

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 3



# Trabajo de Diploma para Optar por el Título de Ingeniero en Ciencias Informáticas

IMPLEMENTACIÓN DEL MÉTODO PROMETHEE  
EN UN SISTEMA DE AYUDA A LA TOMA DE DECISIÓN MULTICRITERIO

**Autor(es):**

Reinier Torres Estol

Javier Enrique Padrón Del Pico

**Tutor(es):**

Ing. Hermes Miguel Velázquez Domínguez

Ing. Lizandra Arza Pérez

La Habana, Junio de 2013

“Año 53 de la Revolución”

“Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber”.

**Albert Einstein**

“El vencer las dificultades es en todas las cosas gran parte del mérito”

**Voltaire**

# *Declaración de Autoría*

---

## **Declaración de autoría**

Declaramos por este medio que, Reinier Torres Estol y Javier Enrique Padrón Del Pico con carnet de identidad 90012226322 y 89100329963 respectivamente, somos autores de la presente tesis y autorizamos a la Universidad de las Ciencias Informáticas hacer uso del mismo en su beneficio, así como los derechos patrimoniales, con carácter exclusivo.

Para que así conste firmamos la presente declaración jurada de auditoría en La Habana a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_  
Javier Enrique Padrón del Pico

Autor

\_\_\_\_\_  
Reinier Torres Estol

Autor

\_\_\_\_\_  
Ing. Hermes Miguel Velázquez Domínguez

Tutor

\_\_\_\_\_  
Ing. Lizandra Arza Pérez

Tutor

# *Datos de Contacto*

---

## **Datos de Contacto**

**Hermes Miguel Velázquez Domínguez:** Graduado de Ciencias Informáticas en la Universidad de las Ciencias Informáticas, 2008, profesor instructor.

**Lizandra Arza Pérez:** Graduada de Ingeniería Informática en el IPSJAE, año 2002. Profesora Asistente. Subdirectora de Investigaciones de Calisoft.

# Agradecimientos

---

*A la Revolución, por permitirnos la educación y formación profesional, por hacernos personas dignas de estos tiempos. A la UCI, por ser nuestro hogar, por darnos la posibilidad de crecer como personas. A mi mamá por darme la educación, los consejos y transmitirme toda la confianza y experiencia para hoy llegar a lograr este sueño, a mi papá que más que padre es mi amigo, sé que siempre podré contar con él para lo que sea cómo puede el hacerlo conmigo. A mi hermana que hoy no está y para mí es una ausencia bien dolorosa pero sé que en su corazón sí lo está, a mi cuñado por darme la luz de que esta carrera era para mí; a mi sobrina linda que la adoro. A mis cuatro abuelos que hoy tengo la suerte de tenerlos vivos y en presencia de unos de los mejores momentos de mi vida pero en especial a ese abuelo que durante toda mi vida ha sido además padre, padrino y amigo no importa la situación ni el momento. A mis tías, mis primos y en fin a mi familia en general. A mis tutores que han estado a pesar de sus obligaciones siempre para ayudarnos y al oponente que con su ojo crítico ayudó a que fuera esto posible con la calidad que lo requiere, al tribunal, siempre aconsejando y sugiriendo en nuestro beneficio. Agradecerle al equipo de trabajo por colaborar, a mis amigos de la universidad tanto del aula como del deporte y a los profesores Jorge Ricardo (El Chino), Alain Eduardo por ser mi profesor ejemplo de profesionalidad y guía de lo que debe ser un informático hoy en día. Al amor de mi vida que durante 4 años ha estado a mi lado a pesar de todos los obstáculos que la vida nos ha puesto, a ella gracias por ser la luz de mis días cuando apenas había nubes grises en el cielo y a su familia como olvidarla si desde que llegue un buen día me acogieron como uno más de ellos, de lo que estoy totalmente agradecido. A mi compañero de tesis Javier que es mi amigo y como mi hermano y aprovecho la oportunidad para agradecerle todo lo que ha hecho por mí y decirle que siempre encontrará aquí a un hermano para todo. En fin a todos los que he mencionado y aquellos que de una forma u otra han tenido que ver en mi formación ¡Gracias!*

Rey

# Agradecimientos

---

*Por darnos la oportunidad de formarnos como ingenieros dignos de esta época, conectados al futuro, pero por sobre todo, conectados a la Revolución.*

*A nuestro líder indiscutible, Fidel Castro, por dedicar más de medio siglo a una causa que para algunos es imposible, pero para muchos es confianza, por ser el creador de esta maravillosa universidad.*

*A nuestros padres. Sin ustedes no existiéramos, gracias por apoyarnos en cada momento y por estar ahí cuando los necesitamos.*

*No, fueron padres, sino amigos, compañeros, eternos compartidores de conocimientos, motivadores por excelencia por extraer lo máximo que podíamos dar en nuestros quehaceres, gracias profesores.*

*Aunque han tenido muchas responsabilidades, nos orientaron, supervisaron y velaron por cada uno de nuestros movimientos, gracias tutores Lizandra y Hermes.*

*Por su profunda y eterna amistad, a todas aquellas personas que alguna vez me brindaron su amistad y ayuda en los momentos más difíciles que he pasado. A mi familia habanera que me han acogido como suyo, a mi compañero de tesis y amigo de toda la vida que ha estado a mi lado sin importar las situaciones. A todos Gracias.*

*Javi*

# Dedicatoria

---

*A mi mamá y mi papá que siempre han estado a mi lado de manera incondicional ante cualquier situación, a mi hermana, a mis abuelos, ¡a los cuatro! que por encima de todo tengo la suerte de tenerlos vivos, a mi cuñado, a mi sobrina linda que la adoro, a mis tías, mis primas y primos, en fin a mi familia en general. A la mujer de mi vida que estuvo, está y estoy convencido que siempre estará a mi lado pese a cualquier dificultad que se nos pueda presentar en el largo camino de la vida y a su familiares que son para mí como segunda familia. A mis amistades y a todos los que se preocuparon por mí alguna vez por mí.*

**Rey**

*Por ser mi faro máspreciado y la razón de mí existir, a ti mi padre adorado, Manuel, va dedicado este trabajo.*

*A mi estrella, por tanta entrega, dedicación y a la que le debo la dulzura en mi carácter, a mi madre Lania.*

*A mi hermano querido Manuel y a toda mi familia por velar y confiar plenamente en mí.*

*A mi linda novia Yenny por tanto amor depositado.*

*A mis amigos que han sido como hermanos todos estos años. A mi compañero de tesis por dedicar horas en el trabajo colectivo.*

**Javi**

## Resumen

El encontrar la solución más acertada entre un conjunto de alternativas factibles siempre ha sido una dificultad para el hombre. Los métodos de evaluación multicriterios brindan la posibilidad de encontrar dicha solución. El uso del PROMETHEE (Preference Ranking Organization Methods for Enrichment Evaluations) está arraigado al manejo de grandes cantidades de alternativas lo que hace que el proceso de aplicación sea complejo y susceptible a errores, además que su aplicación de forma manual se torne poco práctica y en ocasiones imposible; por esta razón se ha planteado una investigación para la implementación de un sistema de apoyo a la decisión, que permita la aplicación del método. En el presente documento se describen las funcionalidades del método PROMETHEE utilizando SXP<sup>1</sup> como metodología de desarrollo y se define una arquitectura en capas. Se realiza una descripción de las herramientas y tecnologías utilizadas para el desarrollo de las funcionalidades del sistema, además se aplican las pruebas definidas determinándose su correcto funcionamiento. El resultado de esta investigación, es un sistema que facilita el trabajo a la hora de usar el método PROMETHEE para resolver problemas de decisión multicriterios.

**Palabras Clave:** criterio de decisión, decisión multicriterios, sistema de apoyo a la decisión

---

<sup>1</sup> SXP es una metodología compuesta por las metodologías ágiles (XP y SCRUM), que brinda una estrategia desde el punto de vista tecnológico al introducir procedimientos ágiles que permiten actualizar los procesos de software.



Capítulo I. Marco Teórico Referencial.....	13
1.1.  Sistemas de Apoyo a la Toma de Decisión.....	13
1.1.1  Proceso de toma de decisión.....	13
1.1.2  Sistemas de soporte a la toma de decisiones.....	14
1.2  Los métodos de Toma de Decisión Multicriterios PROMETHEE.....	16
1.3  Sistemas que implementan PROMETHEE.....	20
1.4  Herramientas y Metodologías de Desarrollo.....	21
1.4.1  Metodologías para el desarrollo.....	21
1.4.2  Lenguaje de Modelado.....	22
1.4.3  Herramientas Case.....	23
1.4.4  Lenguajes utilizados.....	23
1.4.5  Marcos de trabajo.....	23
1.4.6  Entornos de Desarrollo Integrado (IDEs).....	25
1.1.1  Gestores de Base de Datos.....	25
1.4.7  Servidor de Aplicaciones.....	26
Conclusiones Parciales.....	26
Capítulo 2. Características del Sistema.....	27
2.1  Descripción del problema.....	27
2.2  Solución Propuesta.....	27
2.3  Concepción inicial del sistema.....	27
2.4  Captura de requisitos.....	28
2.4.1  Historias de usuarios del negocio.....	28
2.4.2  Lista de reserva del producto (LRP).....	28
2.5  Diseño de metáforas.....	31
2.5.1  Historias de Usuario.....	31
2.5.2  Tareas Ingenieriles.....	32
2.6  Arquitectura seleccionada.....	34
2.7  Patrones de diseño.....	34
2.8  Modelo de Datos.....	36
Conclusiones Parciales.....	38
Capítulo 3 Implementación y Prueba.....	39
3.1  Implementación.....	39
3.1.1  Diagrama de componentes.....	39
3.1.2  Diagrama de Despliegue.....	40
Conclusiones Parciales.....	50
Conclusiones Generales.....	51
Recomendaciones.....	52
Referencias Bibliográficas.....	53
Anexos.....	55
Anexo 1. Plantilla de Concepción Inicial del Sistema.....	55
Anexo 2. Modelo de historias de usuarios del negocio.....	57
Anexo 3. Plantilla de historias de usuarios.....	58

Anexo 4. Plantilla Tarea de Ingeniería .....	62
Anexo 5. Descripción del Modelo de Datos.....	67
Anexo 6. Plantilla Caso de Prueba de Aceptación .....	69
Anexo 7 Interfaces de Usuario .....	73

## ÍNDICE DE FIGURAS

Figura 1: Esquema de la metodología SXP .....	22
Figura 2: Arquitectura cliente-servidor de Vaadin .....	24
Figura 3: Diagrama de Clases .....	36
Figura 4: Modelo de Datos.....	37
Figura 5: Diagrama de componentes .....	39
Figura 6: Diagrama de despliegue .....	40
Figura 7: Esquema de Caja Negra.....	41
Figura 8: Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.....	43
Figura 9: Representación en % de los resultados obtenidos en el atributo Responsabilidad .....	44
Figura 10: Representación en % de los resultados obtenidos en el atributo Complejidad.....	44
Figura 11: Representación en % de los resultados obtenidos en el atributo Reutilización. ....	44
Figura 12: Representación en % de los resultados obtenidos en el atributo Complejidad de Mantenimiento. ....	46
Figura 13: Representación en % de los resultados obtenidos en el atributo Cantidad de Pruebas. ....	47
Figura 14: Representación en % de los resultados obtenidos en el atributo Reutilización. ....	47
Figura 15: Vista ampliada del ranking final mediante el software Decision Lab.....	48
Figura 16: Vista completa del resultado final del problema de decisión multicriterios.....	49
Figura 17: Vista ampliada del ranking final mediante la aplicación MultiDecisionPAAT .....	49
Figura 18: Vista completa del resultado mediante la aplicación MultiDecisionPAAT .....	49
Figura 20. Registrarse.....	73
Figura 21: Crear una Decisión .....	74
Figura 22: Gestionar Criterios. ....	74
Figura 23: Seleccionar una cantidad de especialistas. ....	74
Figura 24: Gestionar Alternativas.....	75
Figura 25: Gestionar los pesos de las alternativas evaluadas en los criterios, además finalizar la creación de la decisión y guardar todos los datos. ....	75
Figura 26: Seleccionar una decisión existente.....	76

## ÍNDICE DE TABLAS

Tabla 1: Lista de Reserva del Producto .....	30
Tabla 2: HU Autenticación de usuarios .....	31
Tabla 3: HU Cargar Alternativas.....	32
Tabla 4: Tarea Ingenieril 1.....	33
Tabla 5: Tarea Ingenieril 4.....	33
Tabla 6: Tabla Decisión de la Base de Datos .....	37

Tabla 7: Diseño de Caso de Prueba de Aceptación .....	41
Tabla 8: Tabla de atributos que mide la métrica de tamaño de clase TOC.....	43
Tabla 9: Rango de valores para la evaluación técnica de los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización) relacionados con la métrica TOC. ....	43
Tabla 10: Tabla de atributos que mide la métrica relación entre clases (RC). ....	45
Tabla 11: Rango de valores para la evaluación técnica de los atributos de calidad (Acoplamiento, Complejidad Mantenimiento, Reutilización y Cantidad de Pruebas) relacionados con la métrica RC. ....	45
Tabla 12: Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.....	46
Tabla 13: Representación en % de los resultados obtenidos en el atributo Acoplamiento. ....	46
Tabla 17: Historia de Usuario 1 .....	59
Tabla 18: Historia de Usuario 2 .....	60
Tabla 19: Historia de Usuario 3.....	60
Tabla 20: Historia de Usuario 4.....	61
Tabla 25: Tarea de Ingeniería 1 .....	62
Tabla 26: Tarea de Ingeniería 2 .....	62
Tabla 27: Tarea de Ingeniería 3 .....	63
Tabla 28: Tarea de Ingeniería 4 .....	63
Tabla 29: Tarea de Ingeniería 5 .....	63
Tabla 30: Tarea de Ingeniería 6 .....	64
Tabla 31: Tarea de Ingeniería 7 .....	64
Tabla 32: Tarea de Ingeniería 8 .....	65
Tabla 33: Tarea de Ingeniería 9 .....	65
Tabla 34: Tarea de Ingeniería 10 .....	65
Tabla 35: Tarea de Ingeniería 11 .....	66
Tabla 36: Tarea de Ingeniería 12 .....	66
Tabla 37: Tarea de Ingeniería 13 .....	66
Tabla 38: Tarea de Ingeniería 14 .....	67
Tabla 39: Tarea de Ingeniería 15 .....	67
Tabla 40: Tablas de la Base de Datos .....	69
Tabla 41: Caso de Prueba de Aceptación 1-1.....	69
Tabla 42: Caso de Prueba de Aceptación 2-1.....	70
Tabla 43: Caso de Prueba de Aceptación 3-1.....	70
Tabla 44: Caso de Prueba de Aceptación 4-1.....	71
Tabla 45: Caso de Prueba de Aceptación 5-1.....	71
Tabla 46: Caso de Prueba de Aceptación 6-1.....	72
Tabla 47: Caso de Prueba de Aceptación 7-1.....	72
Tabla 48: Caso de Prueba de Aceptación 8-1.....	73

## INTRODUCCIÓN

La toma de decisiones es una actividad de la vida diaria de las personas, que abarca desde las decisiones personales hasta las relacionadas con las organizaciones o empresas, es por ello que el tratamiento de esta área del conocimiento desde el punto de vista computacional es un campo activo de investigación. Tomar decisiones implica, a partir de un problema elegir una alternativa lo suficientemente racional que permita maximizar el valor esperado al concluir la acción.

Para tomar una buena decisión es necesario primeramente trazar el objetivo que se quiere lograr, reunir toda la información necesaria teniendo en cuenta las necesidades de quien tiene que tomar dicha decisión. El proceso de toma de decisión aunque haya sido planificado con mucho cuidado en todas sus alternativas, debe de ser asumido con riesgo ya que toda elección lo conlleva en sí misma, y en el momento en el que se decide por una de las alternativas, se renuncia a posibles ventajas de otras opciones.

Una clase importante de problemas decisionales son los problemas multicriterios. Estos se definen como un conjunto de técnicas orientadas a asistir los procesos de decisiones (1). Se basa en la ponderación y compensación de variables que van a influir de manera positiva o negativa, que deben ser inventariados y clasificados previamente.

En la literatura aparece gran cantidad de métodos para resolver estos problemas. Dichos métodos comprenden la selección entre un conjunto de alternativas factibles, la optimización con varias funciones objetivos simultáneamente, un agente decisor y procedimientos de evaluación racionales y consistentes. También son considerados procesos mediante los cuales las empresas o entidades reúnen una serie de datos y criterios de evaluación para tomar decisiones estratégicas. Dentro de los principales métodos tenemos a PROMETHEE<sup>2</sup> por sus siglas en inglés, TOPSIS<sup>3</sup> por sus siglas en inglés, AHP<sup>4</sup> por sus siglas en inglés, ANP<sup>5</sup> por sus siglas en inglés, entre otros.

En el caso de PROMETHEE ofrece un conjunto de pasos para resolver el problema de ordenamiento, en un primer momento se obtiene un pre orden parcial compuesto por las alternativas y los criterios asociados a la decisión, luego se efectúa un ordenamiento completo de todas las alternativas evaluadas en los criterios de selección definidos, siempre sobre el conjunto de alternativas factibles. Luego de haber realizado este conjunto de pasos y de haber formulado los flujos positivos y negativos, se está en condiciones de obtener el criterio de selección.

---

<sup>2</sup> *Preference Ranking Organization Methods for Enrichment Evaluations*

<sup>3</sup> *Technique for Order of Preference by Similarity to Ideal Solution*

<sup>4</sup> *Analytic Hierarchy Process*

<sup>5</sup> *Analytic Network Process*

A partir de la complejidad matemática de los métodos multicriterios, entre ellos el PROMETHEE, se hace necesario para obtener mejores resultados, contar con sistemas que implementen las funciones y los pasos definidos en ellos.

En la Universidad de las Ciencias Informáticas (UCI) existen múltiples organizaciones donde se toman decisiones a diferentes niveles jerárquicos. Muchas de estas decisiones pueden ser representadas como un problema de decisión multicriterios. Esto conlleva a tomar decisiones donde los decisores deban auxiliarse en alguna tecnología que facilite su trabajo. Al tomar decisiones en problemas de esta naturaleza se corre riesgos de no considerar ciertas alternativas, lo que puede tener significación negativa desde el punto de vista económico y estratégico.

Por lo que se describe el siguiente **problema a resolver**: *La complejidad de los problemas de decisión multicriterios en la universidad propician que el proceso de toma de decisiones sea susceptible a errores.*

Como **objeto de estudio** se define *los sistemas de soporte a la toma de decisión multicriterios* y como **campo de acción** *los sistemas de soporte a la decisión que implementan el método PROMETHEE.*

El **objetivo general** de la presente investigación: *Desarrollar el módulo para el sistema de soporte a la toma de decisión multicriterios que implemente el método PROMETHEE para eliminar la complejidad de la confección manual de los mismos y evitar que el proceso sea susceptible a errores.*

A partir de un análisis del objetivo general se derivan los siguientes **objetivos específicos**:

1. Establecer el marco teórico referencial sobre los sistemas de soporte a la decisión, y caracterizar los métodos de toma de decisión multicriterios PROMETHEE.
2. Caracterizar el entorno de desarrollo de la aplicación.
3. Modelar las funcionalidades de los métodos como parte del sistema de ayuda a la toma de decisiones.
4. Implementar las funcionalidades modeladas.
5. Validar la aplicación propuesta.

