajes de programación del lado del servidor

PHP:

Es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML (HyperText Markup Language). Con PHP se puede hacer cualquier cosa que podemos realizar con un script, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas. Una de sus características más potentes es su soporte para gran cantidad de bases de datos. Entre su soporte pueden mencionarse InterBase, mSQL, MySQL, Oracle, Informix, PostgreSQL, entre otras. PHP también ofrece la integración con las varias bibliotecas externas, que permiten que el desarrollador haga casi cualquier cosa desde generar documentos en pdf hasta analizar código XML. PHP ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación. Su diseño elegante lo hace perceptiblemente más fácil de mantener y ponerse al día que el código comparables en otros lenguajes. Debido a su amplia distribución PHP está perfectamente soportado por una gran comunidad de desarrolladores. (33)



PHP provee al producto varias ventajas entre ellas se encuentran (34):

- Lenguaje totalmente libre y abierto
- Curva de aprendizaje muy baja
- Los entornos de desarrollo son de rápida y fácil configuración
- Fácil despliegue: paquetes totalmente autoinstalables que integran PHP
- Fácil acceso a bases de datos
- Comunidad muy grande

1.6.4 Framework de desarrollo Yii

Un marco de trabajo (framework) es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Son diseñados con la intención de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel de proveer un sistema funcional.

Yii es un framework PHP, libre (licencia BSD), basado en programación orientada a objetos y que implementa el patrón de diseño MVC (Modelo-Vista-Controlador) el cuál es ideal para la programación Web. Fue diseñado por el creador de Prado, por lo que ha evitado los errores cometidos en la implementación de este framework. (35)

Puede ser utilizado para todo tipo de aplicaciones Web. Es adecuado para desarrollar aplicaciones de gran tráfico como portales, foros, sistemas de administración de contenidos (CMS). (35).

Algunas de las características con que cuenta este framework son (42):

- Patrón de diseño Modelo Vista Controlador (MVC).
- Database Access Objects (DAO), query builder, Active Record y migración de base de datos.
- Integración con jQuery.
- Entradas de Formulario y validación.
- Widgets de Ajax, como autocompletado de campos de texto y demás.



- Soporte de Autenticación incorporado. Además soporta autorización vía role-based access control (RBAC) jerárquico.
- Personalización de aspectos y temas.
- Generación compleja automática de WSDL, especificaciones y administración de peticiones Web service.
- Internacionalización y localización (I18N and L10N). Soporta traducciones, formato de fecha y hora, formato de números, y localización de la vista.
- Esquema de caching por capas. Soporta el cache de datos, cache de páginas, cache por fragmentos y contenido dinámico. El medio de almacenamiento del cache puede ser cambiado.
- El manejo de errores y logging. Los errores son manejados y personalizados, y los log de mensajes pueden ser categorizados, filtrados y movidos a diferentes destinos.
- Las medidas de seguridad incluyen la prevención cross-site scripting (XSS), prevención cross-site request forgery (CSRF), prevención de la manipulación de cookies, etc.
- Herramientas para pruebas unitarias y funcionales basados en PHPUnit y Selenium.
- Generación automática de código para el esqueleto de la aplicación, aplicaciones CRUD, etc.
- Generación de código por componentes de Yii y la herramienta por línea de comandos cumple con los estándares de XHTML.
- Cuidadosamente diseñado para trabajar bien con código de terceros. Por ejemplo, es posible usar el código de PHP o Zend Framework en una aplicación Yii.

1.6.5 Framework de CSS Bootstrap

Bootstrap es el framework de Twitter que permite crear interfaces web con CSS y Javascript que adaptan la interfaz dependiendo del tamaño del dispositivo en el que se visualice de forma nativa, es decir, automáticamente se adapta al tamaño de un ordenador o de una Tablet sin que el usuario tenga que hacer nada, esto se denomina diseño adaptativo o Responsive Design.

Aun ofreciendo todas las posibilidades que ofrece Bootstrap a la hora de crear interfaces web, los diseños creados con Bootstrap son simples, limpios e intuitivos, esto les da agilidad a la hora de cargar y al



adaptarse a otros dispositivos. El Framework trae varios elementos con estilos predefinidos fáciles de configurar: Botones, Menús desplegables, Formularios incluyendo todos sus elementos e integración jQuery para ofrecer ventanas y tooltips dinámicos. (43)

1.6.6 Sistema Gestor de Base de Datos PostgreSQL

Un Sistema Gestor de Base de Datos es un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan, y controlar la entrada y salida de datos de la base de datos, manteniendo la integridad de la misma. La importancia de su uso es que proporcionan a los usuarios la capacidad de almacenar datos en la base de datos, acceder a ellos y actualizarlos. (36)

PostgreSQL es un sistema objeto-relacional (ORDBMS), de código abierto, que utiliza el lenguaje SQL92/SQL99. Incluye características del paradigma orientado a objetos, como la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. Soporta distintos tipos de datos como: datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes y cadenas de bits; además de permitir la creación de tipos propios e incorporar una estructura de datos arreglos (array), funciones como: manejo de fechas, funciones geométricas y funciones orientadas a operaciones con redes, y también permite la declaración de funciones propias, así como la definición de disparadores. Soporta el uso de índices, reglas y vistas e incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales y por último permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos (36).

1.7 Entorno Integrado de Desarrollo

NetBeans es un IDE que está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el IDE NetBeans. Es un producto libre y gratuito sin restricciones de uso de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles). Entre sus características se encuentra un sistema de proyectos basado en Ant, control de versiones y refactoring. (37)



CONSIDERACIONES PARCIALES

- ✓ Se realizó un estudio sobre los métodos PLC existentes en la bibliografía consultada quedando claro la no existencia de un método que cuente con todos los requisitos necesarios para la aplicación del proceso de consenso mediante el uso de MCD.
- ✓ Se utiliza la metodología XP pues el tiempo de desarrollo del software es relativamente corto y el equipo de trabajo es pequeño.
- ✓ Se seleccionaron una serie de tecnologías y herramientas para el desarrollo del módulo como fueron: el Visual Paradigm como herramienta CASE, el lenguaje de programación PHP del lado del servidor conjuntamente con el framework de desarrollo Yii y Bootstrap, además del NetBeans como entorno de desarrollo y PostgreSQL como gestor de base de datos.



CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA APLICACIÓN.

El presente capítulo abordará sobre el diseño del sistema para el proceso de consenso en la red social Nexo, además de comprender el contexto e identificar las condiciones o capacidades que debe cumplir. También todo lo referente a las fases de Exploración, Planificación e Iteraciones de la metodología de desarrollo utilizada. Se describen los artefactos obtenidos según la metodología. Se realiza el modelo de datos y se especifican los requisitos funcionales y no funcionales que deberá cumplir la solución. Además se describe la arquitectura de software y los patrones de diseños utilizados.

2.1 Fase de exploración

La fase de exploración es la primera de la metodología XP. 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, pues son la forma en que se especifican los requisitos del sistema.

2.1.1 Historias de Usuario

Las 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 (44). De 13 HU totales a continuación se muestran solo 5 HU.

Tabla 2: HU1 Establecer Parámetros del Consenso.

Historia de Usuario	
Número: 1	Nombre de Historia de Usuario: Establecer Parámetros del Consenso.
Modificación de Historia de Usuario Número: 0	
Usuario: Todos.	Iteración Asignada: 2
Prioridad en Negocio: Media	Puntos Estimados: 1
Riesgo en Desarrollo: Medio	Puntos Reales: 1
Descripción: Inicia cuando el usuario decide registrar los parámetros para el	



consenso de uno de sus Proyectos.

Observaciones: Para poder insertar estos parámetros ya el usuario debe de tener creado algún proyecto que sea de tipo Privado.

Prototipo de Interfaz:

Campos con * son requeridos.

¡Hey! Te aconsejamos que los valores para los campos Umbral y Exilon deberían estar entre un intervalo de 0 a 1, y para el Total de Rondas es aconsejable el valor 10. ¡Suerte!

Umbral *

Exilon *

Total Rondas *

Crear

Tabla 3: Listar Parámetros del Consenso.

