

**Universidad de las Ciencias Informáticas**

**Facultad 3**



**Título: “Implementación de los métodos Proceso de Análisis Jerárquico y Proceso Analítico en Red en un sistema de soporte a la toma de decisiones multicriterio”**

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

**Autor(es):** Idel Jorge Sánchez González  
Adrián Román González Landeiro

**Tutor(es):** Ing. Lizandra Arza Pérez  
Ing. Hermes Miguel Velázquez Domínguez

La Habana, Mayo 2013  
“Año 55 de la Revolución”



*“Pero la juventud tiene que crear. Una juventud que no crea es una anomalía realmente. Quien no añade nada a sus conocimientos, los disminuye”*

*Ernesto Guevara de la Serna*

## DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos al Macro proyecto de investigación: Modelo de integración docencia-producción-investigación de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Idel Jorge Sánchez González  
Autor

---

Adrian Román González Landeiro  
Autor

---

Ing. Hermes Miguel Velázquez Domínguez  
Tutor

---

Ing. Lizandra Arza Pérez  
Tutor

## DATOS DE CONTACTO

**Lizandra Arza Pérez:** Graduada de Ingeniería Informática en la CUJAE, año 2002. Profesora Asistente. Directora de Gestión de Proyectos UCI.

**Hermes Miguel Velázquez Domínguez:** Graduado de Ingeniería Informática en la UCI, año 2008. Profesor.

## AGRADECIMIENTOS

*A la Revolución, por permitirnos la educación y formación profesional, por hacernos personas dignas de estos tiempos.*

*A mis abuelos Idelis y Víctor, todo lo que soy se lo debo a esas dos personitas que han dedicado sus vidas para lograr que estuviera aquí hoy realizando mi sueño, mil gracias por ser el ejemplo que son para mí. A mis otros abuelos Chenco y Mariana, gracias por su amor, por siempre estar ahí para mí.*

*A mi mamá por quererme tanto, por tener el corazón tan grande que tiene, por ser mi ejemplo de fuerza, de trabajo, de sacrificio, de amor, hoy soy una mejor persona gracias a ti.*

*A mi papá que aunque ya no está físicamente, siempre estará conmigo, gracias por dejarme una meta en la vida: ser como tú.*

*A mi hermana Lizandra por tanto amor, por siempre hacerme feliz, a mi hermana Irina por su cariño.*

*A mi tía Nany por quererme tanto, a mi tío Ibraín por sus consejos siempre oportunos.*

*A mi novia Diliana, la mujer más linda del mundo, gracias por soportarme estos 5 añitos, gracias por ser mi luz en la oscuridad, siempre que he estado triste o cansado tu me curas con esa sonrisa mágica que me alegra el alma, gracias por quererme tanto, por ser tan atenta conmigo, si he llegado hasta aquí es por tu apoyo.*

*A Albita que ha sido una segunda madre con su apoyo, confianza y cuidados.*

*A mis amigos que durante estos años han sido mi familia y que siempre han estado presentes para mí.*

*A los profesores que ayudaron a nuestra formación personal.*

*A mi segunda familia aquí en la escuela mis compañeros de la FEU, gracias a todos por su amistad, mucho de lo que soy hoy se lo debo al tiempo que pasé trabajando con ustedes.*

*A Lizandra y a Hermes por su apoyo, a Angel Luis y a Yanet por la ayuda incondicional.*

*A mi compañero de tesis Adrian, por ser el amigo que es, por su ayuda en estos años y por abrirme caminos que sirvieron en mi formación como persona y como profesional, a él y a su familia por su cariño.*

**Idel**

*A la Revolución y en especial a nuestro Comandante en Jefe por guiarme siempre y darme la oportunidad de formarme como profesional en una universidad de excelencia como esta.*

*A la FEU, esa organización por la que tanto amor siento y que no se reduce a unas pocas personas, siempre la llevaré en mi corazón, gracias por ser el mayor complemento de formación integral durante mi carrera.*

*A mi mamá, por el amor infinito que me ha brindado siempre, por aguantar mis malcriadeces que no fueron pocas, por ser un ejemplo de sacrificio y enseñarme que por las personas que realmente se quieren todo esfuerzo es poco.*

## AGRADECIMIENTOS

*A mi papá, por ayudarme tanto, por siempre darme sus consejos sabios al punto de cansarme de tantos repetírmelos, por dotarme de la capacidad de enfrentar cualquier problema que la vida me ponga por delante.*