Para dar cumplimiento a los objetivos específicos anteriores se diseñaron las siguientes tareas de investigación:

1. Elaborar el marco conceptual de la investigación.
2. Estudiar los métodos para la toma de decisión multicriterios.
3. Estudiar y caracterizar los sistemas de ayuda a la toma de decisión.

4. Estudiar la metodología definida para el desarrollo del sistema.
5. Estudiar técnicas de captura de requisitos.
6. Estudiar patrones de diseño.
7. Estudiar métricas para validar el diseño.
8. Estudiar las herramientas y tecnologías a utilizar para el desarrollo de la propuesta de solución.
9. Definir las funcionalidades del sistema
10. Definir mecanismos de diseño a utilizar.
11. Realizar diagramas de clases del diseño con la aplicación de patrones.
12. Describir las clases del diseño.
13. Realizar modelo de datos.
14. Realizar diagrama de componentes.
15. Definir estándares de codificación.
16. Realizar el diagrama de despliegue teniendo en cuenta los aspectos técnicos de la solución.
17. Realizar descripción por funcionalidades.
18. Implementar las funcionalidades definidas.
19. Validar el diseño mediante la aplicación de métricas.
20. Validar el software mediante la aplicación de pruebas de caja negra.

Se plantea la siguiente **idea a defender** para la investigación: *Si se desarrolla el módulo para el sistema de soporte a la toma de decisión multicriterios que implemente el método PROMETHEE se logrará eliminar la complejidad de la confección manual de los mismos y se evitará que el proceso sea susceptible a errores.*

Los métodos y técnicas empleados fueron los siguientes:

- **Métodos Teóricos**

Permiten descubrir en el objeto de investigación las relaciones esenciales y las cualidades fundamentales, no detectables de manera senso-perceptual. Por ello se apoya básicamente en los procesos de abstracción, análisis, síntesis, inducción y deducción. (2)

**Análisis-Síntesis:** Mediante este método se descompone un objeto, fenómeno o proceso en los principales elementos que lo integran para analizar, valorar y conocer sus particularidades. (3) Se utilizó para el análisis de la información y así poder llegar a conclusiones sobre el estudio realizado de métodos de toma de decisión multicriterios.

**Histórico-Lógico:** A través de este método se establece la necesaria correspondencia entre los elementos de los métodos lógico e histórico, proyectando el análisis de la evolución histórica de los fenómenos, con la proyección lógica de su comportamiento futuro. En esta investigación se le dio utilidad para establecer el desarrollo y tendencia de los sistemas de apoyo a la decisión.

**Modelación:** Este consiste en la representación ya sea material o teórica de los objetos, fenómenos, o particularidades de estos, lo que permite descomponerlos, abstraer determinadas cualidades, operar y experimentar con él. Para el entendimiento de la metodología SXP y el modelo propuesto se hace uso de este método de la investigación.

- **Métodos Empíricos**

Estos métodos posibilitan revelar las relaciones esenciales y las características fundamentales del objeto de estudio, accesibles a la detección de la percepción, a través de procedimientos prácticos con el objeto y diversos medios de estudio. (2)

**Observación:** Es un método para reunir información visual sobre lo que ocurre, lo que el objeto de estudio hace o cómo se comporta. (4) En este caso se utilizó para entender cómo se han utilizado en la universidad los métodos PROMETHEE en la solución de problemas de toma de decisión multicriterios.

**Análisis de documentos:** Se estudiaron los documentos relacionados con la metodología SXP y los métodos a implementar.

- **Técnicas de recopilación de información**

La recopilación de datos se utiliza para verificar los métodos empleados en lo investigado y llegar a la conclusión del suceso, teniendo las pruebas y una serie de pasos que se llevan a cabo para comprobar la hipótesis planteada. (5)

**Entrevistas:** Es una técnica para obtener datos, que consiste en un diálogo entre dos personas. (5) En este caso se empleó con el fin de obtener información referente a las herramientas a emplear, la metodología a implementar, además de obtener datos referentes a la experiencia de personas que han utilizado los métodos PROMETHEE en la solución de problemas.

El documento está organizado en cuatro capítulos a lo largo del documento, en los que se fundamenta la solución propuesta, se hace referencia al estudio de las tecnologías, las herramientas utilizadas para el desarrollo de la aplicación, así como el modelado del sistema, el análisis y diseño del sistema y la implementación.

## *Capítulo 1. Fundamentación teórica de la propuesta a desarrollar.*

En este capítulo se hace breve descripción del estudio realizado que sirvió de base para llevar a cabo la investigación y en el cual se menciona y caracteriza el método propuesto para dar solución a la problemática. Adicionalmente fue realizado un estudio de las principales tecnologías y herramientas a utilizar en el desarrollo de la aplicación Web.

## *Capítulo 2. Características del sistema a desarrollar*

En este capítulo se describen las características básicas y fundamentales del sistema informático a desarrollar. En el mismo, se realiza una breve descripción del problema en cuestión, especificándose los requisitos funcionales que plantea el cliente y que a su vez el sistema debe cumplir. Se da a conocer todo lo referente al modelado del negocio, incluyendo las historias de usuario del sistema y una breve descripción de cada uno de ellos, englobando todo lo referente al modelado del diseño específicamente.

## *Capítulo 3. Implementación y Prueba del sistema*

Este capítulo abarca el desarrollo de la fase de Implementación y Prueba del sistema. En el mismo figuran los diagramas del diseño y de secuencia de los casos de uso definidos en el capítulo anterior. Además se representa el modelo de despliegue del sistema así como el de componentes.



# Capítulo 1. Marco Teórico Referencial

---

## Capítulo I. Marco Teórico Referencial

Para el desarrollo de la presente investigación se hizo necesaria la revisión, el estudio y el análisis de diversos temas, que permitieron establecer el marco teórico referencial, para la obtención de los resultados que dan cumplimiento a los objetivos propuestos. Los principales temas abordados están relacionados con el objeto de estudio y campo de acción definidos en el diseño de la investigación.

En el presente capítulo se muestra el análisis realizado por los autores, a partir del estudio sobre la toma de decisión y sus sistemas de soporte. Se estudian los métodos a implementar para entender su funcionamiento y definir en qué circunstancias es más idóneo utilizarlos. Por último, para el desarrollo de la aplicación es necesario definir la metodología de desarrollo y herramientas, mostrándose en el último epígrafe la revisión y análisis realizado para ello.

### 1.1 Sistemas de Apoyo a la Toma de Decisión

Según (6) el proceso de toma de decisión es una de las actividades que se realizan con mayor frecuencia en el mundo de los negocios y en otras áreas de la vida en general. Se presenta en todos los niveles de la organización: desde asistentes o auxiliares, hasta los directores generales de las empresas y presidentes de instituciones y a niveles gubernamentales. En todos los casos se tiene uno o varios objetivos que cumplir, considerando un conjunto de restricciones. En general, los Sistemas de Apoyo a las Decisiones tienen como propósito fundamental apoyar, facilitar y viabilizar este proceso, a través de la obtención oportuna y confiable de información relevante. Es a su vez, una herramienta en el área de del conocimiento.

#### 1.1.1 Proceso de toma de decisión

“La toma de decisiones es un proceso de selección entre cursos alternativos de acción, basado en un conjunto de criterios, para alcanzar uno o más objetivos”. (7) Un proceso de toma de decisión comprende de manera general los siguientes pasos:

- Análisis de la situación
- Identificación y formulación del problema
- Identificación de aspectos relevantes que permitan evaluar las posibles soluciones
- Identificación de las posibles soluciones
- Aplicación de un modelo de decisión para obtener un resultado global
- Realización de análisis de sensibilidad

# Capítulo 1. Marco Teórico Referencial

---

La opinión de una única persona en la toma de decisión, puede tornarse insuficiente cuando se analizan problemas complejos, sobre todo aquellos cuya solución puede afectar a muchas otras personas. Debido a lo anterior se debe optar por generar discusión e intercambio entre actores, que por su experiencia y conocimiento pueden ayudar a estructurar el problema y a evaluar las posibles soluciones. Para abordar una situación de un problema de toma de decisión en la que se presentan diversos objetivos o criterios que simultáneamente deben incorporarse.

## 1.1.2 Sistemas de soporte a la toma de decisiones

Según (8) los sistemas de soporte a la toma de decisión DSS (Sistema de Soporte a la Decisión) por sus siglas en inglés (Decision Support System), son sistemas de información basados en computadoras los cuales combinan modelos para intentar resolver problemas no estructurados utilizando un interfaz amigable para el usuario.

Algunas clasificaciones de los **DSS**:

- Según la relación con los usuarios.
- Según el modo de asistencia.

**Según la relación con el usuario (9):**

- **Sistemas cooperativos:** permite al decisor modificar, completar o refinar las sugerencias ofrecidas por el sistema y las envía para su validación, haciendo que el proceso comience de nuevo hasta que se genere una solución consolidada.
- **Activos:** presenta sugerencias y soluciones explícitas.
- **Pasivos:** que no puede presentar sugerencias o soluciones explícitas.

**Según el criterio de asistencia (10):**

- **Impulsado por modelos:** hace énfasis en el acceso y manipulación de modelos estadísticos, financieros, de optimización o simulación. Utiliza datos y parámetros suministrados por usuarios. No se consideran muy dependientes de los datos.
- **Impulsado por Comunicación:** soporta a más de una persona trabajando en una tarea compartida.
- **Impulsado u orientado a Datos:** hace énfasis al acceso y manipulación de series temporales de datos internos de la empresa, y algunas veces, de datos externos.
- **Impulsado por Documentación:** administra, recupera y manipula información no estructurada en una variedad de formatos electrónicos.
- **Impulsado por Conocimiento:** suministra pericia en la solución de problemas especializados almacenados como hechos, reglas, procedimientos o en estructuras similares.

# Capítulo 1. Marco Teórico Referencial

---

## Funciones de los DSS:

- Permiten resolver gran parte de las limitaciones de los programas de gestión.
- Permiten extraer y manipular información de forma flexible.
- Permiten al usuario definir interactivamente qué información y cómo combinarla.
- Suelen incluir herramientas de simulación.

## Tipos de DSS:

- **Sistema de apoyo las decisiones de grupo (GDSS):** es un sistema que como el nombre lo dice apoyan un grupo de personas con un objetivo común.
- **Sistema de información gerencial (MIS):** también llamado Sistema de Información Administrativa (AIS).
- **Sistema de Información Ejecutiva (EIS):** son el tipo de DSS que más se usan en inteligencia de negocio.
- **Sistemas expertos basados en Inteligencia Artificial (SSEE):** los sistemas de expertos simulan el conocimiento de un experto y lo utilizan de forma efectiva para resolver un problema concreto.

### Beneficios y limitaciones. (11)

#### Los beneficios del uso de un SSD son:

- Amplía la habilidad del decisor para procesar y asimilar información.
- Amplía la habilidad del gerente para enfrentar problemas complejos.
- Reduce el tiempo asociado a la toma de decisiones.
- Mejora la confiabilidad del proceso de decisión, así como el resultado.
- Incentiva la exploración y descubrimiento de parte del decisor.
- Crea una ventaja competitiva sobre la competencia.

#### Mientras que entre las limitaciones del uso de un SSD se encuentran:

- No puede incluir talentos distintivos humanos como la creatividad, imaginación, intuición.
- Su desempeño está limitado al sistema computacional en el que funcione.
- Los lenguajes e interfaces no son aún lo suficientemente sofisticados para procesar lenguaje natural.
- Por su especialización, un DSS no puede llegar a usarse de manera generalizada en múltiples contextos de toma de decisiones.

# Capítulo 1. Marco Teórico Referencial

## 1.2 Los métodos de Toma de Decisión Multicriterios PROMETHEE.

Dentro de los Métodos de Relaciones de Superación, ocupan un lugar muy destacado los llamados Métodos PROMETHEE para la ayuda a la decisión multicriterios. Los Métodos de Relaciones de Superación en general, y los Métodos PROMETHEE en particular, admiten la existencia de alternativas incomparables. Debido a la naturaleza conflictiva de los criterios, muchas de las alternativas de un problema multicriterios son incomparables entre sí. Esto conduce al decisor a investigar bajo qué criterios las alternativas evidencian un buen comportamiento y bajo cuáles su desempeño es deficiente, para proceder luego a efectuar la mejor elección de acuerdo a su propio esquema de preferencias. La formulación de un problema multicriterios puede expresarse en los siguientes términos:

$$\text{Max } \{g_1(a), g_2(a), \dots, g_k(a) / a \in A\} \quad (1)$$

Donde A es un conjunto de alternativas factibles y  $\{g_j(\cdot), j=1, \dots, k\}$  es un conjunto de criterios de evaluación. Si bien se expresa como un problema de maximización, lo más normal es que algunos criterios deban maximizarse y otros minimizarse al mismo tiempo, lo cual no representa ningún obstáculo para su consideración. (12)

$$\left\{ \begin{array}{l} \left( \begin{array}{l} \forall j: g_j(a) \geq g_j(b) \\ \exists_h: g_h(a) > g_h(b) \end{array} \right) \Leftrightarrow a P b \text{ (} b \text{ es preferida por } a \text{)} \\ \left( \begin{array}{l} \exists_s: g_s(a) > g_s(b) \\ \exists_r: g_r(a) > g_r(b) \end{array} \right) \Leftrightarrow a R b \text{ (} a \text{ es preferida por } b \text{)} \\ \forall_j: g_j(a) = g_j(b) \Leftrightarrow a I b \text{ (} a \text{ es diferente con } b \text{)} \end{array} \right.$$

Así podrá identificar a las alternativas dominadas, indiferentes o incomparables entre sí. Aquellas alternativas que no son dominadas se denominan soluciones o alternativas eficientes. Sin embargo, la identificación de las alternativas eficientes no resuelve el problema al decisor, ya que es imposible concluir sin alguna información acerca de sus preferencias. En realidad, ningún problema multicriterios puede ser tratado adecuadamente si no se dispone de información adicional. (12)

### Los Métodos PROMETHEE y la información adicional.

Los métodos PROMETHEE tienen una ventaja muy clara con respecto al resto de sus homólogos usado para al mismo fin y es que la información que usan es clara, sencilla de obtener y definir por el decisor.

Fue diseñado y llevado a la práctica para tratar problemas multicriterios donde el conjunto de alternativas, A, es un conjunto finito de alternativas factibles. En este caso, el decisor se enfrenta con una matriz de decisión, que consiste en una tabla de evaluaciones del siguiente tipo:

# Capítulo 1. Marco Teórico Referencial

...	$g_1(\cdot)$	$g_2(\cdot)$	...	$g_j(\cdot)$	...	$g_k(\cdot)$
$A_1$	$g_1(a_1)$	$g_2(a_1)$	...	$g_j(a_1)$	...	$g_k(a_1)$
$A_2$	$g_1(a_2)$	$g_2(a_2)$	...	$g_j(a_2)$	...	$g_k(a_2)$
...	...	...	...	...	...	...
$A_i$	$g_1(a_i)$	$g_2(a_i)$	...	$g_j(a_i)$	...	$g_k(a_i)$
...	...	...	...	...	...	...
$a_n$	$g_1(a_n)$	$g_2(a_n)$	...	$g_j(a_n)$	...	$g_k(a_n)$

Es importante destacar que la matriz debe brindar la posibilidad de incluir nuevas alternativas a medida que se vaya evolucionando y también de eliminarlas en caso de que sea necesario, ya que durante el proceso de análisis de la información las alternativas pueden ir cambiando. La estructuración de la matriz se va alcanzando de forma progresiva. La información entre los distintos criterios consiste en el establecimiento de pesos que denoten la importancia relativa de cada uno de ellos entre sí, teniendo en cuenta la opinión del decisor. Los pesos se asumen siempre positivos y no existe ninguna objeción para considerar pesos normalizados. La labor de determinación de los pesos no resulta trivial ni sencilla debido a que la componente subjetiva existente es muy fuerte. (12)

Para cada criterio se define una función de preferencia particular  $P_j(\_, \_)$ , que indica el grado de preferencia asociado a la mejor alternativa en el caso de las comparaciones binarias, de acuerdo con la desviación entre las evaluaciones de las alternativas para ese criterio en particular. Así pues, para pequeñas desviaciones el decisor asignará una reducida preferencia a la mejor alternativa, mientras que para grandes desviaciones la preferencia será mayor. De esta forma, en los Métodos PROMETHEE se sugiere modificar la modelación de las preferencias del decisor, considerando para cada criterio, algunas posibles extensiones. Tales extensiones reciben el nombre de criterios generalizados. Un criterio generalizado se obtiene asociando a cada criterio  $g_j(\cdot)$  una función de **preferencia**  $P_j(\cdot, \cdot)$  que el decisor posee en mente, de forma tal que:

$$P_j(a, b) = P_j[d_j(a, b)] \quad \forall a, b \in A \text{ donde } d_j(a, b) = g_j(a) - g_j(b)$$

$$\text{siendo } 0 \leq P_j(a, b) \leq 1$$

Entonces, el par  $\{g_j(\cdot), P_j(\cdot, \cdot)\}$  se denomina criterio generalizado asociado al criterio  $g_j(\cdot)$ . (12)

Las funciones de preferencia definidas permiten trasladar las desviaciones observadas en la escala de un criterio específico en grados de preferencia. Una vez formulada la matriz de evaluaciones  $g_j(a_i)$ , y definidos los pesos  $w_j$  y los criterios generalizados asociados ya están creadas las condiciones para que el método PROMETHEE comience el proceso de decisión. (12)

# Capítulo 1. Marco Teórico Referencial

Los Métodos PROMETHEE I y II

El método PROMETHEE basa su proceso de toma de decisión en las comparaciones binarias de alternativas que permite la solución de distintos problemas. En la mayoría de los casos son problemas de ordenamiento si es el caso en que el decisor tiene que sacar la mejor alternativa dentro de un conjunto A de estas. Esto lo tiene que hacer buscando las alternativas desde la más débil hasta las más fuertes dentro del mismo conjunto A. De esta forma existen dos formas o técnicas para resolver el problema de ordenamiento PROMETHEE I y PROMETHEE II. Una vez asociadas las funciones de preferencias a cada criterio, deben definirse los índices de preferencia agregados o índices de preferencia multicriterios y los flujos de superación. (12)

## Índices de preferencia agregados o índices de preferencia multicriterios.

Un índice de preferencia multicriterios se obtiene de la siguiente forma:

$$\pi(a, b) = \sum_{j=1}^k P_j(a, b)W_j$$

Donde  $w_j$  es el peso que indica la relevancia o la importancia del criterio  $g_j(\cdot)$ . Para cada par de alternativas  $a$  y  $b$ ,  $\pi(a, b)$  expresa el grado de preferencia de  $a$  sobre  $b$ . Esto quiere decir que grado de preferencia de  $a$  con respecto a todas las preferencias  $b$  y  $\pi(b, a)$  en este caso quiere decir el grado de preferencia del criterio  $b$  que tiene con respecto a los demás criterios  $a$ . Estos números son usual y simultáneamente positivos y determinan una Relación de Superación Valorada sobre el conjunto A. (12)

## Flujos de Superación

Para cada nodo  $a$ , en el grafo de superación valorado, se define el flujo positivo o de salida:

$$\phi^+(a) = \sum_{n=1}^1_{b \in A} \pi(a, b)$$

Esta fórmula mide con que intensidad la alternativa es preferida a las  $(n-1)$  restantes, con ella se puede saber cuan fuerte es la alternativa en mi sistema de decisión. Simétricamente se define el flujo negativo o de entrada.

$$\phi^-(a) = \sum_{n=1}^1_{b \in A} \pi(b, a)$$

# Capítulo 1. Marco Teórico Referencial

Esta mide con que intensidad otras alternativas son preferidas a la alternativa  $a$ , en otras palabras es capaz de decir cuál es la debilidad de la alternativa  $a$  con respecto a las demás y viceversa. Así una alternativa será mejor con respecto a las demás en la medida que su flujo sea más positivo y peor cuando sea más negativo, esta es la base del ordenamiento del PROMETHEE I. (12)

## El ordenamiento parcial: PROMETHEE I

Teniendo los flujos de superaciones negativos y positivos estamos en condiciones de deducir dos pre órdenes de las alternativas que usualmente son diferentes. La intersección de dichos ordenamiento parcial es el origen del PROMETHEE I, que muestra una estructura de preferencias de pre orden parcial. (12)

$$\left\{ \begin{array}{l} aP^I b \Leftrightarrow \left\{ \begin{array}{l} \phi^+(a) > \phi^+(a) \text{ y } \phi^-(a) > \phi^-(a) \\ \phi^+(a) = \phi^+(a) \text{ y } \phi^-(a) < \phi^-(a) \\ \phi^+(a) > \phi^+(a) \text{ y } \phi^-(a) = \phi^-(a) \end{array} \right. \\ \\ aI^I b \Leftrightarrow \phi^+(a) = \phi^+(a) \text{ y } \phi^-(a) < \phi^-(a) \\ aR^I b \Leftrightarrow \text{cualquier otro caso} \end{array} \right.$$

Donde  $P^I$ ,  $I^I$  y  $R^I$  indican preferencias, indiferencias e incomparabilidad teniendo en cuenta la relación de preferencias del PROMETHEE I.