Historia de Usuario	
Número: 3	Nombre de Historia de Usuario: Listar Parámetros del Consenso.
Modificación de Historia de Usuario Número: 0	
Usuario: Todos.	Iteración Asignada: 2



Prioridad en Negocio: Alta	Puntos Estimados: 1																								
Riesgo en Desarrollo: Medio	Puntos Reales: 0.5																								
Descripción: Inicia cuando el usuario decide listar los Proyectos con Consenso y a la vez poder eliminarlos o visualizar sus datos.																									
Observaciones: Deben existir Proyectos con Consensos creados para que estos puedan ser listados.																									
Prototipo de Interfaz:																									
<table border="1"> <thead> <tr> <th>Proyecto</th> <th>Umb.</th> <th>Exi.</th> <th>Tot. Rondas</th> <th>Ronda Act.</th> <th>Acciones</th> </tr> </thead> <tbody> <tr> <td>Aprendizaje Online</td> <td>0.5</td> <td>0.56</td> <td>8</td> <td>2</td> <td> Ir al Consenso x Eliminar </td> </tr> <tr> <td>Productos Bio-Quimicos</td> <td>0.5</td> <td>0.56</td> <td>20</td> <td>14</td> <td> Ir al Consenso x Eliminar </td> </tr> <tr> <td>Proyecto para Consenso</td> <td>0.5</td> <td>0.5</td> <td>6</td> <td>6</td> <td> Ir al Consenso x Eliminar </td> </tr> </tbody> </table>		Proyecto	Umb.	Exi.	Tot. Rondas	Ronda Act.	Acciones	Aprendizaje Online	0.5	0.56	8	2	Ir al Consenso x Eliminar	Productos Bio-Quimicos	0.5	0.56	20	14	Ir al Consenso x Eliminar	Proyecto para Consenso	0.5	0.5	6	6	Ir al Consenso x Eliminar
Proyecto	Umb.	Exi.	Tot. Rondas	Ronda Act.	Acciones																				
Aprendizaje Online	0.5	0.56	8	2	Ir al Consenso x Eliminar																				
Productos Bio-Quimicos	0.5	0.56	20	14	Ir al Consenso x Eliminar																				
Proyecto para Consenso	0.5	0.5	6	6	Ir al Consenso x Eliminar																				

Tabla 4: HU4 Calcular Consenso.

Historia de Usuario	
Número: 4	Nombre de Historia de Usuario: Calcular Consenso.
Modificación de Historia de Usuario Número: 0	
Usuario: Todos.	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 1.5
Riesgo en Desarrollo: Alto	Puntos Reales: 1.5
Descripción: Inicia cuando el usuario decide comenzar el proceso de Consenso para el proyecto que tenga	



seleccionado.

Observaciones: Para poder calcular el consenso el usuario con anterioridad debe de entrar al sistema los parámetros para dicho consenso.

Prototipo de Interfaz:

The screenshot shows a user interface for monitoring a process. At the top, there are four navigation options: 'Consenso' (star icon), 'Editar Parámetros' (pencil icon), 'Enviar Recomendaciones' (gift icon), and 'Monitorear Recomendaciones' (eye icon). Below this is an 'Información' button. The main content area features a red location pin icon above the text 'Estado del Proceso', followed by 'Inteligencia Artificial' and a large green progress bar indicating '¡0.00 % completado!'. Below the progress bar, there are two statistics: 'Recomendaciones Realizadas' with a bell icon and the number '0', and 'Recomendaciones sin Atender' with a bell icon and the number '0'. At the bottom center, there is a blue button with a checkmark icon and the text '¡Iniciar el Proceso!'.

Tabla 5: HU5 Generar Recomendaciones.

Historia de Usuario	
Número: 5	Nombre de Historia de Usuario: Generar Recomendaciones.
Modificación de Historia de Usuario Número: 0	
Usuario: Todos.	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 1.5
Riesgo en Desarrollo: Alta	Puntos Reales: 1.5
Descripción: Inicia cuando el usuario decide enviar las recomendaciones sobre consenso realizadas por	



el sistema a los participantes para poder lograr el consenso en el proyecto actual.

Observaciones: El usuario debe seleccionar un proyecto para poder observar las recomendaciones y gestionarlas.

Prototipo de Interfaz:

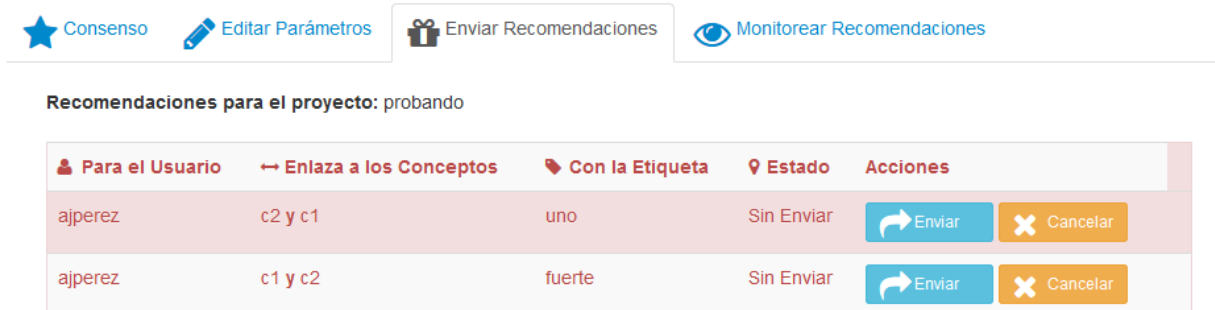


Tabla 6: HU6 Notificar Recomendación.

Historia de Usuario	
Número: 6	Nombre de Historia de Usuario: Notificar Recomendación.
Modificación de Historia de Usuario Número: 0	
Usuario: Todos.	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 1.5
Riesgo en Desarrollo: Alta	Puntos Reales: 1.5
Descripción: Inicia cuando el usuario es notificado para que decida cambiar o no la selección realizada en el MCD del proyecto en el que participa.	
Observaciones: El usuario puede o no aceptar la recomendación.	
Prototipo de Interfaz:	



Responde a tus recomendaciones

probando

Recomendación	Nueva Etiqueta	¿Qué piensas hacer?
En el proyecto probando debe cambiar la etiqueta uno que enlaza a los conceptos c2 y c1 por una nueva etiqueta que Disminuya el valor actual.	<input type="text" value="uno"/>	<input type="button" value="Cambiar"/> <input type="button" value="Rechazar"/>
En el proyecto probando debe cambiar la etiqueta fuerte que enlaza a los conceptos c1 y c2 por una nueva etiqueta que Disminuya el valor actual.	<input type="text" value="uno"/>	<input type="button" value="Cambiar"/> <input type="button" value="Rechazar"/>

2.2 Fase de 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. (12; 59)

2.2.1. Lista de Reservas del Producto

La lista de reserva del producto es un artefacto que recoge los requisitos funcionales (RF) y no funcionales (RNF) de una aplicación.

2.2.1.1. Requisitos Funcionales

RF1: Crear Parámetros para el Consenso.

RF2: Modificar Parámetros para el Consenso.

RF3: Eliminar Parámetros para el Consenso.

RF4: Listar Parámetros para el Consenso.

RF5: Calcular el Consenso de un Proyecto.

RF6: Generar Recomendaciones a los usuarios participantes de un Consenso.

RF7: Monitorear las Recomendaciones hechas a los usuarios participantes de un Consenso.



RF8: Permitir modificar los MCD individuales.

RF9: Notificar a los usuarios sobre recomendaciones realizadas por el Sistema en cuanto al Consenso.

RF10: Permitir filtrar por proyectos las búsquedas más importantes.

RF11: Permitir que el sistema muestre la matriz de Consenso referente a un consenso en específico.

RF12: Permitir que el sistema muestre la matriz de Similaridad referente a un consenso en específico.

RF13: Permitir que el sistema muestre la matriz de Preferencia referente a un consenso en específico.

RF14: Permitir que el sistema muestre la matriz de Proximidad referente a un consenso en específico.

RF15: Permitir que el sistema muestre la matriz de Direcciones referente a un consenso en específico.

2.2.1.2. Requisitos no Funcionales

Usabilidad:

- ✓ El módulo tendrá la información de forma estructurada mediante el uso de títulos y subtítulos que le permitirán al usuario saber a dónde dirigirse para encontrar lo que desea. Los botones y enlaces serán fáciles de identificar en todas las operaciones que el usuario tenga que realizar.

Confiabilidad:

- ✓ En cualquiera situación o evento presentado, el sistema mostrara siempre mensajes informativos.

Restricciones de diseño:

- ✓ Lenguaje de programación a utilizar PHP bajo el IDE NetBeans. Utilización de los Frameworks Yii y Bootstrap y como gestor de Bases de Datos PostgreSQL.

Rendimiento:

- ✓ Los tiempos de respuesta y velocidad de procesamiento de la información serán rápidos, no mayores de 3 segundos para las actualizaciones y 10 para las recuperaciones.



Software:

- ✓ Para el cliente:
 - Utilizar navegador (Mozilla Firefox, Google Chrome, Internet Explorer).
 - Sistema operativo Windows o Linux.

- ✓ Para el servidor:
 - Sistema operativo Windows o Linux en cualquiera de sus distribuciones.
 - Servidor de base de datos PostgreSQL.
 - Servidor con módulo PHP disponible (WAMP, XAMPP, u otros).