*A mi hermana linda, por todas las cosas buenas que ha hecho por mí, decirle que aunque nos hemos fajado mucho ella siempre ha sido un ejemplo para mí y el motor impulsor para superar todas las adversidades que la carrera me ha puesto delante. Yo no pude estar en su tesis y aprovecho este momento para pedirle disculpas y decirle que estoy muy contento de que ella si esté en la mía.*

*A mi tío Pipo, por ser un ejemplo de hombre a seguir, siempre he tratado de imitarlo por lo mucho que lo quiero y admiro. A mi tía Ana, siempre ayudándome, quiero que sepas que aunque no llevamos la misma sangre también te quiero mucho.*

*A mi abuela Adelaida y mi tío Lázaro, siempre los querré mucho, ustedes han sido un eslabón fundamental para que haya podido llegar a este momento y siempre les estaré agradecido.*

*A mi abuelos, Teté, Yaya y Ho, aunque no estén físicamente conmigo los recordaré por siempre y los llevaré en el corazón.*

*A mi novia Isabel y su familia por acogerme como un miembro más, me considero un hombre muy afortunado por poder contar con una mujer tan linda, inteligente y comprensiva, ha hecho una hazaña por aguantarme durante más de tres años y espero que sea por el resto de la vida. TQM mi princesita.*

*A mi compañero de tesis Idel, por ser un ejemplo de humildad, compañerismo y solidaridad, le debo en gran medida el poder graduarme hoy, no solo por la tesis, fueron muchas las horas de sueño sacrificadas por él en la carrera para que yo pudiera vencer prácticamente el 100% de las asignaturas, él no solo es mi compañero, lo considero un hermano, siempre podrá contar conmigo.*

*A mis tutores por la ayuda con la tesis, al Fua que fue como un tutor más, y al resto de los tesisistas que logramos hacer un gran equipo.*

*A una familia que a pesar de estar lejos me ha demostrado que no toda persona que se aleja de sus amistades o conocidos y se abre paso en la vida tiene la facilidad de olvidar, me refiero a Daniel, su esposa Merlys y su niña Daniella, sé que puedo contar con ellos al igual que ellos conmigo.*

*Quiero agradecer al resto de mi familia, amistades y compañeros tanto de la vida como los que conocí acá en la Universidad, no me atrevo a mencionarlos a todos porque creo que necesitaríamos mucho tiempo y posiblemente se queden algunos, solo que les estoy muy agradecidos a todos, siempre podrán contar conmigo en cualquier circunstancia.*

*Los quiero a todos,*

**Adrian**

## DEDICATORIA

*A mis abuelos, a mi mamá, a mi papá* pues todo lo que soy y seré en la vida se lo agradezco a ellos. Porque en cada tropiezo, ante cada momento difícil de mi vida siempre tuve de ellos un gesto de amor y dedicación. Siempre me apoyaron para que estudiara, me superara y me enseñaron que con esfuerzo todo se puede alcanzar. Gracias por haber sido mi motor impulsor en los años de estudios. A toda mi familia en general que es el tesoro más importante que tengo.

*A mi novia* por lo especial y maravillosa que ha sabido ser conmigo, por brindarme tanto amor y comprensión, por ser paciente y ayudarme tanto en todo.

**Idel**

*A la Revolución, familia, amigos y compañeros.*

**Adrian**

El Proceso de Análisis Jerárquico (AHP) y el Proceso Analítico en Red (ANP) son métodos de evaluación y decisión multicriterio. En la aplicación manual de estos métodos se identifican problemas debido a la gran cantidad de información que se maneja y la cantidad de cálculos matemáticos que se realizan, muchos de ellos complejos, lo que hace que sea susceptible a errores y demore el proceso de toma de decisiones. De ahí que el presente trabajo describe los resultados de la investigación orientada a implementar un módulo para un sistema de apoyo a la decisión que permita la aplicación de los métodos AHP y ANP. Para su desarrollo se utilizó la metodología SXP, como sistema gestor de base de datos PostgreSQL, los marcos de trabajo Vaadin, Spring e Hibernate y el entorno integrado de desarrollo Eclipse STS. El módulo implementa 17 funcionalidades que permiten la ejecución de los pasos planteados por los métodos. La aplicación de pruebas de caja negra permitió validar las funcionalidades de la aplicación y los resultados de las métricas TOC y RC validan el diseño realizado. Con las pruebas y las validaciones realizadas se puede concluir que la herramienta desarrollada cumple con las especificaciones y requisitos definidos en la etapa de concepción del sistema. Se comprueba mediante un caso de estudio las variables definidas en la investigación, a partir de desarrollar un caso manual, utilizando el software Jerarquías y el sistema desarrollado MultiDecision PAAT.