Este pre orden parcial es el que se le propone al decisor para que pueda considerar sus problema de decisión. Es válido destacar que si usamos el método PROMETHEE I alguna de las alternativas quedarán incomparables. Generalmente dos alternativas  $a$  y  $b$  son incomparables cuando  $a$  es buena bajo un conjunto de criterios para los cuales  $b$  es débil. Dado que la información correspondiente a ambos flujos no es consistente, parece natural considerar a las alternativas como incomparables. En este caso el método no está en la responsabilidad de decidir cuál de las alternativas es mejor sino que es el decisor quien lo hace. (12)

## El ordenamiento completo: PROMETHEE II

Es normal que el decisor quiera realizar un ordenamiento completo de las alternativas como parte del proceso. En este caso un pre orden completo son las alternativas más señaladas para alcanzar la decisión. Este se fundamenta con el flujo de superación neta de cada alternativa:

$$\phi(a) = \phi^+(a) - \phi^-(a)$$

Cada flujo neto puede calcularse por la diferencia entre el flujo positivo y el flujo negativo, mientras mayor sea el flujo neto mejor será la preferencia de esta alternativa sobre las demás. De esta forma se define el ordenamiento completo en PROMETHEE II. (12)

# Capítulo 1. Marco Teórico Referencial

---

$$\begin{cases} aP^{II} b \Leftrightarrow \phi(a) > \phi(b) \\ aI^{II} b \Leftrightarrow \phi(a) = \phi(b) \end{cases}$$

Con esto ya todas las alternativas del conjunto A ya han sido completamente ordenadas, pero la información resultante es más dudosa ya que ha ocurrido una pérdida entre el balance y el flujo neto. Tanto como PROMETHEE I y PROMETHEE II ayudan al decisor a tomar las decisiones dejando clara visión entre las alternativas. (12)

### 1.3 Sistemas que implementan PROMETHEE.

Existen diversos sistemas de ayuda a la toma de decisión multicriterios como, por ejemplo, el PROMCALC el cual es un software que respalda a la metodología PROMETHEE-GAIA y contiene todos los cálculos del PROMETHEE (PROMethee CALCulation). Es un software particularmente sencillo para los usuarios, está escrito en turbo pascal y ofrece un menú muy interactivo. Para aquellos que poseen un profundo y razonable conocimiento de la metodología no es necesario recurrir a un manual.

El software Decisión Lab 2000 está diseñado para las decisiones multicriterios. Este utiliza dos métodos para la toma de decisiones, el PROMETHEE y GAIA (análisis geométrico para el aprendizaje interactivo). Ambos métodos se consideran entre los más fáciles y eficientes. Es una herramienta muy útil en la toma de decisiones ya que es fácil de integrar en la mayoría de los negocios, dado que está diseñado para su uso con todas las formas plataformas de Windows. Decisión Lab 2000 también es fácil de aprender por ser muy similar a Microsoft Excel, también permite al usuario copiar los conjuntos de datos a Microsoft Excel.

Por otra parte se tiene una aplicación para la selección de un sistema de riego el cual propone una solución al problema de la distribución de agua para tal fin en la cuenca Chorrillos-Bárcena perteneciente al Perú. En la misma se emplearon dos métodos de análisis multicriterios. Un método multicriterios de selección que fortalece y amplía los resultados técnicos fue obtenido a través de la revelación de las preferencias de usuarios involucrados en la toma de decisiones frente a los usos del agua en el caso específico de la subcuenca mencionada. Los métodos de evaluación y decisión multicriterios seleccionados para este trabajo fueron AHP y PROMETHEE.

Existe además una herramienta de soporte a la toma decisión en la comercialización del crudo de la empresa petrolera PETROECUADOR, particularmente para el caso del Crudo de Oriente. En este caso es una investigación basada en que las decisiones de la industria del petróleo determinan la dirección y el curso de miles de millones de dólares cada año. La complejidad de una decisión puede variar desde una simple pregunta hasta niveles que requieren gran detalle. Tomar decisiones complicadas puede tomarle a una empresa meses o años de preparación.



# Capítulo 1. Marco Teórico Referencial

---

Es por ello, que dicha empresa estatal petrolera al no contar con ningún sistema informático que ayude a los funcionarios encargados de la toma de decisiones a fijar el precio óptimo con el cual se deberían realizar las negociaciones de crudo, vio la necesidad de contar con una herramienta de soporte a la toma de decisiones en la comercialización del mismo, particularmente para el caso del Crudo Oriente. Para resolver la necesidad de PETROECUADOR, se resolvió implementar un sistema de toma de decisiones orientado a la negociación del petróleo a corto plazo, específicamente para la venta a un mes, para lo cual, primero se tuvo que investigar y estudiar algunos temas antes de su implementación. Por este motivo, el presente proyecto es de carácter investigativo que concluye con su aplicación práctica.

Todas estas aplicaciones tienen algo en común y es el hecho de que están hechas con un fin específico, es decir, de antemano se sabe cuál va a ser su utilización, sin embargo en esta investigación se pretende realizar un sistema similar pero con la particularidad de que sea global, que sirva para indeterminados usos y necesidades de la vida, en la cual predefiniendo ciertos y determinados parámetros se ajustan a las necesidades de decisión, para de esta manera proporcionarle toda la información en cuanto a la situación se requiera, es por esto que no existe actualmente un software que cumpla con todas las expectativas. De aquí que la UCI se dio a la tarea de implementar un sistema de ayuda a la toma de decisión multicriterios que satisfaga las exigencias planteadas.

## 1.4 Herramientas y Metodologías de Desarrollo.

Es de suma importancia para el adecuado desarrollo de un software realizar, en su fase de concepción, la selección de las herramientas a utilizar, mitigando de esta forma cualquier problema existente con el tiempo o la calidad del software. En Cuba existe la política referente al uso de herramientas libres debido a la imposibilidad de adquirir software con licencias privadas, por lo que se han analizado una serie de herramientas que respondan a esta condición, además de lenguajes de programación y metodologías de desarrollo de software que respondan a las necesidades de la aplicación a implementar.

### 1.4.1 Metodologías para el desarrollo

**SXP**, es una metodología compuesta por las metodologías ágiles (XP y SCRUM), que brinda una estrategia desde el punto de vista tecnológico al introducir procedimientos ágiles que permiten actualizar los procesos de software. Consiste en una programación rápida o extrema, cuya novedad es tener como parte del equipo, al usuario final. Esta metodología toma las buenas prácticas de SCRUM para la gestión, logrando una buena planificación y organización. Por otra parte usa XP para todo el proceso de desarrollo que garantiza un proceso de desarrollo completo. Esta metodología consta de las siguientes fases:

1. Planificación-Definición
2. Desarrollo
3. Entrega

# Capítulo 1. Marco Teórico Referencial

## 4. Mantenimiento

La fase de Planificación-Definición es la etapa inicial del proyecto, en donde se define el alcance del mismo y se realiza la planificación financiera. En la fase de Desarrollo se implementa el sistema hasta que esté listo para su entrega y en la de Mantenimiento se le brinda soporte al cliente. De estas 4 fases y 7 flujos se generan un conjunto de artefactos los cuales se muestran en la siguiente imagen:

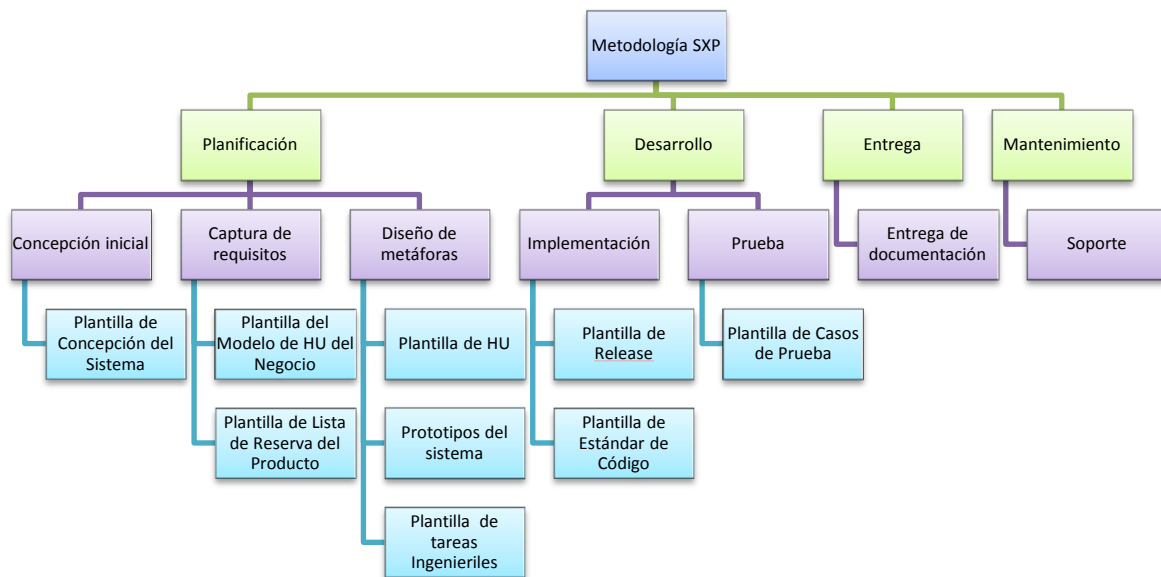


Figura 1: Esquema de la metodología SXP.

En la investigación se decidió usar SXP por las características que la describe ya que logra hacer de un proceso de desarrollo un proceso eficiente y eficaz. (13) (14)

### 1.4.2 Lenguaje de Modelado

El Lenguaje Unificado de Modelado (UML) es un lenguaje visual para especificar, construir y documentar los artefactos de sistemas intensivos en software. (15) Prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación. UML no es un método de desarrollo, lo que significa que no sirve para determinar qué hacer en primer lugar o cómo diseñar el sistema, sino que simplemente le ayuda a visualizar el diseño y a hacerlo más accesible para otros. UML está controlado por el grupo de administración de objetos (OMG) y es el estándar de descripción de esquemas de software. (16)

# Capítulo 1. Marco Teórico Referencial

---

## 1.4.3 Herramientas Case

### 1.4.3.1 Visual Paradigm

Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. También proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. Presenta licencia gratuita y comercial. Es fácil de instalar y actualizar y compatible entre ediciones. Generación de bases de datos; Transformación de diagramas de Entidad-Relación en tablas de base de datos; Importación y exportación de ficheros XMI. ("Visual Paradigm,").

## 1.4.4 Lenguajes utilizados

### 1.4.4.1 Java

Java es un lenguaje de programación libre orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. En la actualidad Java se ha convertido en uno de los lenguajes más usados y más demandados por los desarrolladores. Es independiente de la plataforma de desarrollo. Existen dentro de su librería clases gráficas como awt y swing, las cuales permiten crear objetos gráficos comunes altamente configurables y con una arquitectura independiente de la plataforma. Existen varias clases que contienen las abstracciones básicas para facilitar a los programas una gran capacidad de representación. Para ello se contará con un conjunto de clases comunes que pueden crecer para admitir todas las necesidades del programador. Java permite a los desarrolladores aprovechar la flexibilidad de la Programación Orientada a Objetos en el diseño de sus aplicaciones. Se puede acceder a bases de datos fácilmente con JDBC, independientemente de la plataforma utilizada. El manejo de las bases de datos es uniforme, es decir transparente y simple. En cuanto a la seguridad Java es muy confiable. (17)

## 1.4.5 Marcos de trabajo

### 1.4.5.1 Vaadin

Vaadin es un framework para el desarrollo de aplicaciones web Open Source modernas o también llamadas RIA (Rich Internet Applications) que posee una librería con numerosos componentes de usuario con los que construir aplicaciones web profesionales similares a los interfaces de una aplicación de escritorio. Este framework se ejecuta del lado del servidor lo que significa que la mayoría de la lógica y por tanto la mayor carga del trabajo recae en este. Vaadin que utiliza tecnología Ajax del lado del cliente cuya base es GWT (Google Web Toolkit) para renderizar la interfaz de usuario en el navegador. Las aplicaciones Google Web Toolkit se escriben en Java y son automáticamente "traducidas" a JavaScript y HTML. GWT es muy cómodo para el

# Capítulo 1. Marco Teórico Referencial

desarrollador porque le mantiene al margen de estas acciones de presentación y sin la necesidad de instalar ningún plug-in para su presentación. (18)

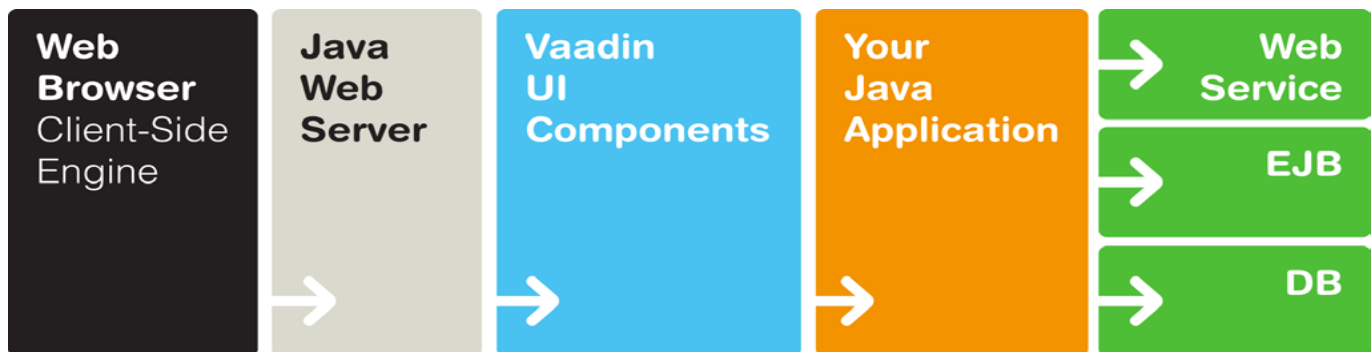


Figura 2: Arquitectura cliente-servidor de Vaadin

## 1.4.5.2 Spring

Según (19) se evidencia que Spring es un marco de trabajo de código abierto para el desarrollo de aplicaciones para la plataforma Java. Es el más popular y ambicioso de todos los marcos de trabajo de peso ligero de este lenguaje. EL mismo interviene en todas las capas arquitectónicas de una aplicación realizada en Java. Teniendo en cuenta que su diseño brinda flexibilidad arquitectónico comenta que los principales valores de Spring, se pueden resumir en: No es agresivo, provee un modelo consistente de programación, promueve la reusabilidad del código, facilita el diseño POO en aplicaciones Java, permite la extracción de valores de configuración desde el código java a archivos XML o archivos de propiedades. Diseñado para ser usado por las aplicaciones para que las pruebas sean lo más fácil posible. Spring hace de soluciones existentes un uso más fácil, dentro de una arquitectura consistente.

Teniendo en cuenta que es de código abierto, siendo esta una característica muy importante para los desarrolladores, debido a que es la utilización de herramientas libres una de las normativas de la UCI. Se decide utilizar, para la capa del modelo de negocio, Spring, por las facilidades que brinda para el desarrollo de la solución. (20)

## 1.4.5.3 Hibernate

Según (21) , Hibernate es un marco de trabajo objeto/relacional y un generador de sentencias sql, liberando al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias. Permite diseñar objetos persistentes que pueden incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos. Se integra a todo tipo de aplicación. Mantiene la

# Capítulo 1. Marco Teórico Referencial

---

portabilidad entre todas las bases de datos con un ligero incremento en el tiempo de ejecución, ofrece también un lenguaje de consulta de datos llamado HQL (*Hibernate Query Language, por sus siglas en inglés*), al mismo tiempo que un API para construir las consultas programáticamente (conocida como "Criteria").

Por decisión del equipo de desarrollo el marco de trabajo a utilizar para la capa de acceso a la base de datos es Hibernate, basándose en cómo genera esta herramienta las sentencias SQL y la posibilidad de liberar al desarrollador del manejo manual de los datos que resultan de dicha generación.

## 1.4.6 Entornos de Desarrollo Integrado (IDEs)

### 1.4.6.1 Eclipse

Eclipse es un entorno de desarrollo integrado de código abierto, multiplataforma para desarrollar "Aplicaciones de Cliente Enriquecido" proporcionando una interfaz gráfica, escrita con una sintaxis basada en XML, inverso a las aplicaciones "Clienteliviano" que son basadas en navegadores.. La interfaz de usuario de Eclipse tiene una capa intermedia Interfaz Gráfica de Usuario, conocida también como GUI (Graphical User Interface) llamada JFace, la cual facilita la construcción de aplicaciones basadas en SWT (Standard Widget Toolkit). (22) Eclipse utiliza módulos (en inglés *plug-in*) para facilitar toda su funcionalidad al frente de la plataforma de cliente enriquecido, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, aunque el usuario no las necesite. Con respecto a las aplicaciones clientes, abastece al programador con marcos de trabajo muy ricos para las aplicaciones gráficas, definición y manipulación de modelos de software, aplicaciones web, entre otros. (23)

## 1.4.7 Gestores de Base de Datos

### 1.4.7.1 PostgreSQL

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia de Distribución de Software Berkeley (BSD, Berkeley Software Distribution, según sus siglas en inglés). PostgreSQL implementa el uso de retrocesos, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz. Posee la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos. Tiene mejor soporte para disparadores y procedimientos en el servidor.