Hardware:

- ✓ Para el servidor:
 - Capacidad de disco duro superior a 80 Giga bytes. Se requiere un mínimo de 2 GB de RAM y 2.1 GHz de velocidad de procesamiento.

- ✓ Para el cliente:
 - Utilizar Capacidad de disco duro superior a 50 Giga bytes. Se requiere un mínimo de 256 MB de RAM y 1.6 GHz de velocidad de procesamiento.

Seguridad:

- ✓ Los usuarios deben autenticarse antes de interactuar con el sistema.

- ✓ 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.

Interfaz:

- ✓ La aplicación contará con una 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.

2.2.2. Estimación del esfuerzo por Historia de Usuario

De acuerdo a la prioridad asignada por el cliente a cada HU y teniendo en cuenta la complejidad y riesgo determinado por el programador, se realiza la estimación de cada una de las HU identificadas, los resultados de la estimación se muestran en la Tabla 7. La unidad de estimación es el punto, un punto equivale a una semana ideal de programación.

Tabla 7: Estimación de esfuerzo por HU

#	Historia de Usuario	Puntos de Estimación
1	Establecer Parámetros del Consenso	1
2	Modificar Parámetros del Consenso	1
3	Listar y Eliminar Parámetros del Consenso	1
4	Calcular Consenso	1.5
5	Generar Recomendaciones	1.5
6	Notificar Recomendaciones	1.5
7	Monitorear Recomendaciones	1
8	Mostrar Matriz de Consenso	0.5
9	Mostrar Matriz de Similitud	0.5
10	Mostrar Matriz de Preferencia	0.5
11	Mostrar Matriz de Proximidad	0.5
13	Mostrar Matriz de Direcciones	0.5



2.2.3. Plan de Iteraciones

Después de identificadas las HU y la estimación del esfuerzo por cada una de ellas, se procede a realizar el plan de iteraciones, donde estarán almacenadas las HU en el orden a realizar por cada iteración según su orden, así como la descripción y el total de semanas que durarán cada una de estas. A continuación se muestra una tabla con el plan de iteraciones:

Tabla 8: Plan de Iteraciones.

Iteración	Descripción de la iteración	Orden de la HU a implementar	Duración total
1	La primera iteración tiene como objetivo llevar a cabo la implementación de las HU de prioridad media, todas las funcionalidades son indispensables pero algunas son de un menor nivel de importancia, además existen pocas HU con esta categoría.	1 y 7	14 días
2	Esta iteración es la encargada de implementar las HU de prioridad alta, debido a la cantidad de HU con esta prioridad se decidió dejarla para esta fase.	2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13	63 días

2.2.4. Plan de Entrega

A partir del plan de iteraciones anterior se decide 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.

Tabla 9: Plan de Entrega.

Iteración	Iteración # 1	Iteración # 2
Cantidad de HU	2	11
Fecha de entrega	14/2/2015	1/5/2015



2.3 Fase de Iteraciones

Esta es la fase principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas en esta fase, generando al final de cada una un entregable funcional que cumpla con lo planteado en las HU asignadas a la iteración. (44) Asociado a cada iteración de la fase anterior se encuentra la planificación de las tareas de ingeniería, donde cada HU se transforma en estas tareas que son desarrolladas por programadores. La distribución de tareas en correspondencia con las HU que se desarrollaron se pueden encontrar en el capítulo de Implementación y Pruebas.

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).

2.3.1 Tarjetas CRC

Las tarjetas CRC 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.

Tabla 10: Tarjeta CRC ConsensoController.

TARJETA CRC	
Clase: ConsensoController	
Responsabilidades	Colaboraciones
<ul style="list-style-type: none">• Registrar Parámetros• Modificar Parámetros• Eliminar Parámetros• Listar Parámetros• Seleccionar Proyecto	Proyecto, Usuario.



Tabla 11: Tarjeta CRC RecomendacionController.

TARJETA CRC	
Clase: RecomendacionController	
Responsabilidades	Colaboraciones
<ul style="list-style-type: none">• Monitorear Recomendaciones• Realizar Recomendaciones• Notificar Recomendaciones• Actualizar Etiqueta• Actualizar Estado• Rechazar Recomendación• Cancelar Recomendación	Proyecto, Etiqueta, Concepto, Usuario.

2.4 Arquitectura de Software

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. Según el estándar IEEE (*Institute of Electrical and Electronics Engineers*).

2.4.1. Patrón Modelo Vista Controlador (MVC)

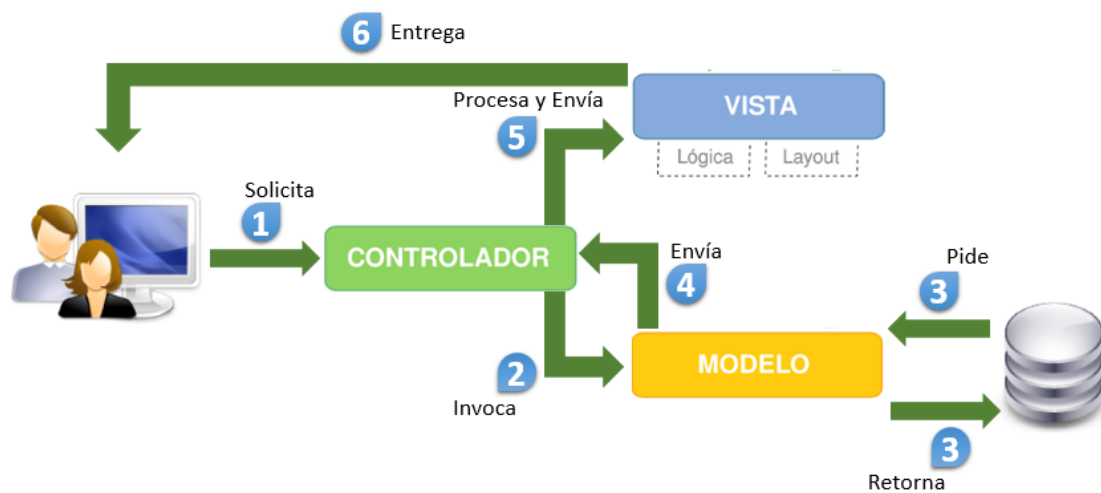


Fig. 5: Funcionamiento del Patrón Modelo Vista Controlador de Yii. (MVC).



MVC es un patrón de diseño que separa de manera clara y precisa los tres componentes de una aplicación: el **modelo**, la **vista** y el **controlador**. Su objetivo principal es el de separar la lógica del negocio de la lógica de la presentación para darle estructura a la implementación y facilitar con esto su posterior mantenimiento. (45)

Como se mencionó anteriormente, el modelo consta de tres partes de acuerdo con la siguiente descripción: (45)

- El **modelo** representa la lógica de la aplicación, que se encuentra inmersa en los datos y en las reglas del negocio.
- La **vista** hace referencia a la interfaz de usuario, a la presentación de la información.
- El **controlador** actúa como mediador entre la solicitud del usuario y los modelos y vistas involucrados en la ejecución.

A continuación se presenta una muestra de la arquitectura en las tres capas.

Capa Modelo:

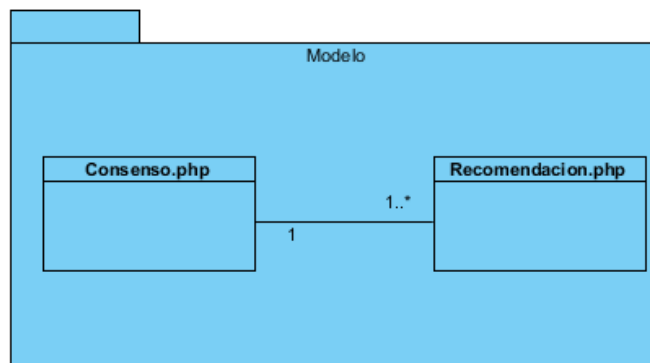


Fig. 6: Aplicación de la arquitectura en la capa Modelo.



Capa Vista:

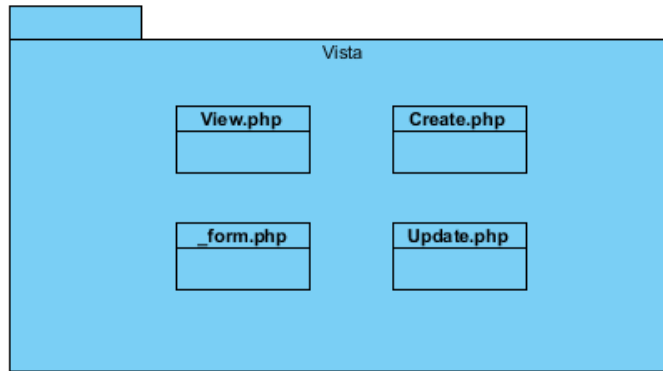


Fig. 7: Aplicación de la arquitectura en la capa Vista.