Introducción .....	13
CAPÍTULO 1: Marco Teórico Referencial.....	17
Sistemas de Soporte a la Decisión.....	17
1.1.1. Proceso de toma de decisión. ....	17
1.1.2. Sistemas de soporte a la toma de decisiones. ....	19
1.2. Los métodos de toma de decisión multicriterio: Proceso de Análisis Jerárquico (AHP) y Proceso Analítico en Red (ANP).....	22
1.3 Herramientas y metodologías de desarrollo.....	27
1.3.1. Metodologías para el desarrollo .....	27
1.3.2 Lenguaje de Modelado UML.....	33
1.3.3 Herramientas Case Visual Paradigm .....	33
1.3.4 Lenguaje de programación Java .....	34
1.3.5 Marcos de trabajo.....	34
1.3.6 Gestor de Base de Datos PostgreSQL.....	36
1.3.7 Entorno de Desarrollo Integrado (IDE) Eclipse.....	37
1.3.8 Servidor de Aplicaciones Tomcat .....	37
Conclusiones Parciales .....	37
CAPÍTULO 2: Características del Sistema .....	39
2.1. Descripción del problema.....	39
2.2. Solución Propuesta.....	39
2.3 Concepción inicial del sistema .....	40
2.4 Captura de requisitos .....	40
2.4.1 Historias de usuarios del negocio.....	40
2.4.2 Lista de Reserva del Producto (LRP).....	41
2.5 Diseño de metáforas .....	43
2.5.1 Historias de Usuario .....	43
2.5.2 Tareas Ingenieriles.....	45
2.6 Arquitectura seleccionada .....	46
2.7 Patrones de diseño.....	48
2.7.1 Patrones GRASP (Patrones de Software para la Asignación General de Responsabilidad).....	48
2.8 Diagrama de Clases .....	49
2.9 Modelo de Datos.....	51
Conclusiones Parciales .....	53
CAPÍTULO 3 Implementación y Prueba .....	54

## ÍNDICE

3.1 Implementación.....	54
3.1.1. Diagrama de componentes.....	54
3.1.2. Diagrama de Despliegue.....	55
3.2 Diseño de los casos de prueba.....	55
3.3 Resultados de la aplicación de las métricas para la validación .....	56
3.3.1 Métrica TOC .....	57
3.3.2 Métrica RC.....	60
3.4 Validación de las variables de la investigación.....	64
Conclusiones parciales.....	66
Conclusiones.....	68
Referencias .....	70

**ÍNDICE DE FIGURAS**

Figura 1: Actividades del Proceso de Resolución de Problemas según Hurtado [2].	18
Figura 2: Actividades del Proceso de Toma de Decisión según Hurtado [2].	19
Figura 3: Fases y flujos de trabajo de SXP [27].	32
Figura 4: Esquema de la metodología SXP.	32
Figura 5: Historias de Usuario del Negocio.	41
Figura 6: Diagrama de Clases.	50
Figura 7: Modelo de datos.	52
Figura 8: Diagrama de Componentes.	54
Figura 9: Diagrama de Despliegue.	55
Figura 10: Representación de la cantidad de clases agrupadas en intervalos según la cantidad de procedimientos.	59
Figura 11: Representación en por ciento (%) de los resultados obtenidos en el atributo Responsabilidad.	60
Figura 12: Representación en por ciento (%) de los resultados obtenidos en el atributo Complejidad de implementación.	60
Figura 13: Representación en por ciento (%) de los resultados obtenidos en el atributo Reutilización.	60
Figura 14: Intervalos de las clases agrupadas según las dependencias entre ellas.	63
Figura 15: Representación en porcentos (%) de los atributos obtenidos en el atributo Acoplamiento.	63
Figura 16: Representación en porcentos (%) de los atributos obtenidos en el atributo Complejidad de Mantenimiento.	63
Figura 17: Representación en porcentos (%) de los atributos obtenidos en el atributo Cantidad de Pruebas.	63
Figura 18: Representación en porcentos (%) de los atributos obtenidos en el atributo Reutilización.	64
Figura 19: Demora en el proceso de decisión utilizando AHP.	64
Figura 20: Susceptibilidad a errores utilizando AHP.	65
Figura 21: Complejidad de la confección utilizando AHP.	65
Figura 22: Demora en el proceso de decisión utilizando ANP.	65
Figura 23: Susceptibilidad a errores utilizando ANP.	66
Figura 24: Complejidad de la confección utilizando ANP.	66