# Capítulo 1. Marco Teórico Referencial

---

## 1.4.8 Servidor de Aplicaciones

### 1.4.8.1 Apache Tomcat

Según (24) Apache Tomcat es un servidor Web con soporte para servlets y JSPs. Trae consigo el compilador Jasper, que compila JSPs convirtiéndolas en servlets, el mismo puede funcionar como servidor Web por sí mismo. Es gratis, fácil de instalar, se ejecuta en máquinas más pequeñas y presenta compatibilidad con las API (Interfaz de Programación de Aplicaciones) por sus siglas en inglés (*Application Programming Interface*) más recientes de Java. Ocupa poco espacio, presentando su código binario un megabyte de tamaño, logrando con ello que se ejecute de manera rápida. Otra característica de Tomcat es que es muy fiable. (25)

### Conclusiones Parciales

- Los sistemas de soporte a la decisión permiten aumentar la capacidad de los decisores para tomar una decisión más acertada utilizando menor tiempo y costo.
- La implementación del método PROMETHEE en un sistema de soporte a la toma de decisión multicriterios garantizará una mayor aceptación en el proceso de toma de decisión.
- De las fases definidas en la metodología para la ubicación se implementan como parte de un sistema de soporte a la decisión, actividades de las fases de ejecución y ubicación.
- Para el desarrollo del sistema se empleará la metodología SXP y la herramienta CASE Visual Paradigm para la descripción y modelación del sistema.
- Como sistema gestor de base de datos se utilizará el PostgreSQL.
- La implementación del sistema se hará empleando IDE Eclipse y en la implementación de cada una de las capas los marcos de trabajo: Hibernate para el acceso a datos, Spring para el negocio y Vaadin para la capa de presentación.
- Para el control de versiones se define SVN y como servidor de aplicaciones el apache Tomcat.

# Capítulo 2. Características del Sistema

---

## Capítulo 2. Características del Sistema

En el presente capítulo se describirán las características del sistema propuesto para darle solución al problema antes mencionado. Se definen conceptos y se identifican las entidades relacionadas con el modelo del dominio, el cual permite determinar los requisitos necesarios que la aplicación debe cumplir para darle solución a las insuficiencias que lo causaron. La implementación de la propuesta se realizará bajo las reglas y principio de la metodología SXP, así como las herramientas y tecnologías definidas. En el Capítulo 1 se describieron las fases de la metodología a usar, así como los artefactos que genera. Posteriormente se realizará la descripción de los artefactos correspondientes a la fase de **Planificación** y **Definición**.

### 2.1 Descripción del problema

Por las características de un proceso de toma de decisión, es fundamental destacar que un error en el mismo provocaría desconfianza y baja aceptación, e incluso en muchas ocasiones ésta corre riesgos de gran escala, tales como bienes materiales y hasta vidas humanas. En la historia se han visto casos donde una incorrecta toma de decisión ha conducido a grandes pérdidas. También se ha demostrado que en su proceso donde el análisis está referido a una gran cantidad de volúmenes de información, a tener en cuenta, es realizado de forma manual, trae como consecuencia errores, los plazos de tiempo necesarios son muy grandes e incluso se puede tornar impracticable. De aquí que la premisa será cómo implementar un sistema de toma de decisión que permita reducir los tiempos e impedir que la misma sea susceptible a errores.

### 2.2 Solución Propuesta

Se tiene como punto de partida la necesidad que tiene la Universidad de las Ciencias Informáticas de implementar un sistema de soporte a la toma de decisión que sea capaz de adaptarse a diferentes tipos de problemas, así como, que tenga en cuenta los criterios de los especialistas (expertos) en su proceso. Se cuenta para ello, con las metodologías y herramientas para el modelado y desarrollo de aplicaciones informáticas, y se hace necesario desarrollar una herramienta que implemente los algoritmos necesarios para la aplicación de la metodología y el modelo basado en las características de un sistema de soporte a la decisión. Esto permitirá el procesamiento de los grandes volúmenes de información necesarios, mitigando la complejidad de la aplicación manual del método y evitando que el sistema sea susceptible a errores.

### 2.3 Concepción inicial del sistema

En la etapa de concepción del sistema se genera el artefacto necesario para la misma, especificando los aspectos generales organizativos, su objetivo, principales involucrados y otros aspectos que permiten la posterior organización del desarrollo de la aplicación. Esta plantilla se muestra en el anexo 1.

# Capítulo 2. Características del Sistema

## 2.4 Captura de requisitos

La Ingeniería de Requisitos, es el proceso de desarrollar una especificación de Software. Las especificaciones pretenden comunicar las necesidades del sistema del cliente a los desarrolladores del sistema. Trata de los principios, métodos, técnicas y herramientas que permiten descubrir, documentar y mantener los requisitos para sistemas basados en computadora, de forma sistemática y repetible. (5) Para la extracción de los requisitos se emplearon las herramientas y métodos existentes garantizando una correcta redacción y análisis de los mismos.

Luego de realizar entrevistas y posteriormente la discusión con el cliente para determinar los requisitos del sistema en las posibles áreas donde se aplicará la herramienta, se lograron extraer los requisitos. Como resultado de la aplicación de estas técnicas se obtuvo la plantilla de modelos de historias de usuarios del negocio y la lista de reserva del producto, elementos que son descritos en los siguientes epígrafes. Estos elementos que constituyen los requisitos de la aplicación a implementar fueron validados por el cliente, partiendo de que se describen estos artefactos de conjunto, desarrolladores y cliente.

Luego se sometieron a 3 rondas de revisiones lo cual permitió llegar a un consenso y claridad de las historias de usuario y sus descripciones.

### 2.4.1 Historias de usuarios del negocio

En la plantilla del modelo de historias de usuarios del negocio se describen los actores y trabajadores del negocio, además presenta un diagrama de historias de usuarios del negocio que permite ver la relación entre los usuarios y las actividades que se realizan, también facilita la comunicación del cliente ya que se usa un lenguaje que el cliente es capaz de entender, de ahí que se puedan obtener de ello los requisitos. Esta plantilla se describe en el anexo 2.

### 2.4.2 Lista de reserva del producto (LRP)

La lista de reserva del producto, es un artefacto generado en la captura de requisitos, donde se describen los mismos como funcionalidades que el sistema debe cumplir en su desarrollo. Para su construcción se cuenta con la colaboración del gerente del proyecto, el cliente y miembros del equipo. Estos son descritos en la siguiente tabla

Prioridad	Ítem	Descripción	Estimación en sprint	Estimado por	Asignado a
<b>Requisitos Funcionales</b>					
<b>Muy Alta</b>					
	1	Autenticación de usuarios	1	programador	Reinier Torres
	2	Crear Decisión	3	programador	Javier Padrón
	3	Gestionar Criterios	3	programador	Reinier Torres
	4	Gestionar Alternativas	3	programador	Javier Padrón
	5	Evaluar las Alternativas en los criterios	3	programador	Reinier Torres



# Capítulo 2. Características del Sistema

Alta					
6	Generar Criterio	3	programador	Javier Padrón	Reinier Torres
Media					
7	Gestionar usuarios	3	programador	Javier Padrón	
8	Seleccionar Decisión	3	programador	Reinier Torres	
Requisitos No Funcionales					
Usabilidad					
1	El software tendrá siempre la posibilidad de ayuda disponible para cualquier tipo de usuario, lo que le permitirá un avance considerable en la explotación de la aplicación en todas sus funcionalidades.		programador	Javier Padrón	
2	El tiempo de entrenamiento requerido para que usuarios normales y avanzados sean productivos operando el sistema es de 1 día.		programador	Javier Padrón	
3	Debe poseer una interfaz agradable para el cliente.		Diseñador	Javier Padrón	
Fiabilidad					
4	El sistema estará disponible 24 horas del día, los siete días de la semana.		programador	Reinier Torres	
5	La herramienta de implementación a utilizar tiene soporte para recuperación ante fallos y errores.		programador	Reinier Torres	
Eficiencia					
6	Tiempo de respuesta promedio a las peticiones que se realizan al servidor no deberá ser mayor de 3 segundos.		programador	Javier Padrón	
7	El número de clientes o transacciones que el sistema puede alojar es de 2000.		programador	Javier Padrón	
Seguridad					
8	Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.		programador	Reinier Torres	
9	El sistema debe mantener en todo momento la seguridad de la		programador	Reinier Torres	

# Capítulo 2. Características del Sistema

	información asegurando la autenticidad de la misma.		
10	El sistema debe garantizar la confidencialidad, integridad y disponibilidad de la información que se procese en el sistema.	programador	Reinier Torres
11	El control de acceso se establecerá por roles que se le asignarán a los usuarios que interactúen con el sistema.	programador	Reinier Torres
<b>Soporte</b>			
12	Soporte para grandes volúmenes de datos y velocidad de procesamiento.	programador	Javier Padrón
13	Tiempo de respuesta rápido en accesos concurrentes.	programador	Javier Padrón
<b>Restricciones de diseño</b>			
14	El lenguaje de programación es Java.	programador	Reinier Torres
15	Los marcos de trabajo de desarrollo son Hibernate 5.0, Vaading 6.8 y Spring 3.0.2.	programador	Reinier Torres
16	La herramienta IDE de desarrollo utilizada será Eclipse STS 2.5.0.	programador	Reinier Torres
17	La herramienta case utilizada es Visual Paradigm 8.0 para modelado.	programador	Reinier Torres
18	La herramienta gestor de base de datos es el PostgreSQL 9.1	programador	Reinier Torres
<b>Interfaz</b>			
19	El sistema tiene que ofrecer una interfaz amigable, fácil de operar.	Diseñador	Javier Padrón
20	Diseño sencillo, con pocas entradas, permitiendo que no sea necesario mucho entrenamiento para que los usuarios puedan utilizar el sistema.	Diseñador	Javier Padrón

Tabla 1: Lista de Reserva del Producto

# Capítulo 2. Características del Sistema

## 2.5 Diseño de metáforas

A partir de la definición de las funcionalidades descritas en la lista de reserva del producto (LRP) es posible establecer las historias de usuarios, prototipos del sistema y las tareas ingenieriles que permiten su desarrollo siendo estas las actividades que se realizan y se describen a continuación.

### 2.5.1 Historias de Usuario

Básicamente una historia es una lista priorizada de requisitos o funcionalidades, descritas usando la terminología del cliente o en otras palabras un lenguaje que el cliente es capaz de comprender. Estas historias de usuario se especifican en el anexo 3. Seguidamente se presentan dos de ellas para dar continuidad en la explicación de las tareas ingenieriles definidas.

Historia de Usuario	
Número: HU_1	Nombre Historia de Usuario: Autenticar usuarios
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Reinier Torres	Iteración Asignada: Sprint 1
Prioridad en Negocio: <i>Muy Alta</i>	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: <i>El usuario accede al sistema insertando un usuario y su contraseña, posteriormente se procede a comprobar que estos datos son correctos, al serlo la aplicación le dará al usuario acceso a las funcionalidades a las cuales tiene permisos.</i>	
Observaciones:	
Prototipo de interfaces:	
	

Tabla 2: HU Autenticación de usuarios

# Capítulo 2. Características del Sistema

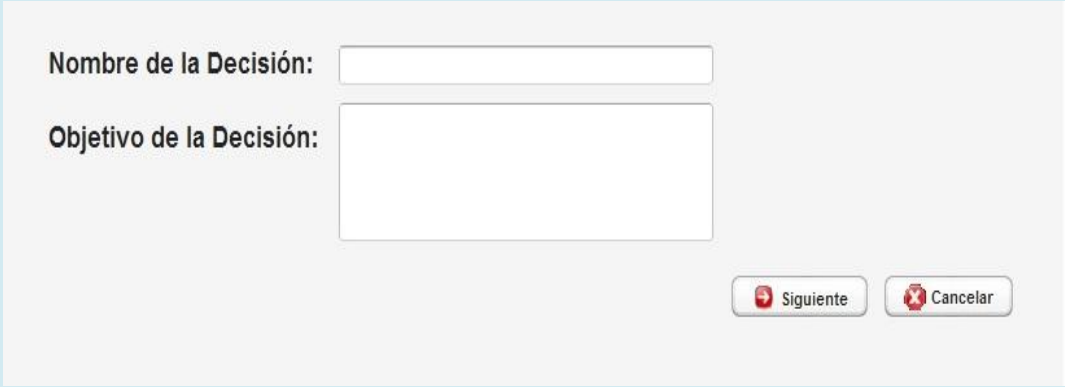
Historia de Usuario	
Número: HU_3	Nombre Historia de Usuario: Crear decisión
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Reinier Torres	Iteración Asignada: Sprint 3
Prioridad en Negocio: Muy Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Alto	Puntos Reales: 3
Descripción: <i>El usuario especialista crea una decisión, introduciendo el nombre de la decisión y una descripción .Este es el primer paso de una serie de pasos lógicos.</i>	
Observaciones:	
Prototipo de interfaces:	
	

Tabla 3: HU Cargar Alternativas

## 2.5.2 Tareas Ingenieriles

Partiendo de que ya se tienen predefinidas las historias de usuarios se establecen un grupo de tareas ingenieriles para cada una de ellas, que sirven de guía para el posterior desarrollo de la solución propuesta.

Algunas de las tareas ingenieriles están relacionadas con las historias de usuario, ya que estas están encaminadas en el montaje de las condiciones de desarrollo de la propuesta:

## Capítulo 2. Características del Sistema

Tarea de Ingeniería	
Número Tarea: T_1	Número Historia de Usuario: todas
Nombre Tarea: Generación de la base de datos	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 15-enero -2013	Fecha Fin: 20- enero- 2013
Programador Responsable: Reinier Torres	
Descripción: <i>Se genera la base de datos a partir del diseño realizado, esta debe implementarse sobre el sistema gestor definido para el desarrollo de la aplicación.</i>	

**Tabla 4: Tarea Ingenieril 1**

Otras de las tareas ingenieriles están asociadas a cada historia de usuario, en particular, para poder dar cumplimiento a la implementación de las funcionalidades definidas en ellas, por ejemplo las relacionadas con la HU\_1 son:

Tarea de Ingeniería	
Número Tarea: T_4	Número Historia de Usuario: HU_1
Nombre Tarea: Diseñar las interfaces requeridas para la funcionalidad autenticación de usuario.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 21-Febrero-2013	Fecha Fin: 28-Marzo-2013
Programador Responsable: Javier Padrón	
Descripción: Crear las interfaces necesarias, teniendo en cuenta los prototipos definidos en la Historia de Usuario.	

**Tabla 5: Tarea Ingenieril 4**

Las tareas ingenieriles definidas se encuentran en la plantilla del anexo 4.

# Capítulo 2. Características del Sistema

---

## 2.6 Arquitectura seleccionada

La programación de n capas es una arquitectura cliente-servidor con el objetivo primordial de separar la lógica de negocios de la lógica de diseño; un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario. La arquitectura de tres capas es un diseño reciente que introduce una capa intermedia en el proceso. Cada capa es un proceso separado y bien definido corriendo en plataformas separadas. En la arquitectura tradicional de tres capas se instala una interfaz de usuario en la computadora del usuario final (el cliente). La arquitectura basada en Web transforma la interfaz de búsqueda existente (el explorador de Web), en la interfaz del usuario final. (26)

En general, suele plantearse esta visión sobre tres niveles o capas:

**La capa de presentación:** Esta capa se encarga de proveer una interfaz entre el sistema y el usuario. Básicamente, se responsabiliza de que se le comunique información al usuario por parte del sistema y viceversa, manteniendo una comunicación exclusiva con la capa de negocio. Además dentro de esta capa entraría aquello que el usuario “ve” cuando se conecta a la aplicación.

**La capa de negocio:** Es la capa que contiene los procesos a realizar con la información recibida desde la capa de presentación, las peticiones que el usuario ha realizado, y responsabilizados de que se le envíen las respuestas adecuadas a la capa de presentación. Se podría verla como una capa intermedia, a medio camino entre la capa de presentación y la capa de datos, puesto que se relaciona con ambas y por supuesto, procesa también la información devuelta por la capa de datos.

**La capa de datos:** Por último, la capa donde se almacenan los datos. Mediante la capa de negocio, se puede encargar de ofrecer, modificar, almacenar, borrar y recuperar datos, mediante el gestor (o los gestores) de bases de datos que la aplicación requiera.

## 2.7 Patrones de diseño

El uso de patrones para el desarrollo de aplicaciones informáticas, es una buena práctica, aportando seguridad y robustez en la misma. Además de facilitar el trabajo, son de gran importancia en todos los proyectos informáticos, garantizando claridad en la estructura de la aplicación y como consecuencia mayor organización. El sistema que se desea implementar basa su arquitectura empleando patrones arquitectónicos y de diseños, en la definición de las clases del sistema y el diseño del sistema a implementar, compuesta por una arquitectura en tres niveles o capas. Para ello tiene suma importancia la utilización de los

# Capítulo 2. Características del Sistema

---

patrones GRASP. En la tecnología de objetos un Patrón es una descripción de un problema y la solución, a la que se le da un nombre, y que se puede aplicar a nuevos contextos. (27)

## 2.7.1 Patrones GRASP (Patrones de Software para la Asignación General de Responsabilidad)

“Los patrones **GRASP** describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades. Constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable.” (20)

El uso de estos patrones es la clave para garantizar en la aplicación final un producto de calidad y de mayores servicios. Además permite disminuir la complejidad de los mantenimientos al sistema. Estos patrones se describen a continuación:

**Bajo acoplamiento:** El Bajo Acoplamiento es un patrón evaluativo que el diseñador aplica al juzgar sus decisiones de diseño. Estimula asignar una responsabilidad de modo que su colocación no incremente el acoplamiento tanto que produzca los resultados negativos propios de un alto acoplamiento. (20) Que tiene como objetivo fundamental dar soporte a las bajas dependencias y al incremento de la reutilización. Y que brinda como solución diseñar con el objetivo de tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases

**Alta Cohesión:** Asignar una responsabilidad de modo que la cohesión siga siendo alta. Representa una Clase con responsabilidades moderadas en un área funcional, colaborando con otras para concretar tareas. Diseño más claro y comprensible. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. (20) Este patrón consigue mantener manejable la complejidad, asignando responsabilidades de manera que la información que almacena una clase sea coherente y esté relacionada con la clase.

**Experto:** Es un principio básico que suele utilizarse en el diseño orientado a objetos. Con él no se pretende designar una idea oscura ni extraña; expresa simplemente la "intuición" de que los objetos hacen cosas relacionadas con la información que poseen. Ofrece una analogía con el mundo real. (20) El mismo logra mitigar el problema de saber qué responsabilidad delegar a cada objeto para lo cual asigna responsabilidad al experto en información; la clase que tiene la información necesaria para llevar a cabo la responsabilidad.