Capa Controlador:

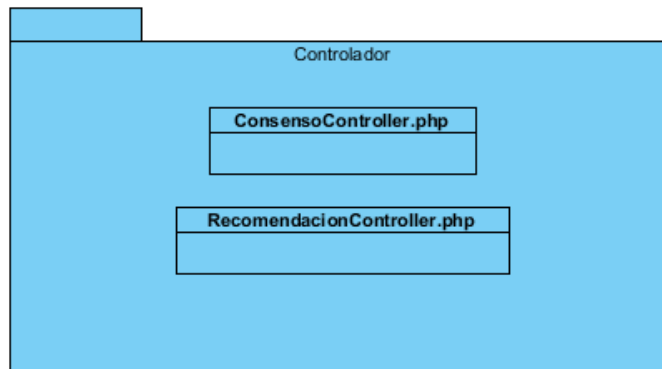


Fig. 8: Aplicación de la arquitectura en la capa Controlador.

2.5. Patrones de Diseño

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. (46) 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 remodelación 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. (47)

2.5.1. 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. Almacena o maneja varias instancias de la clase. (48)

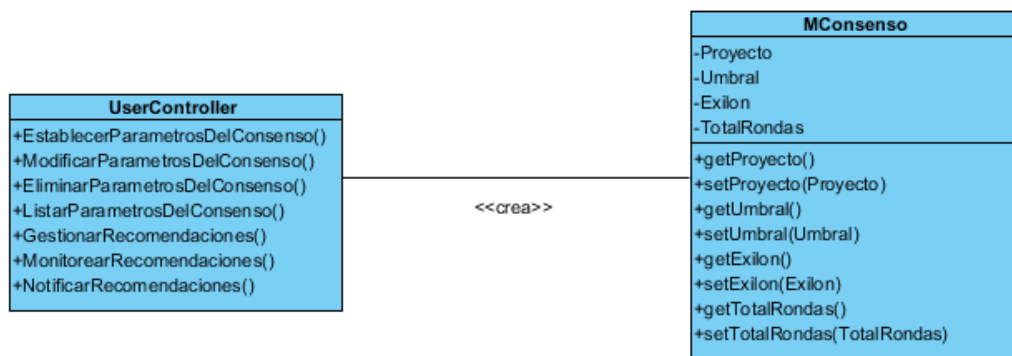


Fig. 9: Diagrama de clases que representa el Patrón Creador.

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. (48)

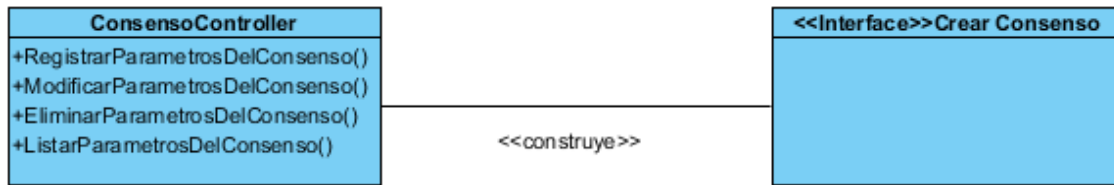


Fig. 10: Diagrama de clases que representa el Patrón Controlador.

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. (48)

Este patrón se puede evidenciar en la clase RecomendacionController ya que funciona como experta en información para llevar a cabo el trabajo que realiza todo el algoritmo del Consenso, por ejemplo es la responsable de capturar todos los datos de las recomendaciones y del consenso de un proyecto, necesarios para poder ejecutar el proceso de consenso.

Bajo Acoplamiento: el patrón bajo acoplamiento impulsa la asignación de responsabilidades de manera que su localización no incremente el acoplamiento hasta un nivel que lleve a los resultados negativos que puede producir un acoplamiento alto. No soporta el diseño de clases que son más independientes, lo que reduce el impacto del cambio. El mismo no se puede considerar de manera aislada a otros patrones como el Experto o el de Alta Cohesión, sino que necesita incluirse como uno de los diferentes principios de diseño que influyen en una elección, al asignar una responsabilidad. (48)

Las clases controladoras heredan únicamente de la clase Controller para alcanzar un bajo acoplamiento de clases. Las clases que implementan la lógica del negocio y de acceso a datos se encuentran en el modelo, las cuales no tiene asociaciones con las de la vista o el controlador, lo que proporciona que la dependencia en este caso sea baja.

Alta Cohesión nos indica que la información que almacena una clase debe de ser coherente y debe estar en la medida de lo posible relacionada con la clase. Para la implementación de este módulo se hizo uso



de las formas de organización que brinda el Framework Yii. El mismo permite la organización del trabajo en cuanto a la estructura del proyecto y la asignación de responsabilidades con una alta cohesión.

2.5.2 Patrones GOF

Los patrones Banda de los Cuatro o GOF (Gang of Four) son soluciones técnicas. Indican resoluciones técnicas basadas en Programación Orientada a Objetos (POO). En ocasiones tienen más utilidad con algunos lenguajes de programación y en otras son aplicables a cualquier lenguaje. (49)

Fachada: añade dinámicamente nuevas responsabilidades a un objeto, proporcionando una alternativa flexible a la herencia para extender la funcionalidad. (50)

Este patrón se observa cuando las vistas se separan en una plantilla base y varias plantillas que heredan de esta. Normalmente, la plantilla base es global en toda la aplicación y contiene el código HTML que es común a la mayoría de las páginas, para no tener que repetirlo en cada página. Lo cual es una implementación del patrón Decorador.

CONSIDERACIONES PARCIALES

- ✓ La definición de los requisitos permitieron identificar las funcionalidades con las que constará el sistema, las cuales darán respuesta a las necesidades del problema.
- ✓ Con la descripción detallada de las Historias de Usuario se logró un mayor acercamiento a lo que el sistema deberá cumplir.
- ✓ Las definiciones de las Tarjetas CRC aportaron en la identificación de las responsabilidades de las clases y las colaboraciones de las mismas.
- ✓ Los patrones de diseño aplicados conllevaron a lograr un diseño flexible y eficiente.



CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA.

Una vez concluido el análisis y diseño de la aplicación ya se cuenta con una visión más general de la arquitectura del sistema que se quiere obtener. En este capítulo se abordan la implementación del sistema perteneciente a la fase Iteraciones y las pruebas realizadas pertenecientes a la fase de Producción. Se identifican y describen las tareas de ingeniería y se llevan a cabo las pruebas de aceptación realizadas a cada una de las HU con el objetivo de validar que el sistema cumpla con lo esperado. Por otra parte se muestra el diagrama de despliegue del sistema a implementar.

3.1 Tareas de Ingeniería

Las Tareas de Ingeniería (TI) tienen como objetivo definir cada una de las actividades que dan cumplimiento a cada HU y se describen de forma clara y simple; de forma tal que se entienda lo que el sistema tiene que hacer y facilite su construcción. Los puntos estimados se transformaron en días para hacer más entendible el tiempo. A continuación se describen 7 de estas, correspondientes a 7 HU.

Tabla 12: Tarea de Ingeniería 1 de la HU7

Tarea de Ingeniería	
Número de Tarea: 1	Número de Historia de Usuario: 7
Nombre de la Tarea: Implementar monitorear Recomendaciones.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 7 días
Fecha de Inicio: 1/2/2015	Fecha de Fin: 6/2/2015
Programador Responsable: Orlando Cruz Torres	
Descripción: Se implementan las funcionalidades necesarias para que el usuario pueda monitorear las recomendaciones generadas con respecto al consenso de uno de sus proyectos.	



Tabla 13: Tarea de Ingeniería 1 de la HU1

Tarea de Ingeniería	
Número de Tarea: 1	Número de Historia de Usuario: 1
Nombre de la Tarea: Implementar registrar datos del Consenso.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 7 días
Fecha de Inicio: 7/2/2015	Fecha de Fin: 14/2/2015
Programador Responsable: Orlando Cruz Torres	
Descripción: Se implementan las funcionalidades necesarias para que el usuario pueda registrar los datos de su consenso de forma correcta.	

Tabla 14: Tarea de Ingeniería 1 de la HU2

Tarea de Ingeniería	
Número de Tarea: 1	Número de Historia de Usuario: 2
Nombre de la Tarea: Implementar modificar datos del Consenso.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 7 día
Fecha de Inicio: 15/2/2015	Fecha de Fin: 22/2/2015
Programador Responsable: Orlando Cruz Torres	
Descripción: Se implementan las funcionalidades necesarias para que el usuario pueda modificar los datos de su consenso de forma exitosa.	