**Creador:** El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento, se debe buscar una clase de objeto que agregue, contenga y realice otras operaciones sobre este tipo de instancias. (20) Este patrón es capaz darle solución a quién debería ser responsable de crear una nueva instancia, creando una nueva instancia por la clase que tiene la información necesaria para realizarla.

# Capítulo 2. Características del Sistema

## Otros patrones utilizados:

**Patrón Fachada:** Según (28) es un patrón de diseño de tipo estructural. Proporciona una interfaz unificada de alto nivel para un subsistema, que oculta las interfaces de bajo nivel de las clases que lo implementan. Con esto se consiguen dos objetivos fundamentales: hacer el subsistema más fácil de usar y desacoplar a los clientes de las clases del subsistema. En el diagrama de clases se especifica cómo se materializa el uso de estos patrones en el diseño realizado.

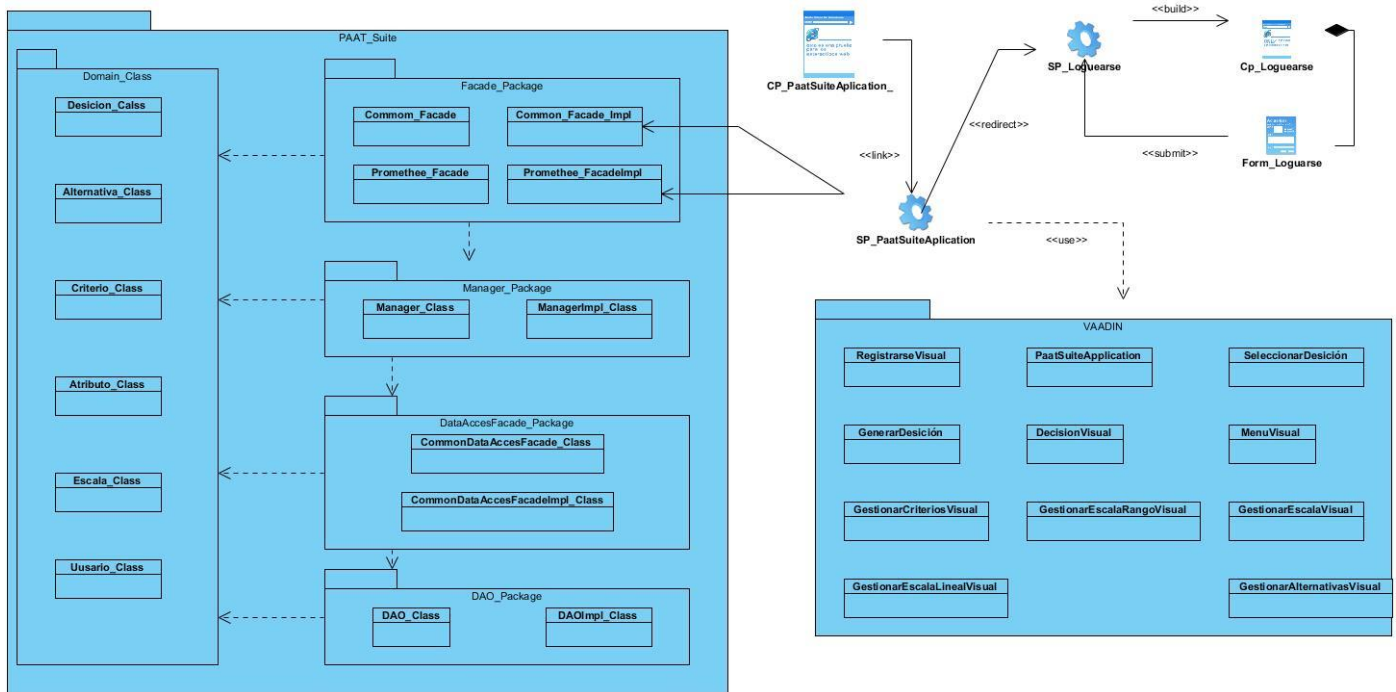


Figura 3: Diagrama de Clases

## 2.8 Modelo de Datos

Es un conjunto de conceptos que permiten describir, a distintos niveles de abstracción, la estructura de una base de datos, la cual se denomina esquema. Según el nivel de abstracción, el modelo de datos que permite su descripción será un modelo externo, global o interno, cada uno de los cuales ofrece distintos elementos de descripción. Los modelos externos permiten representar los datos que necesita cada usuario en particular con las estructuras propias del lenguaje de programación que va a emplear. Los modelos globales ayudan a describir los datos para el conjunto de usuarios, se podría decir que es la información a nivel de empresa; y, por último, los modelos internos que están orientados a la máquina, siendo sus elementos de descripción punteros, índices, agrupamientos. (29)



# Capítulo 2. Características del Sistema

Existen 3 tipos de modelos de datos: (29)

- **Los modelos conceptuales** (también denominados de alto nivel) facilitan la descripción global del conjunto de información de la empresa con independencia de la máquina, por lo que sus conceptos son cercanos al mundo real. (29)
- **Los modelos convencionales** se encuentran soportados por los Sistemas de Gestión de Bases de Datos (SGBD) y están orientados a describir los datos a nivel lógico para el SGBD, por lo que sus conceptos son propios de cada SGBD. (29)

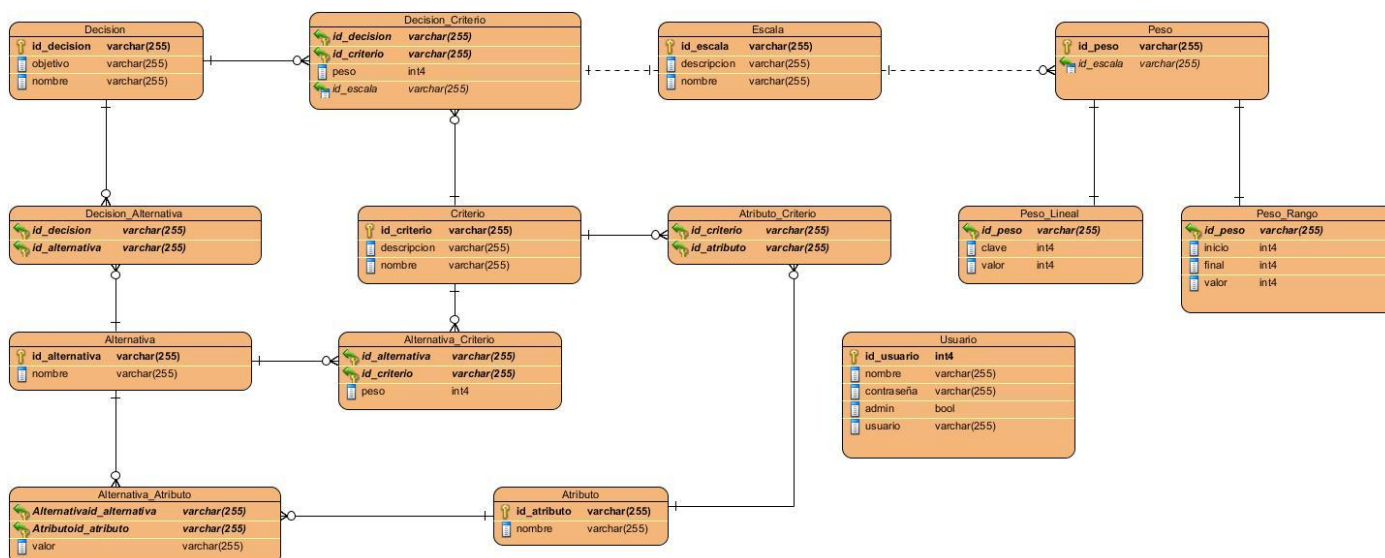


Figura 4: Modelo de Datos

Se describe el modelo de datos en el anexo 5, cada una de las tablas y sus atributos, un ejemplo de la descripción de estos modelos es la tabla candidato.

Nombre Tabla: decision		
<b>Descripción:</b> En esta tabla se almacenan los datos correspondientes a una decisión.		
Atributos	Tipo	Descripción
id_decision	String	Es el identificador de la decisión. Es la llave primaria de la tabla.
objetivo	String	Describe el objetivo de la decisión. Describe la decisión

Tabla 6: Tabla Decisión de la Base de Datos

# Capítulo 2. Características del Sistema

---

## Conclusiones Parciales

- Al poder realizarse una correcta captura de requisitos, el equipo de desarrollo tuvo una mayor claridad de las necesidades del cliente permitiendo esto la comprensión del sistema y posterior representación de las funcionalidades requeridas.
- Se documentaron en la Lista de Reserva del Producto las principales funcionalidades a implementar.
- Se realizó la planificación de las tareas a cumplir por el equipo como parte de la planificación del mismo y siguiendo con las normas de la metodología seleccionada.
- Mediante las historias de usuarios, se sentaron las bases para las restantes fases del proceso.
- Con la concepción y posterior diseño del Modelo de Datos se representaron las entidades relevantes del sistema, así como la estructura y diseño de la base de datos a utilizar y por medio del Diagrama de Clases se visualizaron las relaciones entre las clases que involucran dicho sistema, permitiendo una mejor comprensión del mismo.
- El capítulo anteriormente descrito será utilizado como entrada en la siguiente etapa de trabajo.

# Capítulo 3. Implementación y Prueba

## Capítulo 3 Implementación y Prueba

En este capítulo se describe cómo fue implementada la aplicación en términos de componentes, así como las pruebas que se realizaron a la aplicación y al código para la evaluación de los mismos. Se detalla mediante el diagrama de despliegue cómo quedará distribuida la aplicación y el hardware a utilizar. Se presenta la validación de las pruebas de calidad desarrolladas. Al final del mismo se realiza un análisis de la solución obtenida.

### 3.1 Implementación

#### 3.1.1 Diagrama de componentes

Un Diagrama de Componente es, como su nombre lo indica, un esquema o diagrama que muestra las interacciones y relaciones de los componentes de un modelo. Entendiéndose como componente a una clase de uso específico, que puede ser implementada desde un entorno de desarrollo, ya sea de código binario, fuente o ejecutable; dichos componentes poseen tipo, que indican si pueden ser útiles en tiempo de compilación, enlace y ejecución. Este tipo de diagrama se representa mediante componentes unidos mediante relaciones de dependencia (generalmente de compilación). (30)

Los Diagramas de Componentes ilustran las piezas del software, controladores embebidos que conformarán un sistema. Un diagrama de Componentes tiene un nivel más alto de abstracción que un diagrama de clase, usualmente un componente se implementa por una o más clases en tiempo de ejecución. Estos son bloques de construcción, como eventualmente un componente puede comprender una gran porción de un sistema.

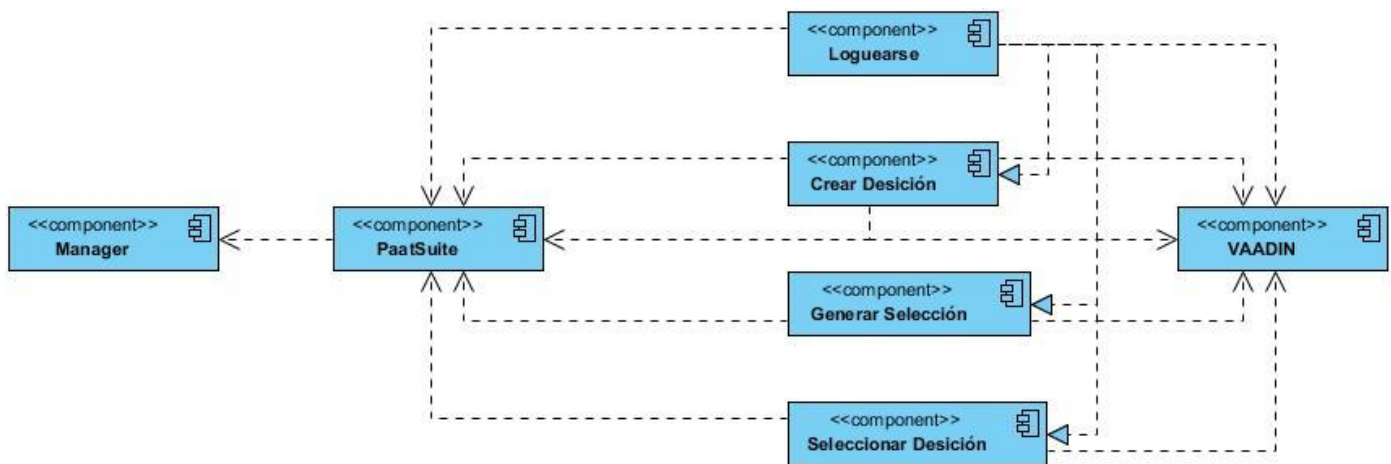


Figura 5: Diagrama de componentes

# Capítulo 3. Implementación y Prueba

## 3.1.2 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. Los nodos son conectados por asociaciones de comunicación tales como enlaces de red y conexiones TCP/IP. (30)

Es la etapa del desarrollo que describe la configuración del sistema para su ejecución en un ambiente del mundo real. Para el despliegue se deben tomar decisiones sobre los parámetros de la configuración, funcionamiento, asignación de recursos, distribución y concurrencia. Un diagrama de despliegue muestra la configuración de nodos que participan en la ejecución y de los componentes que residen en ellos.

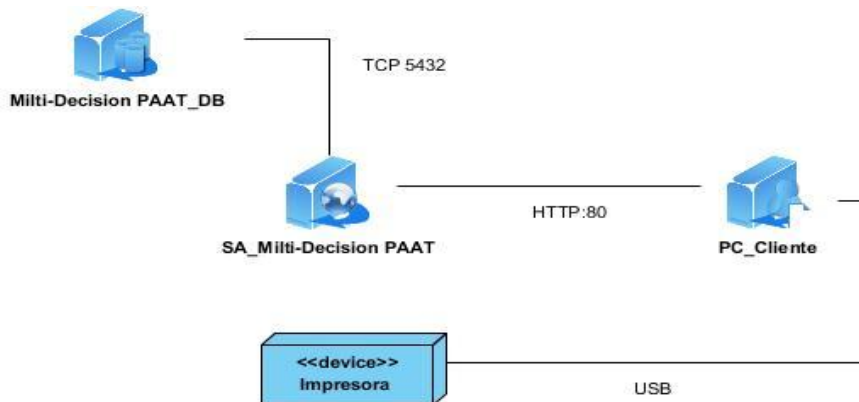


Figura 6: Diagrama de despliegue

En la propuesta del sistema se puede observar que la PC\_Cliente está conectada por USB a un dispositivo de impresión, donde se podrá imprimir y así persistir de forma física los resultados de las decisiones, y por el protocolo HTTP y el puerto 8080 al servidor de aplicaciones. Por último se puede apreciar mediante el protocolo TCP y puerto 5432 conectarse a la base de datos del sistema.

### Diseño de los casos de prueba

Las pruebas de caja negra también conocidas con sus varios nombres, como pruebas funcionales, pruebas de caja opaca, pruebas de entrada/salida, pruebas inducidas por los datos, son las que no toman en cuenta el código, como quien dice el que lo prueba no sabe cómo está estructurado por dentro el programa o bien no necesita saber nada de programación, solo necesita saber cuáles pueden ser las posibles entradas sin necesidad de entender cómo se deben obtener las salidas, donde se trata de encontrar errores en la interfaz mientras se está usando, el cómo luce, se maneja. (31) Este grupo de pruebas se utilizó para la validación de las funcionalidades del sistema. Las pruebas de caja negra se centran principalmente en lo que “se quiere” de un módulo o sección específica de un software, es decir, es una manera de encontrar casos en ese modulo que atiendan a su especificación.

# Capítulo 3. Implementación y Prueba

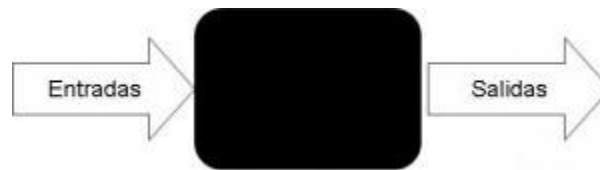


Figura 7: Esquema de Caja Negra

Según (32) las pruebas de caja negra permiten identificar problemas tales como:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las Bases de Datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación.

A continuación se muestra un caso de prueba de aceptación realizados a la aplicación.

Caso de Prueba de Aceptación	
Código Caso de Prueba: A-1-1	Nombre Historia de Usuario: Autenticación de usuarios
Nombre de la persona que realiza la prueba: Hermes Miguel	
Descripción de la Prueba: Se le proporciona al sistema el usuario y la contraseña, comprobando que el usuario se autentifique correctamente.	
Condiciones de Ejecución: <ul style="list-style-type: none"><li>• Que se encuentre activa la conexión con los servidores de Base de Datos.</li><li>• Que el usuario y la contraseña sean proporcionados correctamente.</li></ul>	
Entrada / Pasos de ejecución: Se le introducen el usuario y la contraseña al sistema, posteriormente se verifica que sean los correctos.	
Resultado Esperado: Que el usuario especialista acceda a las funcionalidades a las cuales tiene acceso.	
Evaluación de la Prueba: Satisfactorio	

Tabla 7: Diseño de Caso de Prueba de Aceptación

En el anexo 7 se encuentra el resto de las pruebas de aceptación realizadas.

# Capítulo 3. Implementación y Prueba

---

## Resultados de la aplicación de las métricas para la validación

Según (32) Las métricas de software son una medida cuantitativa del grado que pueden ser usados para determinar el valor que toman cierta característica en un producto de software concreto. Permiten definir la corrección de las clases del sistema y el sistema en sí. Estas se usan con el objetivo de ampliar la calidad del producto final. También es usado para la evaluación de diferentes indicadores tales como: El trabajo del equipo de desarrollo y la calidad de los procesos llevados a cabo.

Para la evaluación de la calidad del diseño propuesto se hizo un estudio de las métricas básicas inspiradas en la calidad del diseño orientado a objeto, en el mismo se abarcan atributos de calidad que permiten medir la calidad del diseño propuesto. Dentro de estos se encuentran: (32)

- ✓ **Responsabilidad:** consiste en la responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto, de la problemática propuesta.
- ✓ **Complejidad de implementación:** consiste en el grado de dificultad que tiene que implementar un diseño de clases determinado.
- ✓ **Reutilización:** consiste en el grado de reutilización presente en una clase o estructura de clase, dentro de un diseño de software.
- ✓ **Acoplamiento:** consiste en el grado de dependencia o interconexión de una clase o estructura de clase con otras; está muy ligada a la característica de Reutilización.
- ✓ **Complejidad del mantenimiento:** consiste en el grado de esfuerzo necesario a realizar para desarrollar un arreglo, una mejora o una rectificación de algún error de un diseño de software. Puede influir indirecta, pero fuertemente en los costos y la planificación del proyecto.
- ✓ **Cantidad de pruebas:** consiste en el número o el grado de esfuerzo para realizar las pruebas de calidad del producto diseñado.

Las métricas concebidas como instrumento para evaluar la calidad del diseño y su relación con los atributos de calidad definidos son las siguientes:

1. **Tamaño operacional de clase (TOC):** se refiere al número de métodos pertenecientes a una clase. Está determinada por los atributos: Responsabilidad, Complejidad de implementación y la Reutilización, existiendo una relación directa con los dos primeros e inversa con el último antes mencionado.