Tabla 15: Tarea de Ingeniería 1 de la HU3

Tarea de Ingeniería	
Número de Tarea: 1	Número de Historia de Usuario: 3
Nombre de la Tarea: Implementar listar y eliminar datos del Consenso.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 7 día
Fecha de Inicio: 23/2/2015	Fecha de Fin: 1/3/2015
Programador Responsable: Orlando Cruz Torres	
Descripción: Se implementan las funcionalidades necesarias para que el usuario pueda listar y eliminar los datos de su consenso de forma exitosa.	

Tabla 16: Tarea de Ingeniería 1 de la HU5

Tarea de Ingeniería	
Número de Tarea: 1	Número de Historia de Usuario: 5
Nombre de la Tarea: Implementar generar Recomendaciones.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 10 días
Fecha de Inicio: 2/3/2015	Fecha de Fin: 12/3/2015
Programador Responsable: Orlando Cruz Torres	
Descripción: Se implementan las funcionalidades necesarias para que el usuario pueda enviar las recomendaciones generadas de su consenso a los usuarios participantes.	



Tabla 17: Tarea de Ingeniería 1 de la HU6

Tarea de Ingeniería	
Número de Tarea: 1	Número de Historia de Usuario: 6
Nombre de la Tarea: Implementar notificar Recomendaciones.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 10 días
Fecha de Inicio: 12/3/2015	Fecha de Fin: 22/3/2015
Programador Responsable: Orlando Cruz Torres	
Descripción: Se implementan las funcionalidades necesarias para que el usuario pueda recibir recomendaciones de consenso de acuerdo al proyecto donde este participando.	

Tabla 18: Tarea de Ingeniería 1 de la HU4

Tarea de Ingeniería	
Número de Tarea: 1	Número de Historia de Usuario: 4
Nombre de la Tarea: Implementar calcular Consenso.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 10 días
Fecha de Inicio: 22/3/2015	Fecha de Fin: 1/4/2015
Programador Responsable: Orlando Cruz Torres	
Descripción: Se implementa el funcionamiento del modelo Ágora.	

3.2 Estilos de Programación

Los estilos de programación se enfocan en definir la estructura y apariencia física del código a programar, lo que facilita su entendimiento y lectura. En la implementación de la aplicación se utilizaron diferentes estilos que se describen en los siguientes acápite:



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 con mayúscula. Ver ejemplo en la figura 11.

```
class Recomendacion extends CActiveRecord
{
    ...
    public $id_recomendacion;
    ...
}
```

Fig. 11: Definición de Clases.

3.2.2 Definiciones 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. Ver ejemplo en la figura 12.

```
public function tableName()
{
    return 'recomendacion';
}
```

Fig. 12: Definición de Métodos.

3.2.3 Llamadas a funciones y asignación de variables

Las llamadas a funciones se realizarán sin espacios entre el nombre de la función y los paréntesis, los atributos pasados por parámetro se separarán por una coma y un espacio después de ésta en caso de existir más de un atributo. Las asignaciones se realizarán mediante un espacio después del signo de igualdad (=), o también mediante este otro símbolo (->). Ver ejemplo en la figura 13.



```
extract($ _POST);
$arreglo = array();
$usuario='ajperez';
$buscarproyectos=new CDbCriteria;
$buscarproyectos->compare('usuario_id_usuario', $usuario);
$buscarproyectos->select="t.nombre,t.id_proyecto";
$buscarproyectos->join = "INNER JOIN consenso c on c.proyecto_id_proyecto= t.id_proyecto";
$proyectosmonitorear= Proyecto::model()->findAll($buscarproyectos);
$arreglo['proyectosmonitorear']=$proyectosmonitorear;
```

Fig. 13: Llamadas a funciones y asignación de variables.

3.2.4 Estructuras de Control

Las estructuras de control incluyen if else, for, while, etc. Ver ejemplo en la figura 14 de la estructura if else.

```
if(isset($project)) {
    if( $project==-1){
        $this->render('gestionar', $arreglo);
    }else{
        $usuario='';
        $estadoactual='Sin Enviar';
    }
    ...
}
```

Fig. 14: Estructura de control.

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. (42)



A continuación se muestra el diagrama de despliegue que se corresponde con el sistema que se desea implementar.

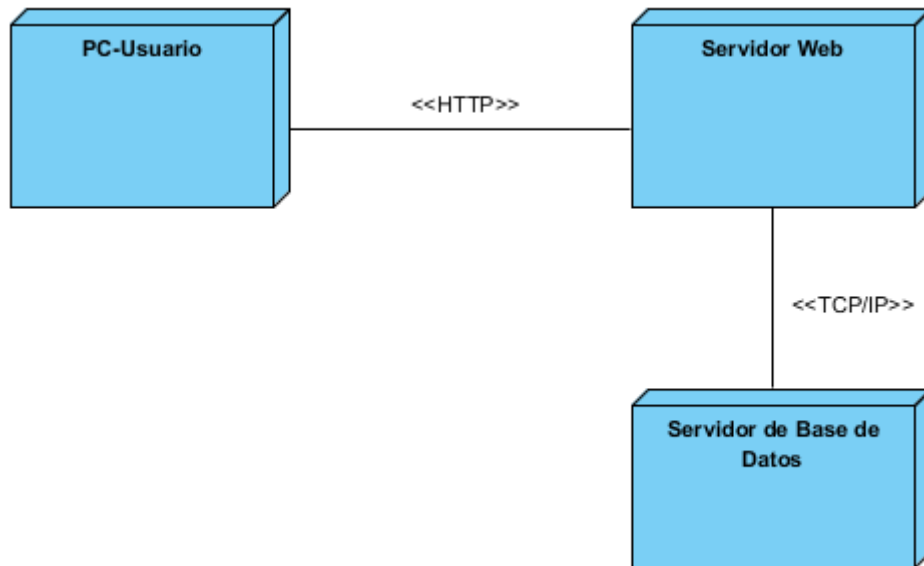


Fig. 15: Diagrama de Despliegue.

- ✓ **Estaciones de trabajo con la aplicación cliente:** Está definida por las estaciones de trabajo que el usuario utilizará para acceder a la aplicación Web.
- ✓ **Servidor de aplicación:** 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

Uno de los pilares de XP es el proceso de pruebas. Este 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 y refactorizaciones.

XP divide las pruebas del sistema en dos grupos: pruebas unitarias, encargadas de verificar el código y diseñada por los programadores, y pruebas de aceptación destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final. (51)

✓ **Pruebas Unitarias**

Las pruebas unitarias están enfocadas a 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 basan 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. (52)

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. (51)

Para implementar este nivel de prueba se utilizaran 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 la prueba 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. (53) 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 **actionCreate** contenidos en la clase **ConsensoController**. Este método permite al usuario crear los parámetros para sus consensos. Ver código del método en la Figura 16.

```
public function actionCreate() {  
    $model = new Consenso; // 1  
    $model->ronda_actual = 0; // 2  
  
    // Uncomment the following line if AJAX validation is needed  
    $this->performAjaxValidation($model); // 3  
  
    if (isset($_POST['Consenso'])) { // 4  
        $model->attributes = $_POST['Consenso']; // 5  
        if ($model->save()) //6  
            $this->redirect(array('view', 'id' => $model->id_consenso)); // 7  
    }  
    // 8  
    $this->render('create', array(  
        'model' => $model,  
    ));  
}
```

Fig. 16: Método que permite el registro de parámetros sobre un consenso.

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

Una vez construido el grafo de flujo asociado al procedimiento anterior se determina la complejidad ciclomática, la cual es una métrica de software 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.

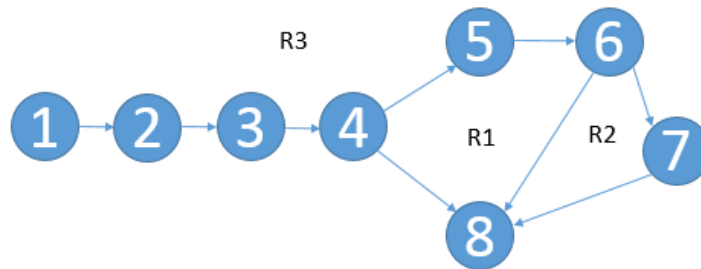


Fig. 17: Grafo de flujo asociado al método actionCreate.

Cálculo de la complejidad ciclomática para el grafo de flujo de la Fig. 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.

$$V(g) = (9-8) + 2 = 3 \text{ (I)} \quad V(g) = 2 + 1 = 3 \text{ (II)} \quad V(g) = 3 \text{ (III)}$$

La evaluación de las fórmulas I, II y III arroja un valor de complejidad ciclomática igual a 3, de manera que existen 3 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 – 5 – 6 – 7 – 8

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

Camino básico #3: 1 – 2 – 3 – 4 – 8

Se procede a ejecutar los casos de pruebas para cada uno de los caminos básicos determinados en el grafo de flujo (Ver Tablas 19, 20, 21). 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.

Tabla 19: Caso de Prueba para el Camino Básico #1.

Caso de Prueba Camino Básico #1
<p>Descripción: La Variable “model” contiene los siguientes datos definidos en el modelo Consenso. Los cuales cumplirán los siguientes requisitos:</p> <ul style="list-style-type: none">-Proyecto: se debe seleccionar un proyecto que no haya sido utilizado antes.-Umbral: debe ser un número real entre el intervalo [0; 1].-Exilon: debe ser un número real entre el intervalo [0; 1].-Total de Rondas: debe ser un número entre el intervalo [5; 20].
<p>Condiciones: El usuario debe de estar autenticado y con el formulario de registro abierto.</p>
<p>Entrada:</p> <ul style="list-style-type: none">✓ Proyecto: “Economía Mundial”.✓ Umbral: “0.5”.✓ Exilon: “0.3”.✓ Total de Rondas: “5”.
<p>Resultado Esperado: El sistema debe de almacenar los datos correctamente sin ningún inconveniente.</p>
<p>Resultado: Satisfactorio.</p>

Tabla 20: Caso de Prueba para el Camino Básico #2.

Caso de Prueba Camino Básico #2
<p>Descripción: La Variable “model” contiene los siguientes datos definidos en el modelo Consenso. Los cuales cumplirán los siguientes requisitos:</p>



<p>-Proyecto: se debe seleccionar un proyecto que no haya sido utilizado antes.</p> <p>-Umbral: debe ser un número real entre el intervalo [0; 1].</p> <p>-Exilon: debe ser un número real entre el intervalo [0; 1].</p> <p>-Total de Rondas: debe ser un número entre el intervalo [5; 20].</p>
<p>Condiciones: El usuario debe de estar autenticado y con el formulario de registro abierto.</p>
<p>Entrada:</p> <ul style="list-style-type: none">✓ Proyecto: “Inteligencia Artificial”.✓ Umbral: “0.8”.✓ Exilon: “0.5”.✓ Total de Rondas: “7”.
<p>Resultado Esperado: El sistema muestra un mensaje especificando que el proyecto seleccionado ya está siendo utilizado en el proceso de Consenso.</p>
<p>Resultado: Satisfactorio.</p>

Tabla 21: Caso de Prueba para el Camino Básico #3.

Caso de Prueba Camino Básico #3
<p>Descripción: La Variable “model” contiene los siguientes datos definidos en el modelo Consenso. Los cuales cumplirán los siguientes requisitos:</p> <p>-Proyecto: se debe seleccionar un proyecto que no haya sido utilizado antes.</p> <p>-Umbral: debe ser un número real entre el intervalo [0; 1].</p> <p>-Exilon: debe ser un número real entre el intervalo [0; 1].</p> <p>-Total de Rondas: debe ser un número entre el intervalo [5; 20].</p>
<p>Condiciones: El usuario debe de estar autenticado y con el formulario de registro abierto.</p>
<p>Entrada:</p>



<ul style="list-style-type: none">✓ Proyecto: “Practica Profesional”.✓ Umbral: “5”.✓ Exilon: “3”.✓ Total de Rondas: “4”.
Resultado Esperado: El sistema muestra un mensaje de error especificando que los campos Umbral y Exilon contienen valores muy grandes, mientras el campo Total de Rondas contiene un valor muy pequeño.
Resultado: Satisfactorio.

3.5.1 Resultados

Se realizaron un total de 20 casos de pruebas de caja blanca (pruebas unitarias), de ellos 2 resultaron no satisfactorios, lo cual representa el 10% del total de casos de prueba de caja blanca realizados, ver Figura 18. Mientras los 18 casos de prueba restantes resultaron satisfactorios para un 90% del total. Los errores detectados por los casos de pruebas no satisfactorios fueron mitigados después de 1 iteración de prueba.



Fig. 18: Resultados de las pruebas de caja blanca.



3.6 Pruebas de Caja Negra

Las pruebas de caja negra o también llamadas pruebas de aceptación 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 historia de usuario se realizan las pruebas de aceptación que se necesiten. Su propósito es garantizar que las historias de usuario se hayan cumplido y la aplicación sea realmente lo solicitado por el cliente. Una historia de usuario no se considera terminada hasta que no haya pasado sus pruebas de aceptación.

Para la realización de las pruebas de caja negra se empleó la técnica Partición de Equivalencia. Esta permite examinar los valores válidos e inválidos de las entradas existentes en el software (46).

A continuación se muestran 6 casos de prueba de aceptación realizados sobre 6 HU:

Tabla 22: Prueba de aceptación para la HU Establecer Parámetros del Consenso.

Caso de Prueba de Aceptación	
Código: H1_P1	Historia de Usuario: 1
Nombre: Registrar Parámetros	
Descripción: El formulario debe permitir la inserción de datos válidos para poder almacenar parámetros de consenso.	
Condiciones de Ejecución: La página de registro de los datos del consenso debe de estar abierta. El usuario debe de estar autenticado en la aplicación.	
Entradas/Pasos de Ejecución: <ol style="list-style-type: none">1. Selecciona la opción Mis Proyectos del menú principal.2. Selecciona la opción Consenso del proyecto que desee analizar.3. Introducir los datos en los campos Umbral, Exilon y Total de Rondas.4. Click en el botón Crear.	
Resultado Esperado: El sistema debe almacenar en la base de datos los valores introducidos en los parámetros del consenso, en caso de no ser valores válidos el sistema informara de los errores.	



Resultado Obtenido: Los datos fueron creados correctamente. No hubo ningún error de validación.
Evaluación de la Prueba: Satisfactoria.

Tabla 23: Prueba de aceptación para la HU Modificar Parámetros del Consenso.

Caso de Prueba de Aceptación	
Código: H2_P2	Historia de Usuario: 2
Nombre: Modificar Parámetros	
Descripción: El formulario debe permitir la modificación de los datos ya almacenados para los parámetros de un consenso.	
Condiciones de Ejecución: El usuario debe de estar autenticado en la aplicación. Debe de haber registrado con anterioridad parámetros de consenso para el proyecto que este analizando.	
Entradas/Pasos de Ejecución: <ol style="list-style-type: none">1. Selecciona la opción Mis Proyectos del menú principal.2. Selecciona la opción Consenso del proyecto que desee analizar.3. Selecciona del menú de Consenso la opción Editar Parámetros.4. Introducir los datos a modificar en los campos Umbral, Exilon y Total de Rondas.5. Click en el botón Actualizar.	
Resultado Esperado: El sistema debe almacenar en la base de datos los valores introducidos para la modificación de los parámetros del consenso, en caso de no ser valores válidos el sistema informara de los errores.	
Resultado Obtenido: Los datos de los parámetros del consenso fueron modificados en la base de datos correctamente. No hubo ningún error de validación.	
Evaluación de la Prueba: Satisfactoria.	



Tabla 24: Prueba de aceptación para la HU Calcular Consenso.

Caso de Prueba de Aceptación	
Código: H4_P3	Historia de Usuario: 4
Nombre: Calcular Consenso	
Descripción: Debe permitir la ejecución del algoritmo para la generación de recomendaciones para lograr el consenso.	
Condiciones de Ejecución: El usuario debe de estar autenticado en la aplicación. Debe de haber registrado con anterioridad parámetros de consenso para alguno de sus proyectos.	
Entradas/Pasos de Ejecución: <ol style="list-style-type: none">1. Selecciona la opción Mis Proyectos del menú principal.2. Selecciona la opción Consenso del proyecto que desee analizar.3. Selecciona la opción Consenso del menú de Consenso y hace click sobre el botón Iniciar el Proceso.	
Resultado Esperado: El sistema debe generar automáticamente las recomendaciones necesarias para lograr el consenso en el proyecto seleccionado en caso de que este lo requiera y ser enviadas al usuario creador del proyecto.	
Resultado Obtenido: Las recomendaciones fueron generadas satisfactoriamente y fueron notificadas al usuario.	
Evaluación de la Prueba: Satisfactoria.	



Tabla 25: Prueba de aceptación para la HU Generar Recomendaciones.