## Capítulo 3. Implementación y Prueba

Atributo que afecta	Modo en que lo afecta
Responsabilidad	Un aumento del TOC implica un aumento de la responsabilidad asignada a la clase.
Complejidad de implementación	Un aumento del TOC implica un aumento de la complejidad de implementación de la clase.
Reutilización	Un aumento del TOC implica una disminución en el grado de reutilización de la clase.

Tabla 8: Tabla de atributos que mide la métrica de tamaño de clase TOC.

Atributo	Categoría	Criterios
Responsabilidad	Baja	$\leq$ Prom.
	Media	Entre Prom. y $2^*$ Pom.
	Alta	$> 2^*$ Prom.
Complejidad implementación	Baja	$\leq$ Prom.
	Madia	Entre Prom. y $2^*$ Pom.
	Alta	$> 2^*$ Prom.
Reutilización	Baja	$> 2^*$ Prom.
	Madia	Entre Prom. y $2^*$ Pom.
	Alta	$\leq$ Prom.

Tabla 9: Rango de valores para la evaluación técnica de los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización) relacionados con la métrica TOC.

Seguidamente se muestra el resultado del uso de la métrica Tamaño operacional de Clases (TOC).

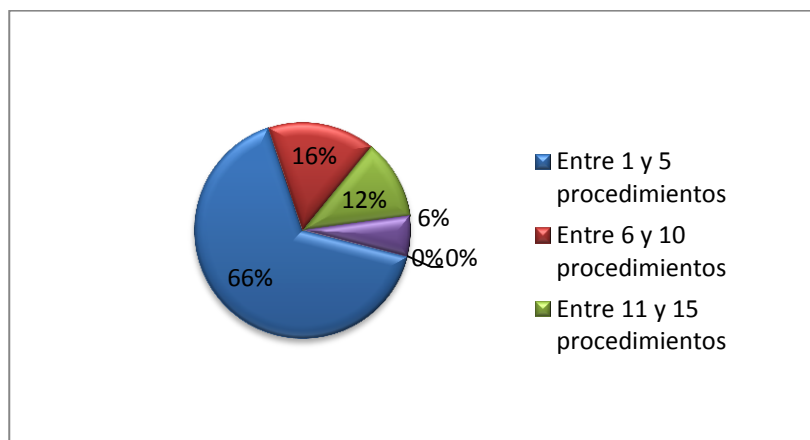


Figura 8: Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.

# Capítulo 3. Implementación y Prueba

Representación gráfica de la aplicación de la métrica TOC para el atributo Responsabilidad.

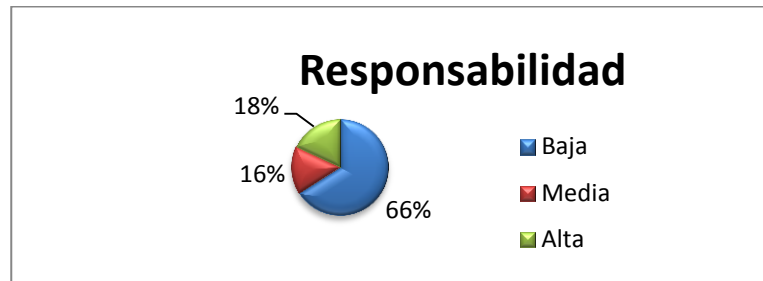


Figura 9: Representación en % de los resultados obtenidos en el atributo Responsabilidad.

Representación gráfica de la aplicación de la métrica TOC para el atributo Complejidad.

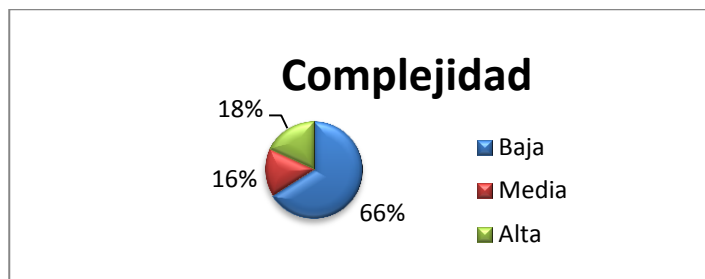


Figura 10: Representación en % de los resultados obtenidos en el atributo Complejidad.

Representación gráfica de la aplicación de la métrica TOC para el atributo Reutilización.

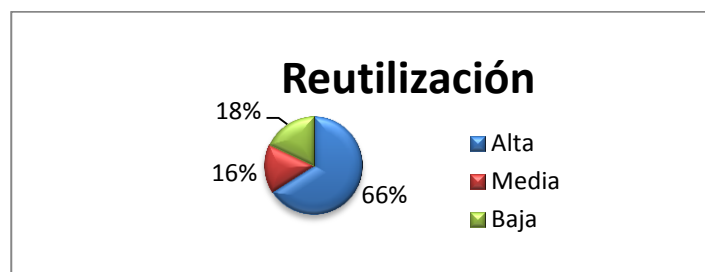


Figura 11: Representación en % de los resultados obtenidos en el atributo Reutilización.

Al analizar los resultados obtenidos luego de aplicar el instrumento de medición de la métrica TOC, se puede concluir que el diseño propuesto para el sistema es simple y tiene una calidad aceptable, teniendo en cuenta que la mayoría de las clases posee menos cantidad de operaciones que la media registrada en las mediciones. Los atributos de calidad se encuentran en un nivel satisfactorio, de manera que se puede observar cómo se fomenta la Reutilización (elemento clave en el proceso de desarrollo de software) y cómo están reducidas en menor grado la Responsabilidad y la Complejidad de implementación.



# Capítulo 3. Implementación y Prueba

## 2. Relaciones entre clases (RC):

Está dado por el número de relaciones de uso de una clase con otras.

Atributo que afecta	Modo en que lo afecta
<b>Acoplamiento</b>	Un aumento del RC implica un aumento del acoplamiento de la clase.
<b>Complejidad del mantenimiento</b>	Un aumento del RC implica un aumento de la complejidad de mantenimiento de la clase.
<b>Reutilización</b>	Un aumento del RC implica una disminución en el grado de reutilización de la clase.
<b>Cantidad de pruebas</b>	Un aumento del RC implica un aumento de la cantidad de pruebas de unidad necesarias para probar una clase.

Tabla 10: Tabla de atributos que mide la métrica relación entre clases (RC).

Atributo	Categoría	Criterios
Acoplamiento	Ninguno	0
	Bajo	1
	Media	2
	Alta	>2
Complejidad Mantenimiento	Baja	< =Prom.
	Madia	Entre Prom. y 2* Pom.
	Alta	> 2* Prom.
Reutilización	Baja	> 2*Prom.
	Madia	Entre Prom. y 2* Pom.
	Alta	<= Prom.
Cantidad de Pruebas	Baja	< =Prom.
	Madia	Entre Prom. y 2* Pom.
	Alta	> 2* Prom.

Tabla 11: Rango de valores para la evaluación técnica de los atributos de calidad (Acoplamiento, Complejidad Mantenimiento, Reutilización y Cantidad de Pruebas) relacionados con la métrica RC.

A continuación se muestra el resultado del uso de la métrica Relaciones entre Clases (RC).

# Capítulo 3. Implementación y Prueba

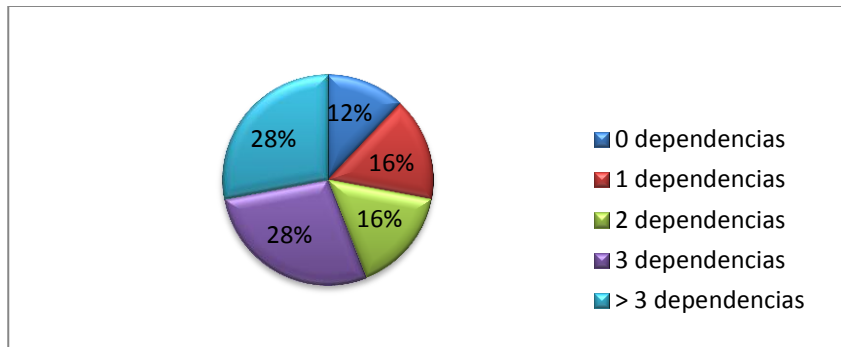


Tabla 12: Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.

Representación gráfica de la aplicación de la métrica RC para el atributo Acoplamiento. Ver Figura #

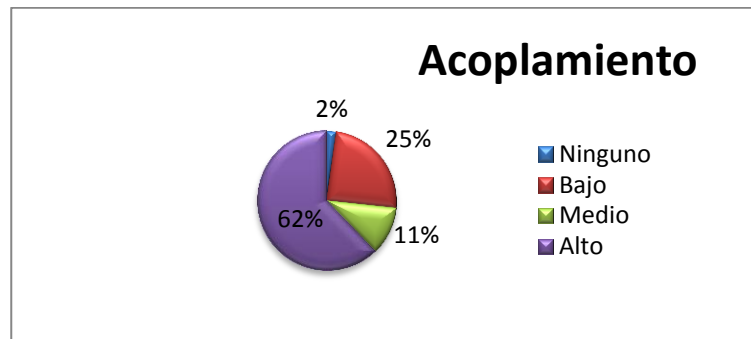


Tabla 13: Representación en % de los resultados obtenidos en el atributo Acoplamiento.

Representación gráfica de la aplicación de la métrica RC para el atributo Complejidad de Mantenimiento.

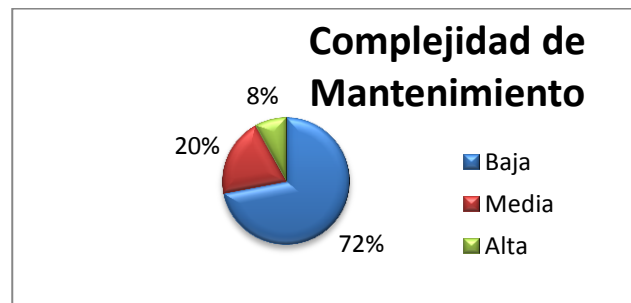


Figura 12: Representación en % de los resultados obtenidos en el atributo Complejidad de Mantenimiento.

Representación gráfica de la aplicación de la métrica RC para el atributo Cantidad de Pruebas. Ver Figura #

# Capítulo 3. Implementación y Prueba

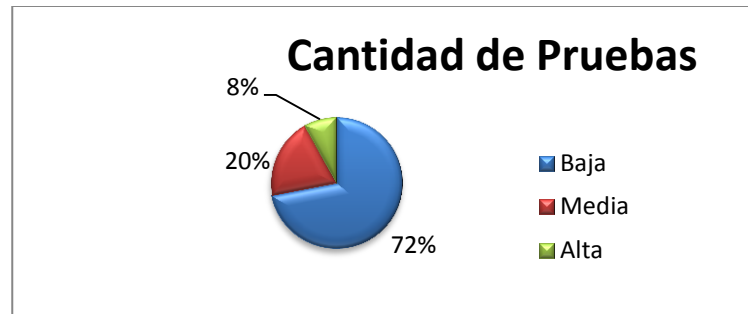


Figura 13: Representación en % de los resultados obtenidos en el atributo Cantidad de Pruebas.

Representación gráfica de la aplicación de la métrica RC para el atributo Reutilización. Ver Figura #



Figura 14: Representación en % de los resultados obtenidos en el atributo Reutilización.

Al analizar los resultados obtenidos luego de aplicar el instrumento de medición de la métrica RC, se puede concluir que el diseño propuesto para el sistema es simple y tiene una calidad aceptable, teniendo en cuenta que la mayoría de las clases poseen 2 o menos dependencias respecto a otras. Los atributos de calidad se encuentran en un nivel satisfactorio, el grado de acoplamiento es mínimo, la Complejidad de mantenimiento, la Cantidad de pruebas y la Reutilización se comportan favorablemente.

### Caso de Análisis:

La UCI, se propuso realizar una selección de un estudiante dentro de un grupo de 10 para que ocupe el cargo de programador en el proyecto Aduana de la Facultad 3. Como parte de dicha decisión los expertos determinaron una serie de criterios de selección con los cuales se evaluarán a los estudiantes, con el fin de garantizar un mejor desempeño. Los criterios seleccionados fueron:

- ✓ Evaluación Final en la Asignatura de Programación 5.
- ✓ Evaluación Final en la Asignatura de Ingeniería de Software.
- ✓ Trabajo en Equipo.
- ✓ Cumplimiento en los Horarios Establecidos.
- ✓ Seriedad en la Entrega de las Tareas Asignadas.

# Capítulo 3. Implementación y Prueba

Para la misma se definieron además un conjunto de pesos los cuales brindan la posibilidad de ponderar que criterios tienen mayor relevancia en la decisión. Los mismos quedaron de la siguiente forma:

- 1) 9
- 2) 4
- 3) 6
- 4) 4
- 5) 7

A continuación se listan las alternativas a tener en cuenta en la selección.

- ✓ Reinier Torres Estol
- ✓ Javier E. Padrón Del Pico
- ✓ Rayner Guerra Aizpurúa
- ✓ Rafael Rojas Sánchez
- ✓ Felipe Díaz Pérez
- ✓ Astroberto Cárdenas Órfica
- ✓ Dayanna Gómez
- ✓ Jenny Leal González
- ✓ Janet Tarama
- ✓ José Raúl Estol

Para validar la veracidad de los datos que se arrojan en la aplicación MultiDecisionPAAT se cree necesario que este caso de análisis sea ejecutado mediante otra herramienta para ser comparado, la cual se determinó que fuese el software Decision Lab por ser una poderosa herramienta que implementa el método propuesto. Luego de haber ejecutado el problema mediante estas herramientas de apoyo a la toma de decisión se obtuvieron los siguientes resultados:

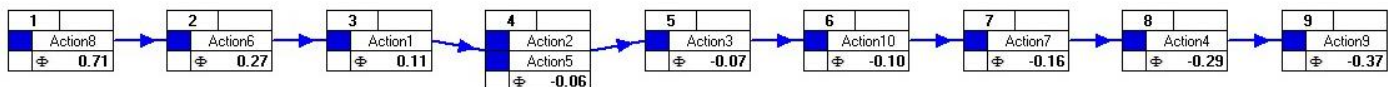


Figura 15: Vista ampliada del ranking final mediante el software Decision Lab

# Capítulo 3. Implementación y Prueba

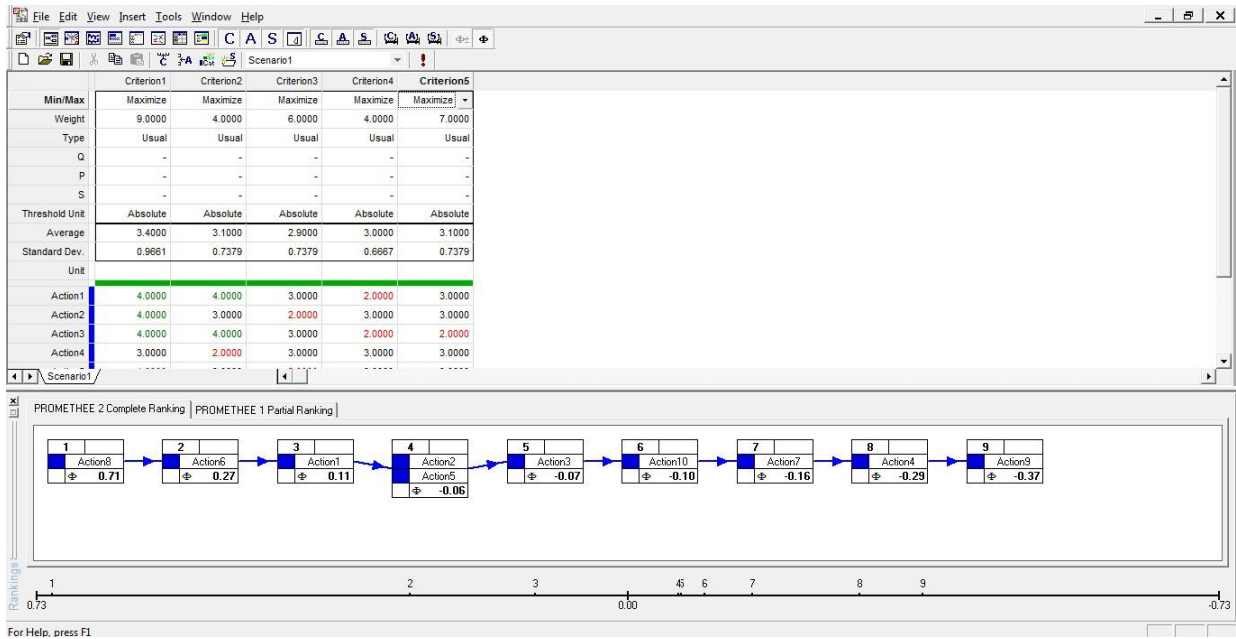


Figura 16: Vista completa del resultado final del problema de decisión multicriterios

POSICIÓN FINAL	NOMBRE DE LAS ALTERNATIVAS	FLUJO NETO
1	A8	2.1444444444444444
2	A6	0.8111111111111111
3	A1	0.34444444444444466
4	A5	-0.18888888888888877
5	A2	-0.18888888888888877
6	A3	-0.20000000000000018
7	A10	-0.2888888888888889
8	A7	-0.46666666666666656
9	A4	-0.8666666666666665
10	A9	-1.1

Figura 17: Vista ampliada del ranking final mediante la aplicación MultiDecisionPAAT



Figura 18: Vista completa del resultado mediante la aplicación MultiDecisionPAAT

## Capítulo 3. Implementación y Prueba

---

Luego de los resultados obtenidos se puede llegar a la conclusión que los rankings finales de posicionamiento de las alternativas son iguales, aunque que los flujos netos no coincidan sí existe una proporcionalidad entre ellos. Por lo que queda demostrado el hecho de que la mejor alternativa para el puesto de programador en el proyecto Aduana es la alternativa 8, que responde a la estudiante Jenny Leal González y de esta manera queda validado el módulo PROMETHEE de la herramienta de apoyo a la toma de decisión multicriterios MultiDecisionPAAT.

### **Conclusiones Parciales.**

En este capítulo se realizó la implementación y validación del sistema, lo que permitió obtener importantes resultados los cuales se mencionan a continuación:

1. Fueron expuestas las métricas para validar el diseño, las cuales arrojaron resultados satisfactorios sobre el diseño realizado, demostrando que este era simple y los atributos de calidad alcanzaban niveles favorables.
2. Se realizaron al sistema pruebas de caja negra. Estas arrojaron resultados satisfactorios, demostrándose con ello la calidad del sistema propuesto.
3. Con el caso de análisis se demostró el cumplimiento de las variables del problema a resolver.

## Conclusiones Generales.

Una vez terminado el presente trabajo se puede concluir que fueron cumplidos los objetivos planteados. Para los cuales se arrojan las siguientes conclusiones generales:

- ✓ Mediante la utilización de un sistema de ayuda a la toma de decisión se puede optimizar las tareas y ahorrar tiempo. Además de que puede ser usado en diversas ramas de la vida y en cualquier renglón de la sociedad.
- ✓ Los sistemas existentes que implementan el método PROMETHEE son privativos, en su mayoría personalizados hacia problemas específicos y no poseen la ventaja de la tecnología WEB.
- ✓ El sistema se desarrolló utilizando el IDE Eclipse y en la implementación de cada una de las capas los marcos de trabajo: Hibernate para el acceso a datos, Spring para el negocio y Vaadin para la capa de presentación. Como sistema gestor de base de datos Postgres SQL.
- ✓ Con la aplicación de la metodología se fueron generando los artefactos que la misma exigía, lo cual ayuda al entendimiento del sistema por parte del cliente.
- ✓ El sistema fue sometido a rigurosas pruebas de funcionalidad y de validaciones del diseño, las cuales arrojaron que la aplicación cumple con las funcionalidades requeridas por el sistema.

## **Recomendaciones**

1. El uso de la aplicación durante el proceso de ubicación de los estudiante en los roles de proyecto en la UCI.
2. Agregarle las funcionalidades que le permitan cargar la información a la aplicación de forma automática, mediante servicios telemáticos.
3. El uso de la misma con carácter educativo y forme parte de la cantera de productos de la UCI a comercializar.



# Referencias Bibliográficas

---

## Referencias Bibliográficas

1. [http://www.urbanismogranada.com/administrador/archivos/12\\_12\\_12\\_gvSig\\_TALLER\\_5.pdf](http://www.urbanismogranada.com/administrador/archivos/12_12_12_gvSig_TALLER_5.pdf). [En línea] [Citado el: 06 de 03 de 2013.]  
[http://www.urbanismogranada.com/administrador/archivos/12\\_12\\_12\\_gvSig\\_TALLER\\_5.pdf](http://www.urbanismogranada.com/administrador/archivos/12_12_12_gvSig_TALLER_5.pdf).
2. **R., Iannier.** Buenas Tareas. *Métodos Teóricos*. [En línea]  
<http://www.buenastareas.com/ensayos/Metodos-Teoricos/136411.html>.
3. **Hurtado et al, T., Bruno, Gérard.** *El proceso de analisis jerárquico (AHP) como herramienta para la toma de decisiones en la selección de proveedores*. Lima : s.n., 2005.
4. **Rodríguez, Rolando.** scribd. *Métodos empíricos*. [En línea]  
<http://es.scribd.com/doc/21229743/METODOS-EMPIRICOS>.
5. **Ferrer, Antonio.** OBSERVACIÓN DESCRIPTIVA Y EXPERIMENTO. [En línea]  
<HTTP://WWW.UIAH.FI/PROJECTS/METODI/262.HTM>.
6. **SlideShare.** Tipos de Sistemas de Información. [En línea] 2009. Available from:  
<http://www.slideshare.net/about>.
7. **García, J.A.** [En línea] 2007. <http://www.cop.es/colegiados/m-00451/tomadeciones.htm>.
8. **Colero, Irina Canderó y cristina.** [En línea] 2010. <http://www.slideshare.net/irinacen/sistemas-de-sopORTE-a-la-toma-de-decisiones-dss>.
9. **Haettenschwiler, P.** *Neues anwenderfreundliches Konzept der Entscheidungsunterstützung*. Wirtschaft : s.n., 1999.
10. **Power, D.J.** *The On-Line Executive Journal for Data-Intensive Decision Support 1*. 1997.
11. **Delfos, C.E.** *SOFTWARE PARA LA IMPLEMENTACIÓN DE LOS SISTEMAS DE SOPORTE A LAS DECISIONES (SSD) A NIVEL INTERNACIONAL*. 2008.
12. **Barberis, Gabriela M. Fernández.** *LOS MÉTODOS PROMETHEE: Una Metodología de Ayuda a la Toma de Decisiones Multicriterio Discretas*. 2010.
13. Metodología Ágil de Desarrollo SXP. [En línea]  
[http://www.ecured.cu/index.php/Metodologia\\_Agil\\_de\\_Desarrollo\\_SXP](http://www.ecured.cu/index.php/Metodologia_Agil_de_Desarrollo_SXP).
14. Sxp Metodología Ágil Para Proyectos De Software Libre. [En línea] 2011.  
<http://www.buenastareas.com/ensayos/Sxp-Metodolog%C3%ADa-%C3%81gil-Para-Proyectos-De/1405930.html>.
15. **IBM.** Unified Modeling Language (UML). [En línea] <http://www-01.ibm.com/software/rational/uml/>.
16. Introducción a UML. [En línea] 2011. <http://docs.kde.org/stable/es/kdesdk/umbrello/uml-basics.html#about-uml>.
17. **Castells, Pablo.** *Programación Orientada a Objetos*. Madrid : s.n., 2010.
18. **Pesquera, Carlos.** Arquitectura, análisis y desarrollo de software empresarial. [En línea] 2012.  
<http://carlospesquera.com/vaadin-aplicaciones-web-java-como-si-fuesen-de-escritorio/>.
19. **JOHNSON, R.** *Professional Java Development with the Spring Framework*. 2005.
20. **Marquez, Iskael Díaz.** *“Diseño e implementación de las capas de negocio y acceso a datos de los módulos Planificación y Ejecución de Visitas Familiares”*. 2008.

# Referencias Bibliográficas

---

21. *Relational Persistence for Java*. 2010.
22. **Fernando, D.M.** Patrones de Diseño de Arquitecturas de Software Enterprise. [En línea] 2005. <http://materias.fi.uba.ar/7500/montaldo-esisdegradoingenieriainformatica.pdf>.
23. **website, The Eclipse Foundation open source community**. [En línea] <http://www.eclipse.org/>.
24. Elección del servidor de aplicaciones web. [En línea] <http://www.help400.es/asp/scripts/nwart.asp?Num=131&Pag=10&Tip=T>.
25. Servidor Web Tomcat. [En línea] [http://www.ecured.cu/index.php/Servidor\\_Tomcat](http://www.ecured.cu/index.php/Servidor_Tomcat).
26. **Veracruz, Instituto tecnológico de**. [En línea] <http://www.prograweb.com.mx/pweb/0201arquiAplicaweb.html>.
27. **Gutierrez, Jorge A. Saavedra**. El mundo informático. [En línea] <http://jorgesaavedra.wordpress.com/2007/05/08/patrones-grasp-patrones-de-software-para-la-asignacion-general-de-responsabilidadparte-ii/>.
28. **Fernández, Pablo Ramos**. *El Patrón Fachada*. 2009.
29. **R.J, Fernandez**. *Modelo de Datos*.
30. **D.S, Wilder**. Diagrama de Componentes. [En línea] 2009. [http://diagramacomponente.blogspot.com/2009/05/definicion\\_18.html](http://diagramacomponente.blogspot.com/2009/05/definicion_18.html).
31. **Ingeniería, Universidad Autónoma de Chihuahua.Facultad de**. Buenas Tareas. [En línea] <http://www.buenastareas.com/ensayos/Pruebas-De-Caja-Negra/1477804.html>.
32. **Pressman, Roger S**. Ingeniería del Software. *Ingeniería del Software*.
33. **Simon, Hebert**. *The New Science of Management Decision*. New York : Harper and Row, 1960.
34. INTELIGENCIA ARTIFICIAL Y ROBOTICA. [En línea] 2009. [http://www.inteligenciaartificial.cl/ciencia/software/ia/inteligencia\\_artificial.htm](http://www.inteligenciaartificial.cl/ciencia/software/ia/inteligencia_artificial.htm).
35. The AMPAC people. [En línea] 2010. <http://www.semichem.com/codessa/default.php>.
36. **Sebastian, Juan**. comusoft.com. [En línea] <http://www.comusoft.com/modelo-vista-controlador-definicion-y-caracteristicas>.

## Anexos

### Anexo 1. Plantilla de Concepción Inicial del Sistema

#### 1. Polo productivo.

Macro proyecto de investigación: Sistema para la ayuda a la toma de decisión multicriterios.

#### 2. Clasificación del proyecto.

Desarrollo de aplicación

#### 3. Tipo de proyecto.

Nacional

#### 4. Resumen:

En el documento se refleja la visión general del producto a implementar, además de realizar la descripción de los roles que intervienen en el desarrollo del software, así como las responsabilidades que tendrán en dicho proceso. Se documenta el tipo de proyecto al que pertenece, así como la especificación del Polo Productivo y su clasificación. Contiene además, cuales herramientas serán utilizadas para el desarrollo de la aplicación, el alcance que va a tener, una descripción de los involucrados en el negocio, cuales son los motivos de la necesidad del desarrollo del software y la propuesta de solución.

#### 5. Surgimiento.

La necesidad de realizar el sistema surge luego de haber realizado un análisis en las áreas donde la toma de decisión juega un papel fundamental. Son aquellas donde el decisor tiene que analizar una gran cantidad de información y procesarla de forma manual, ya que por lo general el resultado no aporta criterios eficientes y confiables, e incluso en ocasiones no es factible su realización.

#### 6. ¿Qué es?

**SXP** es una metodología consistente en una programación rápida o extrema, cuya particularidad es tener como parte del equipo al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto y está basada completamente en los valores y principios de las metodologías ágiles expuestos en el Manifiesto Ágil.

SXP se crea por la unión de las metodologías XP y Scrum. En ella se utiliza el Scrum para la gestión, logrando una correcta planificación y organización; mientras que XP respalda con sus prácticas todo el ciclo de desarrollo y de esta forma se obtiene un proceso de software completo.

#### **Metodología a utilizar.**

**SXP** es la metodología que consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar el éxito del proyecto. Basada completamente en los valores y principios de las metodologías ágiles expuestos en el Manifiesto Ágil.

La creación de SXP con la unión de las metodologías XP y Scrum se centra principalmente en que con la utilización de SCRUM para la gestión, se logra una correcta planificación y organización; mientras que XP respalda con sus prácticas todo el proceso de desarrollo y de esta forma se obtiene un proceso de software completo.

## 7. Involucrados.

Lizandra Arza Pérez y Hermes Miguel Velázquez expertos en la metodología y el modelo.

## 9. Misión.

Brindar los criterios de decisión en ambientes donde la confección manual del mismo se torna impracticable y es susceptible a errores.

## 10. Alcance

El desarrollo de las funcionalidades necesarias para hacer el análisis de la información y dar los criterios para la toma de decisión.

## 11. Herramientas utilizadas.

### Herramienta Case

**Visual Paradigm** que es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. También proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. Presenta licencia gratuita y comercial. Es fácil de instalar y actualizar y compatible entre ediciones. Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos. Importación y exportación de ficheros XMI.

### Lenguaje utilizado

**Java** el cual es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90 (Gómez, 2010). Se puede considerar el lenguaje Java como de Software Libre. En la actualidad java se ha convertido en uno de los lenguajes más usados y más demandados por los desarrolladores. Es independiente de la plataforma de desarrollo. Existen dentro de su librería clases gráficas como awt y swing, las cuales permiten crear objetos gráficos comunes altamente configurables y con una arquitectura independiente de la plataforma. Java permite a los desarrolladores aprovechar la flexibilidad de la Programación Orientada a Objetos en el diseño de sus aplicaciones. Se puede acceder a bases de datos fácilmente con JDBC, independientemente de la plataforma utilizada. El manejo de las bases de datos es uniforme, es decir transparente y simple.

### Entornos de Desarrollo Integrado (IDEs)

**Eclipse** es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Eclipse fue desarrollado originalmente por IBM (Corporación Internacional de Negocios Informáticos) por sus siglas en inglés (*International Business Machines Corporation*) como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse.

La interfaz de usuario de Eclipse tiene una capa GUI (Interfaz Gráfica de Usuarios) por sus siglas en inglés (*Graphical User Interface*) intermedia llamada JFace, la cual simplifica la construcción de aplicaciones basadas en SWT (Standard Widget Toolkit). El entorno de desarrollo integrado de Eclipse emplea módulos (en inglés *plug-in*) para proporcionar toda su funcionalidad al frente de la plataforma de cliente enriquecido, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Este mecanismo de módulos es una plataforma ligera para componentes de software. En cuanto a las aplicaciones clientes, Eclipse provee al programador con marcos de trabajo muy ricos para el desarrollo de aplicaciones gráficas, definición y manipulación de modelos de software, aplicaciones web, etc.

## 12. Solución propuesta.

Partiendo de que la información que se tiene del estudiante se encuentra digital en un grupo de sistemas desarrollados en la universidad y teniendo en cuenta los elementos antes expuestos, se hace necesario el desarrollo de una aplicación informática que implemente los algoritmos necesarios para la implementación y mejore los tiempos de ejecución del modelo.

### Anexo 2. Modelo de historias de usuarios del negocio.

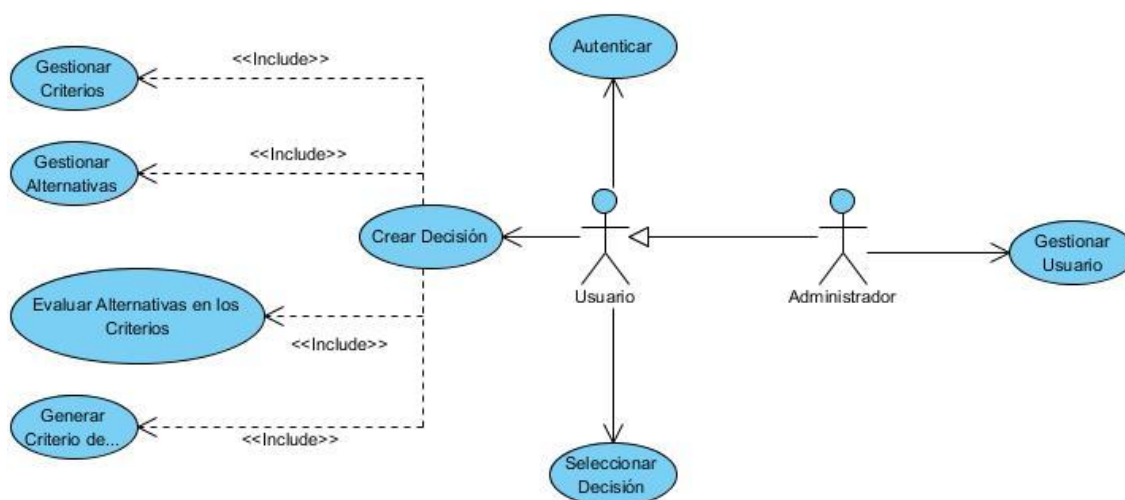
#### 1. Actores del negocio

Actor	Descripción
Usuario(Especialistas)	Este personal se beneficia con el negocio al ser ellos los interesados de que la selección se realice lo más acertada posible, permitiendo así que se realice de una forma ágil el trabajo de selección.

#### 2. Trabajadores del negocio

Trabajador	Descripción
Administrador del sistema	Será el encargado de realizar la gestión de los usuarios.

### 3. Diagrama de Historias de usuario del Negocio.



### Anexo 3. Plantilla de historias de usuarios

Historia de Usuario	
Número: HU_1	Nombre Historia de Usuario: Autenticar usuarios
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Reinier Torres	Iteración Asignada: Sprint 1
Prioridad en Negocio: <i>Muy Alta</i>	Puntos Estimados: 1
Riesgo en Desarrollo: Alto	Puntos Reales: 1
Descripción: <i>El usuario accede al sistema insertando un usuario y su contraseña, posteriormente se procede a comprobar que estos datos son correctos, al serlo la aplicación le dará al usuario acceso a las funcionalidades a las cuales tiene permisos.</i>	
Observaciones:	

Prototipo de interfaces:



Regístrate




Usuario:

Contraseña:



Tabla 14: Historia de Usuario 1

Historia de Usuario	
Número: HU_2	Nombre Historia de Usuario: Crear decisión
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Reinier Torres	Iteración Asignada: Sprint 3
Prioridad en Negocio: Muy Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Alto	Puntos Reales: 3
Descripción: <i>El usuario especialista crea una decisión, introduciendo el nombre de la decisión y una descripción .Este es el primer paso de una serie de pasos lógicos.</i>	
Observaciones:	
Prototipo de interfaces:	
	

**Tabla 15: Historia de Usuario 2**

Historia de Usuario							
Número: HU_3	Nombre Historia de Usuario: Gestionar Criterio						
Modificación de Historia de Usuario Número: Ninguna							
Usuario: <i>Javier Padrón</i>	Iteración Asignada: Sprint 3						
Prioridad en Negocio: Muy Alta	Puntos Estimados: 3						
Riesgo en Desarrollo: Alto	Puntos Reales: 3						
<p>Descripción <i>El usuario especialista podrá adicionar o eliminar un criterio, así como asignarle un peso un nombre y una descripción. Es el segundo paso de la secuencia lógica.</i></p>							
Observaciones:							
Prototipo de interfaces:							
<p><b>Gestionar Criterios</b></p> <p>Nombre: Matemática 1   Descripción: Criterio de Matemática 1   Peso: 8   Adicionar</p> <table border="1"> <thead> <tr> <th>NOMBRE</th> <th>DESCRIPCIÓN</th> <th>PESO</th> </tr> </thead> <tbody> <tr> <td>Base de Datos 1</td> <td>Criterio de Base de Datos 1</td> <td>5</td> </tr> </tbody> </table> <p>Eliminar   Atras   Siguiete   Cancelar</p>		NOMBRE	DESCRIPCIÓN	PESO	Base de Datos 1	Criterio de Base de Datos 1	5
NOMBRE	DESCRIPCIÓN	PESO					
Base de Datos 1	Criterio de Base de Datos 1	5					

**Tabla 16: Historia de Usuario 3**



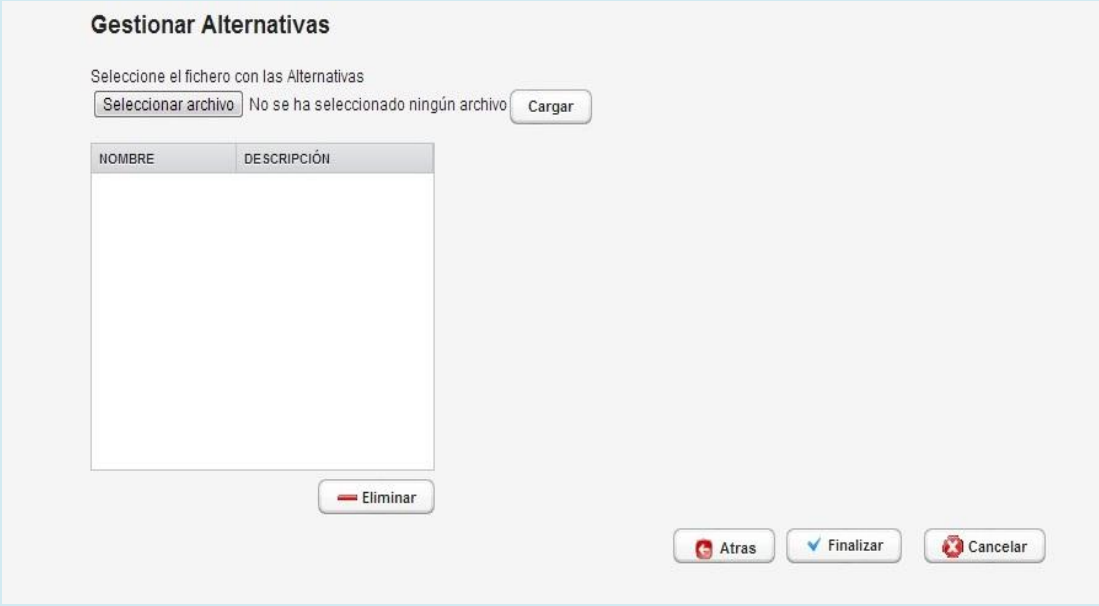
Historia de Usuario	
Número: HU_4	Nombre Historia de Usuario: Gestionar Alternativa
Modificación de Historia de Usuario Número: Ninguna	
Usuario: <i>Javier Padrón</i>	Iteración Asignada: Sprint 3
Prioridad en Negocio: Muy Alta	Puntos Estimados: 3
Riesgo en Desarrollo: Alto	Puntos Reales: 3
Descripción: <i>El usuario especialista podrá adicionar o eliminar alternativa desde un Excel. Es el último paso de esta secuencia lógica.</i>	
Observaciones:	
Prototipo de interfaces:	
	