Caso de Prueba de Aceptación	
Código: H5_P4	Historia de Usuario: 5
Nombre: Generar Recomendaciones	
Descripción: El sistema debe mostrar una lista de notificaciones de las recomendaciones realizadas para el proceso de consenso en el proyecto que el usuario seleccione para estas ser enviadas y notificadas a los usuarios involucrados en ellas.	
Condiciones de Ejecución: El usuario debe de estar autenticado en la aplicación. Deben de existir recomendaciones almacenadas para sus proyectos.	
Entradas/Pasos de Ejecución: <ol style="list-style-type: none">1. Selecciona la opción Mis Proyectos del menú principal.2. Selecciona la opción Consenso del proyecto que desee analizar.3. Selecciona la opción Enviar Recomendaciones del menú de Consenso.4. Puede decidir si Enviar o Cancelar la recomendación que desee.5. Click en el botón Enviar.	
Resultado Esperado: El sistema debe enviar y notificar automáticamente las recomendaciones enviadas a los usuarios involucrados en ellas.	
Resultado Obtenido: Las recomendaciones fueron enviadas y notificadas satisfactoriamente a los usuarios pertinentes.	
Evaluación de la Prueba: Satisfactoria.	



Tabla 26: Prueba de aceptación para la HU Notificar Recomendaciones.

Caso de Prueba de Aceptación	
Código: H6_P5	Historia de Usuario: 6
Nombre: Notificar Recomendaciones	
Descripción: El sistema debe mostrar una lista de notificaciones de las recomendaciones enviadas sobre el proceso de consenso en el proyecto que el usuario está participando para este poder elegir si desea o no aceptar la recomendación.	
Condiciones de Ejecución: El usuario debe de estar autenticado en la aplicación. Deben de estar participando en un proyecto y que este a la vez contenga recomendaciones sobre el consenso y que hayan sido enviadas al él.	
Entradas/Pasos de Ejecución: <ol style="list-style-type: none">1. Se dirige al menú principal, selecciona la opción Recomendaciones.2. Selecciona un proyecto y hace click sobre el botón Filtrar.3. Selecciona una nueva etiqueta.4. Puede decidir si Cambiar o Rechazar la recomendación que desee.5. Click en el botón Cambiar.	
Resultado Esperado: El sistema debe suprimir la recomendación analizada y actualizar la base de datos con el nuevo dato proporcionado por la nueva etiqueta que el usuario eligió.	
Resultado Obtenido: Las recomendaciones fueron eliminadas una vez que el usuario procedió a aceptar lo notificado. Los datos en la base de datos fueron actualizados satisfactoriamente por la nueva etiqueta seleccionada.	
Evaluación de la Prueba: Satisfactoria.	



Tabla 27: Prueba de aceptación para la HU Monitorear Recomendaciones.

Caso de Prueba de Aceptación	
Código: H7_P6	Historia de Usuario: 7
Nombre: Monitorear Recomendaciones	
Descripción: El sistema debe mostrar una lista recomendaciones generadas por el proceso de consenso en el proyecto seleccionado para ver el estado en que esta se encuentra (Sin Enviar, Enviado, Cancelado, Corregido).	
Condiciones de Ejecución: El usuario debe de estar autenticado en la aplicación. Deben de existir recomendaciones almacenadas para sus proyectos.	
Entradas/Pasos de Ejecución: <ol style="list-style-type: none">1. Selecciona la opción Mis Proyectos del menú principal.2. Selecciona la opción Consenso del proyecto que desee analizar.3. Selecciona la opción Monitorear Recomendaciones del menú de Consenso.	
Resultado Esperado: El sistema debe mostrar un listado con todas las recomendaciones existentes en el proyecto seleccionado y el estado en que esta se encuentra.	
Resultado Obtenido: Las recomendaciones para el proyecto seleccionado fueron listadas correctamente al igual que el estado en que estas se encontraban.	
Evaluación de la Prueba: Satisfactoria.	

3.6.1 Resultados

Se realizaron un total de 13 casos de pruebas de caja negra (pruebas de aceptación), de ellos todos resultaron satisfactorios, lo cual representa el 100% de satisfacción, ver Figura 19.



Fig. 19: Resultados de las pruebas de caja negra.

3.7 Resultados Esperados

Las pruebas unitarias estuvieron formadas por 20 casos de pruebas, donde se mitigaron 2 casos de pruebas que resultaron no satisfactorios después de una iteración de prueba. Por otro lado las pruebas de caja negra o aceptación fueron satisfechas al 100% lo que demuestra que los requisitos funcionales especificados en las historias de usuarios fueron cumplidos en su totalidad validando así la solución propuesta. Ver Anexo 1.

CONSIDERACIONES PARCIALES

- ✓ Se implementaron todas las funcionalidades definidas para el sistema.
- ✓ Las tareas de Ingeniería definidas evitaron una sobrecarga de trabajo, agilizando aún más el proceso.
- ✓ Fueron detectados 2 casos de pruebas de caja blanca con resultados no satisfactorios, estos fueron mitigados tras una iteración de prueba.
- ✓ La aplicación de las pruebas de aceptación a la solución permitieron comprobar la correcta implementación de las historias de usuarios definidas con anterioridad y permitir así la validación de la solución propuesta.



CONCLUSIONES GENERALES

En el presente trabajo de diploma se presentó la investigación y posterior proceso de desarrollo del módulo de Consenso para la Red Social Nexo mediante el uso de los MCD, que permite realizar este proceso en ambientes multiusuarios. De esta forma se le da cumplimiento al objetivo general de la investigación presentada y se pudo arribar a las siguientes conclusiones:

- La elaboración del Marco Teórico permitió conocer que en la literatura consultada solo se referencia la existencia de 2 métodos para el proceso de logro de consenso mediante el empleo de los MCD, y la presencia de desventajas presentadas por los mismos para poder lograr completamente este proceso.
- El trabajo en equipo definido por la metodología de desarrollo de software utilizada permitió la correcta definición por parte del cliente de las Historias de Usuario, logrando así ver con claridad los requisitos funcionales que hicieron posible el modelado de la solución propuesta, en cuanto al análisis y diseño de la herramienta.
- Se obtuvo un módulo funcional que responde a los requisitos planteados y a las necesidades del cliente.
- La realización de las pruebas definidas al módulo, permitió apreciar que todas en su gran medida fueron satisfactorias, lográndose de esta manera el cumplimiento de los requisitos exigidos inicialmente y la validación de la solución propuesta mediante la completa satisfacción del cliente.



RECOMENDACIONES

Teniendo en cuenta los resultados obtenidos en esta investigación y basado en la experiencia adquirida, se recomienda:

- Continuar el perfeccionamiento del módulo añadiendo la posibilidad de que el proceso de consenso también pueda ser aplicado a proyectos de tipo Públicos.



REFERENCIAS BIBLIOGRÁFICAS

- (1) «Toma de decisiones en grupo». [En línea]. Disponible en: <http://www.elergonomista.com/psicologia/17my19.html>. [Accedido: 06-nov-2014].
- (2) «TÉCNICAS POLITOLÓGICAS PARA LA GESTIÓN DE PROYECTOS SOCIALES». [En línea]. Disponible en: <http://www.eumed.net/libros-gratis/2008a/362/index.htm> [Accedido: 06-nov-2014]. Autor: Eduardo Jorge Arnoletto.
- (3) MATA, F. AND J. C. MARTÍNEZ. Consensus reaching with different aggregation techniques. In III Simposio sobre Lógica Fuzzy y Soft Computing. 2010.
- (4) SENGE, P. La Quinta Disciplina En La Practica/Fifth Discipline In The Practice. Edtion ed.: Ediciones Granica SA, 2005. ISBN 9506414211.
- (5) MATA, F., L. MARTÍNEZ AND E. HERRERA-VIEDMA An adaptive consensus support model for group decision-making problems in a multigranular fuzzy linguistic context. Fuzzy Systems, IEEE Transactions on, 2009, 17(2), 279-290.
- (6) MATA, F. Modelos para Sistemas de Apoyo al Consenso en Problemas de Toma de Decisión en Grupo definidos en Contextos Lingüísticos Multigranulares. Universidad de Jaén, Doctoral Thesis 2006.
- (7) J. L. Salmeron, "Supporting decision makers with Fuzzy Cognitive Maps," vol. 52, ed: Industrial Research Institute, Inc, 2009, pp. 53-59.
- (8) B. Kosko, "Fuzzy cognitive maps," International Journal of ManMachine Studies, vol. 24, pp. 65-75, 1986.
- (9) 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.
- (10) 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.
- (11) ALTAY, A. AND G. KAYAKUTLU Fuzzy cognitive mapping in factor elimination: A case study for innovative power and risks. Procedia Computer Science, 2011, 3(0), 1111-1119.
- (12) IAKOVIDIS, D. K. AND E. PAPAGEORGIU Intuitionistic Fuzzy Cognitive Maps for Medical Decision Making. Information Technology in Biomedicine, IEEE Transactions on, 2011, 15(1), 100-107.