Tabla 17: Historia de Usuario 4

## Anexo 4. Plantilla Tarea de Ingeniería

Tarea de Ingeniería	
Número Tarea: T_1	Número Historia de Usuario: todas
Nombre Tarea: Generación de la base de datos	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 15-enero -2013	Fecha Fin: 20- enero- 2013
Programador Responsable: Reinier Torres	
<p>Descripción: <i>Se genera la base de datos a partir del diseño realizado, esta debe implementarse sobre el sistema gestor definido para el desarrollo de la aplicación.</i></p>	

**Tabla 18: Tarea de Ingeniería 1**

Tarea de Ingeniería	
Número Tarea: T_2	Número Historia de Usuario: todas
Nombre Tarea: Montaje del ambiente de desarrollo con la integración de los marcos de trabajo seleccionados para el desarrollo	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 15-enero-2013	Fecha Fin: 5-febrero-2013
Programador Responsable: Javier Padrón	
<p>Descripción: <i>Instalar y configurar los frameworks definidos para el desarrollo de manera que se pueda comenzar la implementación de la aplicación.</i></p>	

**Tabla 19: Tarea de Ingeniería 2**

Tarea de Ingeniería	
Número Tarea: T_3	Número Historia de Usuario: todas
Nombre Tarea: Realizar el mapeo de la base de datos en el framework Hibernate.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 5-Febrero-2013	Fecha Fin: 20-Febrero-2013
Programador Responsable: Reinier Torres	
Descripción: <i>Realizar el mapeo de la base de datos en el framework de acceso a datos seleccionado.</i>	

**Tabla 20: Tarea de Ingeniería 3**

Tarea de Ingeniería	
Número Tarea: T_4	Número Historia de Usuario: HU_1
Nombre Tarea: Diseñar las interfaces requeridas para la funcionalidad: autenticación de usuario.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 21-Febrero-2013	Fecha Fin: 28-Marzo-2013
Programador Responsable: Javier Padrón	
Descripción: <i>Crear las interfaces necesarias, teniendo en cuenta los prototipos definidos en la Historia de Usuario.</i>	

**Tabla 21: Tarea de Ingeniería 4**

Tarea de Ingeniería	
Número Tarea: T_5	Número Historia de Usuario: HU_1
Nombre Tarea: Implementar funcionalidad de autenticación de usuarios	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 1-Marzo-2013	Fecha Fin: 5-Marzo-2013
Programador Responsable: Reinier Torres	
Descripción: <i>Crear las clases y algoritmos necesarios para la implementación de la funcionalidad de autenticación de usuario.</i>	

**Tabla 22: Tarea de Ingeniería 5**

Tarea de Ingeniería	
Número Tarea: T_6	Número Historia de Usuario: HU_5
Nombre Tarea: Estudio relacionado con el manejo de ficheros Excel desde Java.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 6-Marzo-2013	Fecha Fin: 11-Marzo-2013
Programador Responsable: Reinier Torres	
Descripción: <i>Se estudian las formas de manejar información contenida en ficheros Excel. Diseñar el algoritmo y definir las sentencias a utilizar para implementar las funcionalidades que requieren de este conocimiento.</i>	

**Tabla 23: Tarea de Ingeniería 6**

Tarea de Ingeniería	
Número Tarea: T_7	Número Historia de Usuario: HU_2, HU_3, HU_4, HU_5
Nombre Tarea: Diseñar las interfaces necesarias	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 12-Marzo-2013	Fecha Fin: 25-Marzo-2013
Programador Responsable: Javier Padrón	
Descripción: <i>Se diseñan las interfaces necesarias para la implementación de las funcionalidades, teniendo en cuenta los prototipos diseñados en las HU.</i>	

**Tabla 24: Tarea de Ingeniería 7**

Tarea de Ingeniería	
Número Tarea: T_8	Número Historia de Usuario: HU_6, HU_7, HU_8
Nombre Tarea: Diseñar las interfaces necesarias	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 12-Marzo-2013	Fecha Fin: 25-Marzo-2013
Programador Responsable: Reinier Torres	
Descripción: <i>Se diseñan las interfaces necesarias para la implementación de las funcionalidades, teniendo en cuenta los prototipos diseñados en las HU.</i>	

**Tabla 25: Tarea de Ingeniería 8**

Tarea de Ingeniería	
Número Tarea: T_9	Número Historia de Usuario: HU_2
Nombre Tarea: Implementar: Crear Decisión	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 26-Marzo-2013	Fecha Fin: 2-Abril-2013
Programador Responsable: Javier Padrón	
Descripción: <i>Se implementa la funcionalidad vinculada con la historia de usuario.</i>	

**Tabla 26: Tarea de Ingeniería 9**

Tarea de Ingeniería	
Número Tarea: T_10	Número Historia de Usuario: HU_3
Nombre Tarea: Implementar: Gestionar Criterios	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 3-Abril-2013	Fecha Fin: 7-Abril-2013
Programador Responsable: Reinier Torres	
Descripción: <i>Se implementa la funcionalidad vinculada con la historia de usuario.</i>	

**Tabla 27: Tarea de Ingeniería 10**

Tarea de Ingeniería	
Número Tarea: T_11	Número Historia de Usuario: HU_4
Nombre Tarea: Implementar: Gestionar Alternativas	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 8-Abril-2013	Fecha Fin: 13-Abril-2013
Programador Responsable: Javier Padrón	
Descripción: <i>Se implementa la funcionalidad relacionada con la historia de usuario.</i>	

**Tabla 28: Tarea de Ingeniería 11**

Tarea de Ingeniería	
Número Tarea: T_12	Número Historia de Usuario: HU_5
Nombre Tarea: Implementar: Evaluar Alternativas en los Criterios	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 14-Abril-2013	Fecha Fin: 19-Abril-2013
Programador Responsable: Reinier Torres	
Descripción: <i>Se implementa la funcionalidad relacionada con la historia de usuario.</i>	

**Tabla 29: Tarea de Ingeniería 12**

Tarea de Ingeniería	
Número Tarea: T_13	Número Historia de Usuario: HU_6
Nombre Tarea: Implementar: Generar Criterio	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 20-Abril-2013	Fecha Fin: 25-Abril-2013
Programador Responsable: Javier Padrón	
Descripción: <i>Se implementa la funcionalidad vinculada con la historia de usuario.</i>	

**Tabla 30: Tarea de Ingeniería 13**

Tarea de Ingeniería	
Número Tarea: T_14	Número Historia de Usuario: HU_7
Nombre Tarea: Implementar: Gestionar usuarios	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 26-Abril-2013	Fecha Fin: 1-Mayo-2013
Programador Responsable: Javier Padrón	
Descripción: <i>Se implementa la funcionalidad vinculada con la historia de usuario.</i>	

**Tabla 31: Tarea de Ingeniería 14**

Tarea de Ingeniería	
Número Tarea: T_15	Número Historia de Usuario: HU_8
Nombre Tarea: Implementar: Seleccionar Decisión	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 2-Mayo-2013	Fecha Fin: 7-Mayo-2013
Programador Responsable: Reinier Torres, Javier Padrón	
Descripción: <i>Se implementan las funcionalidades vinculada con la historia de usuario.</i>	

**Tabla 32: Tarea de Ingeniería 15**

## Anexo 5. Descripción del Modelo de Datos

<b>Nombre Tabla:</b> usuario		
<b>Descripción:</b> En esta tabla se almacenan los datos correspondientes a los usuarios.		
<b>Atributos</b>	<b>Tipo</b>	<b>Descripción</b>
id_usuario	Int	Es el identificador de los usuarios. Es la llave primaria de la tabla.
nombre	String	Es el nombre del usuario.
contraseña	String	Es la contraseña del usuario.
admin	boolean	Atributo que define el nivel de privilegio del usuario.
usuario	String	Campo que identifica a cada usuario.

**Nombre Tabla:** decision**Descripción:** En esta tabla se almacenan los datos correspondientes a una decisión.

Atributos	Tipo	Descripción
id_decision	String	Es el identificador de la decisión. Es la llave primaria de la tabla.
objetivo	String	Describe el objetivo de la decisión. Describe la decisión

**Nombre Tabla:** alternativa**Descripción:** En esta tabla se almacenan los datos correspondientes a una alternativa

Atributos	Tipo	Descripción
id_alternativa	String	Es el identificador de la alternativa. Es la llave primaria de la tabla.

**Nombre Tabla:** criterio**Descripción:** En esta tabla se almacenan los datos correspondientes a un criterio.

Atributos	Tipo	Descripción
id_criterio	Int	Es el identificador del criterio. Es la llave primaria de la tabla.

**Nombre Tabla:** peso**Descripción:** En esta tabla se almacenan los datos correspondientes a un peso.

Atributos	Tipo	Descripción
id_peso	Int	Es el identificador del peso. Es la llave primaria de la tabla.
valor	Int	Es el valor que toma el peso.
nomenclador	String	Es la variante cualitativa que puede tomar el peso.

**Nombre Tabla:** decision\_alternativa**Descripción:** En esta tabla se almacenan los datos correspondientes a un peso.

Atributos	Tipo	Descripción
id_decision	Int	Es la llave foránea de la relación existente entre la tabla decisión y la tabla alternativa.
id_alternativa	Int	Es la llave foránea de la relación existente entre la tabla decisión y la tabla alternativa.

**Nombre Tabla:** decision\_criterio**Descripción:** En esta tabla se almacenan los datos correspondientes a la relación la tabla decisión y la tabla criterio.

Atributos	Tipo	Descripción
id_decision	Int	Es la llave foránea de la relación existente entre la tabla decisión y la tabla criterio.
id_criterio	Int	Es la llave foránea de la relación existente entre la tabla decisión y la tabla criterio.
peso	Int	Valor del atributo de la alternativa.



Nombre Tabla: alternativa_criterio		
<b>Descripción:</b> En esta tabla se almacenan los datos correspondientes a la relación entre la tabla decisión y la tabla criterio.		
Atributos	Tipo	Descripción
id_alternativa	Int	Es la llave foránea de la relación existente entre la tabla alternativa y la tabla criterio.
id_criterio	Int	Es la llave foránea de la relación existente entre la tabla alternativa y la tabla criterio.
Id_peso	Int	Es la llave foránea de la relación existente entre la tabla alternativa_criterio y la tabla peso.

**Tabla 33: Tablas de la Base de Datos**

## Anexo 6. Plantilla Caso de Prueba de Aceptación

Caso de Prueba de Aceptación	
Código Caso de Prueba: A-1-1	Nombre Historia de Usuario: Autenticación de usuarios
Nombre de la persona que realiza la prueba: Hermes Miguel	
Descripción de la Prueba: Se le proporciona al sistema el usuario y la contraseña, comprobando que el usuario se autentifique correctamente.	
Condiciones de Ejecución: <ul style="list-style-type: none"> <li>• Que se encuentre activa la conexión con los servidores de Base de Datos.</li> <li>• Que el usuario y la contraseña sean proporcionado correctamente.</li> </ul>	
Entrada / Pasos de ejecución: Se le introducen el usuario y la contraseña al sistema posteriormente se verifica que sean los correctos.	
Resultado Esperado: Que el usuario especialista acceda a las funcionalidades a las cuales tiene acceso.	
Evaluación de la Prueba: Satisfactoria	

**Tabla 34: Caso de Prueba de Aceptación 1-1**

Caso de Prueba de Aceptación	
Código Caso de Prueba: A-2-1	Nombre Historia de Usuario: Crear Decisión
Nombre de la persona que realiza la prueba: Javier Padrón Del Pico	
Descripción de la Prueba El especialista Selecciona en al menú del sistema la opción que se describe.	
Condiciones de Ejecución:	
Entrada / Pasos de ejecución: Cuando el sistema cargue la pantalla inicial para comenzar la carga de los datos.	
Resultado Esperado: Que el sistema muestre la pantalla correspondiente	
Evaluación de la Prueba: Satisfactoria	

**Tabla 35: Caso de Prueba de Aceptación 2-1**

Caso de Prueba de Aceptación	
Código Caso de Prueba: A-3-1	Nombre Historia de Usuario: Gestionar Criterios
Nombre de la persona que realiza la prueba: Javier Padrón Del Pico	
Descripción de la Prueba Se ejecuta la prueba y se verifica que el usuario experto introduzca los criterios correctamente así como asignarle el peso correspondiente a la decisión.	
Condiciones de Ejecución:	
Entrada / Pasos de ejecución: Cuando el sistema cargue la pantalla Gestionar Criterios el especialista le proporcionara al sistema los criterios y el peso asociado a la decisión.	
Resultado Esperado: Que el sistema persista correctamente los datos en la base de datos	
Evaluación de la Prueba: Satisfactoria	

**Tabla 36: Caso de Prueba de Aceptación 3-1**

Caso de Prueba de Aceptación	
Código Caso de Prueba: A-4-1	Nombre Historia de Usuario: Gestionar Alternativas
Nombre de la persona que realiza la prueba: Reinier Torres	
Descripción de la Prueba: Se ejecuta la prueba y se verifica que el especialista cargue correctamente la información referente a las alternativas.	
Condiciones de Ejecución: <ul style="list-style-type: none"><li>• Se comprueba que el archivo a cargar se halla llenado previamente.</li></ul>	
Entrada / Pasos de ejecución: Cuando el sistema cargue la pantalla Gestionar Alternativas el sistema genera un fichero Excel especificándole al especialista la ubicación del mismo, el cual deberá llenar con las alternativas de la decisión. Posteriormente el especialista presionará el botón de cargar para completar la carga de las alternativas.	
Resultado Esperado: Que el sistema cargue correctamente las alternativas introducidas por el especialista.	
Evaluación de la Prueba: Satisfactoria	

**Tabla 37: Caso de Prueba de Aceptación 4-1**

Caso de Prueba de Aceptación	
Código Caso de Prueba: A-5-1	Nombre Historia de Usuario: Evaluar Alternativas en los Criterios
Nombre de la persona que realiza la prueba: Reinier Torres	
Descripción de la Prueba: Se ejecuta la prueba y se verifica que el especialista haya introducido la evaluación previamente	
Condiciones de Ejecución: <ul style="list-style-type: none"><li>• Se comprueba que el archivo a cargar esté en el formato establecido.</li></ul>	
Entrada / Pasos de ejecución: Cuando el sistema cargue la pantalla Evaluar Alternativas en los Criterios el sistema genera los ficheros Excel especificándole al especialista la ubicación de los mismos, el cual deberá llenar con las evaluaciones. Posteriormente el especialista presionará el botón de cargar para completar la carga de las evaluaciones.	
Resultado Esperado: Que el especialista cargue correctamente las evaluaciones de las Alternativas en los Criterios	
Evaluación de la Prueba: Satisfactoria	

**Tabla 38: Caso de Prueba de Aceptación 5-1**

Caso de Prueba de Aceptación	
Código Caso de Prueba: A-6-1	Nombre Historia de Usuario: Generar Criterio
Nombre de la persona que realiza la prueba: Javier Padrón	
Descripción de la Prueba: Se ejecuta la prueba y se verifica que se ordene correctamente.	
Condiciones de Ejecución: <ul style="list-style-type: none"><li>• Que se halla cargado previamente toda información.</li></ul>	
Entrada / Pasos de ejecución: Para realizar esta historia es necesario que se halla persistido satisfactoriamente toda la información antes mencionada. Luego de haber verificado esto se le muestra al especialista una tabla con las alternativas ordenadas descendentemente.	
Resultado Esperado: Que se ordenen las alternativas adecuadamente	
Evaluación de la Prueba: Satisfactoria	

**Tabla 39: Caso de Prueba de Aceptación 6-1**

Caso de Prueba de Aceptación	
Código Caso de Prueba: A-7-1	Nombre Historia de Usuario: Gestionar Usuarios.
Nombre de la persona que realiza la prueba: Javier Padrón	
Descripción de la Prueba: Se ejecuta la prueba y se verifica que se modifiquen, se agreguen y se eliminen los usuarios.	
Condiciones de Ejecución: <ul style="list-style-type: none"><li>• Que se haya usuarios registrados en el sistema.</li></ul>	
Entrada / Pasos de ejecución: El usuario especialista podrá administrar los usuarios que podrán interactuar con la aplicación.	
Resultado Esperado: Que se gestionen correctamente los usuarios.	
Evaluación de la Prueba: Satisfactoria	

**Tabla 40: Caso de Prueba de Aceptación 7-1**

Caso de Prueba de Aceptación	
Código Caso de Prueba: A-8-1	Nombre Historia de Usuario: Seleccionar Decisión.
Nombre de la persona que realiza la prueba: Reinier Torres	
Descripción de la Prueba: Se ejecuta la prueba y se verifica que se muestre el listado de las decisiones previamente cargadas.	
Condiciones de Ejecución:	
Entrada / Pasos de ejecución: Para la ejecución de esta historia el especialista podrá regenerar cualquier decisión cargada por él.	
Resultado Esperado: Se muestra un listado con las decisiones anteriormente cargada.	
Evaluación de la Prueba: Satisfactoria	

**Tabla 41: Caso de Prueba de Aceptación 8-1.**

## Anexo 7 Interfaces de Usuario



**Figura 19. Registrarse.**

**Nombre de la Decisión:**

**Objetivo de la Decisión:**



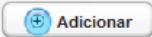
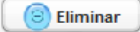
 **Siguiente**  **Cancelar**

Figura 20: Crear una Decisión.

**Gestionar Criterios**

**Nombre**  **Descripción**   **Adicionar**

NOMBRE	DESCRIPCIÓN

 **Eliminar**




 **Atras**  **Siguiente**  **Cancelar**

Figura 21: Gestionar Criterios.

**Seleccionar los Especialistas a Consultar**

**Cantidad de Espacialistas:**

Figura 22: Seleccionar una cantidad de especialistas.



Figura 23: Gestionar Alternativas.



Figura 24: Gestionar los pesos de las alternativas evaluadas en los criterios, además finalizar la creación de la decisión y guardar todos los datos.

**Seleccionar una Decisión**

NOMBRE DE LA DECISIÓN	OBJETIVO DE LA DECISIÓN
decision 1	asdsadsa


 **Correr esta Decisión**

Figura 25: Seleccionar una decisión existente.