- (13)** ROUSE, W. B. AND N. M. MORRIS On looking into the black box: Prospects and limits in the search for mental models. *Psychological bulletin*, 1986, 100(3), 349.
- (14)** SALMERON, J. L. Fuzzy cognitive maps for artificial emotions forecasting. *Applied Soft Computing*, 2012.
- (15)** GEORGOPOULOS, V. C. 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*. Springer, 2013, p. 415-436.
- (16)** LEE, S., J. YANG AND J. HAN Development of a decision making system for selection of dental implant abutments based on the fuzzy cognitive map. *Expert Systems with Applications*, 2012, 39(14), 11564-11575.
- (17)** PAPAGEORGIU, E. I. AND J. L. SALMERON. A Review of Fuzzy Cognitive Maps research during the last decade. *IEEE Transactions on Fuzzy Systems*. 2012.
- (18)** KHAN, M. S. AND M. QUADDUS Group Decision Support Using Fuzzy Cognitive Maps for Causal Reasoning. *Group Decision and Negotiation*, 2004, 13(5), 463-480.
- (19)** BRADLEY, R., F. DIETRICH AND C. LIST. Aggregating causal judgements. In. London: London School of Economics, 2006, vol. 2012.
- (20)** Salmeron, J. L. Ranking fuzzy cognitive map based scenarios with TOPSIS," *Expert Systems with Applications*. 2012. pp. 2443-2450. Vol. 39.
- (21)** Bryson, N. (1997). Generating consensus fuzzy cognitive maps. 1997 IASTED International Conference on Intelligent Information Systems (IIS '97), Grand Bahama Island, BAHAMAS.
- (22)** Herrera-Viedma, E., F. Cabrerizo, et al. (2011). Applying Linguistic OWA Operators in Consensus Models under Unbalanced Linguistic Information Recent Developments in the Ordered Weighted Averaging Operators: Theory and Practice. R. Yager, J. Kacprzyk and G. Beliakov, Springer Berlin / Heidelberg. 265: 167-186.
- (23)** Espinilla, M., J. Liu, et al. (2011). "An extended hierarchical linguistic model for decision making problems." *Computational Intelligence* 27(3): 489-512.



- (24) Bueno, S. and J. L. Salmeron (2009). "Benchmarking main activation functions in fuzzy cognitive maps." *Expert Systems with Applications*. 36(3): 5221-5229.
- (25) Salmeron, J. L. (2009). "Augmented fuzzy cognitive maps for modelling LMS critical success factors." *Knowledge-Based Systems* 22(4): 275-278.
- (26) Salmeron, J. L. (2009). "Supporting decision makers with Fuzzy Cognitive Maps." *Research-Technology Management* 52(3): 53-59.
- (27) Linstone, H. A. and M. Turoff (1979). *The Delphi Method: techniques and applications*, Addison-Wesley Massachusetts.
- (28) Pérez-Teruel, Leyva-Vázquez, Estrada-Sentí (2014). "Proceso de consenso en modelos mentales usando mapas cognitivos difusos y computación con palabras." Universidad de las Ciencias Informáticas.
- (29) Pérez-Teruel, (2014). "MODELO DE PROCESO DE LOGRO DE CONSENSO EN MAPAS COGNITIVOS DIFUSOS PARA LA TOMA DE DECISIONES EN GRUPO." Tesis presentada en opción al Grado Científico de Doctor en Ciencias Técnicas, Universidad de las Ciencias Informáticas.
- (30) «Loaiza, Douglas Alfredo. ». [En línea] Disponible en: <http://pnfiingenieriadesoftwaregrupocuatro.blogspot.com/2012/07/bienvenidos-al-blog.html>.
- (31) «El lenguaje Unificado De Modelado (UML)». [En línea]. Disponible en: <http://profesores.fib.unam.mx/carlos/aydoo/uml.html>
- (32) «Herramientas Case». [En línea]. Disponible en: <http://www.mitecnologico.com/Main/HerramientasCase>
- (33) «Introducción a PHP». [En línea]. Disponible en: <http://www.maestrosdelweb.com/phpintro/>
- (34) « Buenos motivos para trabajar con PHP». [En línea]. Disponible en: <http://www.lancetalent.com/blog/6-buenos-motivos-para-trabajar-con-php/>
- (35) «Yii Framework». [En línea]. Disponible en: <http://caraballomaestre.blogspot.com/2011/05/por-que-yii-framework.html>
- (36) «PostgreSQL vs. MySQL». [En línea]. Disponible en: http://danielpecos.com/docs/mysql_postgres/x15.html
- (37) «NetBeans». [En línea]. Disponible en: <http://www.ecured.cu/index.php/NetBeans>
- (38) Leyva-Vázquez, M.Y., et al., Modelo para el análisis de escenarios basado en mapas cognitivos difusos. *Ingeniería y Universidad* 2013. 17(2).



- (39)** Groumpos, P., Fuzzy Cognitive Maps: Basic Theories and Their Application to Complex Systems, in Fuzzy Cognitive Maps. 2010, Springer Berlin / Heidelberg. p. 1-22.
- (40)** A probabilistic model for linguistic multi-expert decision making involving semantic overlapping. Yan, H.-B. 2011, Expert Systems with Applications, Vol. 38, pp. 8901-8912.
- (41)** Sandy Schuman. The IAF Handbook of Group Facilitation: Best Practices from the Leading Organization in Facilitation. Jossey-Bass, 2005. ISBN 0-7879-7160-X
- (42)** «Yii Framework». [En línea]. Disponible en: <http://www.yiiframework.com/doc/guide/1.1/es/quickstart.what-is-yii>
- (43)** «Bootstrap». [En línea]. Disponible en: <http://www.ecured.cu/index.php/Bootstrap>
- (44)** «Patricio Letelier, M^a Carmen Penadés. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP) ». [En línea] Disponible en: <http://eva.uci.cu>.
- (45)** «Introducción al MVC de Yii – Parte I ». [En línea]. Disponible en: <http://blog.jorgeivanmeza.com/2009/03/introduccion-al-mvc-de-yii-parte-i/>
- (46)** « ¿Qué es un Patrón de Diseño? ». [En línea]. Disponible en: <https://msdn.microsoft.com/es-es/library/bb972240.aspx>
- (47)** CAMPO, G. D. Patrones de Diseño, Refactorización y Antipatrones. Ventajas y Desventajas de su Utilización en el Software Orientado a Objetos. 2009.
- (48)** 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.
- (49)** « Patrones de Diseño (Design Patterns)». [En línea]. Disponible en: http://www.ecured.cu/index.php/Patrones_Gof
- (50)** Sagulo, Ignacio. Integración de los patrones de Diseño en los ambientes de desarrollo Orientados a Objetos. Buenos Aires: Universidad de Buenos Aires: s.n., 2012. págs. 19-22.
- (51)** 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.
- (52)** BORIS BEIZER. Software Testing Techniques. 2da. S.l.: Intl Thomson Computer Pr (T), 1990.



- (53) ROGER S. PRESSMAN. Ingeniería del Software: Un Enfoque Práctico. 5ta. S.I.: McGraw-Hill Companies, 2002.
- (54) Herrera et al. 2009a; Leyva-Vázquez 2013a
- (55) Pavan, Bárbara. “ENTENDIENDO HTML5: GUÍA PARA PRINCIPIANTES” [Consultado en línea: 15-01-15] <http://bitelia.com/2013/05/entendiendo-html5-guia-para-principiantes>
- (56) <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-262.pdf>
- (57) 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.
- (58) Eguíluz Pérez, Javier. “Introducción a CSS” [Disponible en:] <http://www.librosweb.es/css>
- (59) K BECK. Extreme Programming Explained. Embrace Change. In: Pearson Education. 1999
- (60) « Toma de Decisiones ». [En línea]. Disponible en: http://www.ecured.cu/index.php/Toma_de_decisiones
- (61) PEREZ TERUEL, LEYVA VAZQUEZ, ESTRADA SENTI, “Proceso de consenso en modelos mentales usando mapas cognitivos difusos y computación con palabras.”, Universidad de las Ciencias Informáticas, 2014.



ANEXOS

Anexo1: Carta de Aceptación.

CARTA DE ACEPTACION DEL CLIENTE

Por medio de la presente yo **Karina Pérez Teruel** cliente del *Módulo para la automatización del proceso de Consenso en Mapas Cognitivos Difusos para la Red Social NEXO*, hago constar que el módulo ha sido desarrollado completamente y cumple con el 100% de los requisitos funcionales acordados con el cliente. Además, los casos de prueba de aceptación realizados fueron satisfactorios en su totalidad. Por lo que se expende la presente como muestra de que el módulo desarrollado cumple con las necesidades y expectativas del cliente. Y para que pueda hacerlo constar se emite la misma en la Universidad de Ciencias Informáticas (UCI), a los 1 días del mes de Junio de 2015.

Dr. Karina Pérez Teruel

Est. Orlando Cruz Torres