**ÍNDICE DE TABLAS**

Tabla 1: Lista de Reserva del Producto	43
Tabla 2: HU Autenticación de usuarios.	43
Tabla 3: HU Ponderar alternativas según los criterios	44
Tabla 4: Historia de Usuario Ponderar los criterios según la importancia.	44
Tabla 5: HU Obtener ranking de alternativas	45
Tabla 6: Tarea Ingenieril 1	45
Tabla 7: Tarea Ingenieril 2	45
Tabla 8: Tarea Ingenieril 3	45
Tabla 9: Tarea de Ingeniería 13.	46
Tabla 10: Tarea de Ingeniería 14.	46
Tabla 11: Tarea de Ingeniería 18.	46
Tabla 12: Tarea de Ingeniería 19.	46
Tabla 13: Tabla usuario.	52
Tabla 14: Tabla Decisión.	52
Tabla 15: Tabla AHP_Alternativa.	52

## ÍNDICE

Tabla 16: Tabla AHP_Criterio. ....	52
Tabla 17: Caso de Prueba de Autenticación de usuarios .....	56
Tabla 18: Caso de Prueba de Ponderar alternativas según los criterios .....	56
Tabla 19: Caso de Prueba de Elegir mejor alternativa según ANP .....	56
Tabla 20: Rango de valores para la evaluación técnica de los atributos de calidad relacionados con la métrica TOC [58]......	58
Tabla 21: Evaluación de las clases del sistema mediante la métrica TOC. ....	59
Tabla 22: Rangos de valores para la evaluación técnica de los atributos de calidad relacionados con la métrica RC [58]......	61
Tabla 23: Clases del Sistema evaluadas en los atributos de calidad según la métrica RC. ....	62

Hoy en día se sabe como el mundo, tanto en los negocios como en muchos procesos de la vida, se ha vuelto cada vez más competitivo, y por ello, tanto las entidades, las empresas comerciales, como las organizaciones no lucrativas reconocen como recurso principal a sus profesionales. El valor de los profesionales en muchos casos se demuestra en su capacidad y experiencia al momento de tomar decisiones, pero las decisiones de hoy difícilmente se toman de forma individual o aislada, se ha hecho imprescindible la concurrencia de grupos que aporten su experiencia a la hora de resolver problemas de decisión.

“La toma de decisiones es un proceso de selección entre cursos alternativos de acción, basados en un conjunto de criterios, para alcanzar uno o más objetivos” [1]. En todas las organizaciones existen problemas de diversa naturaleza, sin embargo tienen un denominador común: la necesidad de elegir entre diferentes alternativas que han de evaluarse según los criterios. En este contexto, los decisores tienen como función principal seleccionar las alternativas adecuadas para así lograr los mejores resultados. En muchas ocasiones un proceso de decisión está marcado por la complejidad a la hora de elegir la mejor alternativa en función de criterios tanto de carácter cualitativo como cuantitativo y que a menudo pueden entrar en conflicto. Como los problemas de toma de decisiones son procesos complejos en los cuales intervienen múltiples criterios, se hace necesario utilizar herramientas que permitan discernir entre estos para obtener una solución que satisfaga en mejor grado la combinación de alternativas posibles. Tradicionalmente muchos decisores piensan que estos problemas deben ser resueltos mediante la utilización de técnicas sofisticadas como herramientas de apoyo a la hora de adoptar las decisiones.

Los Sistemas de Soporte a la Decisión son herramientas que tienen como propósito apoyar y facilitar el proceso de toma de decisiones, a través del procesamiento de la información disponible para brindar criterios oportunos y confiables de relevancia para los decisores. La mayoría de estos sistemas implementan métodos los cuales definen procedimientos que se deben seguir para resolver problemas decisorios.

Los métodos de evaluación y decisión multicriterio comprenden la selección de entre un conjunto de alternativas factibles, la optimización con varias funciones objetivo simultáneas, un agente decisor y procedimientos de evaluación racionales y consistentes. Es decir, en un problema de decisión multicriterio se trata de identificar la mejor o las mejores soluciones considerando simultáneamente múltiples criterios en competencia [2].

Algunos de estos métodos son: Proceso de Análisis Jerárquico (AHP), Proceso Analítico en Red (ANP), TOPSIS y los métodos PROMETHEE, durante la aplicación de los mismos se identifican problemas en los casos donde se manejan numerosas alternativas y gran variedad de criterios. Lo

## INTRODUCCIÓN

anterior implica que el proceso de aplicación de estos métodos en esos casos es complejo y su confección de forma manual se torna en ocasiones impracticable, debido a la gran cantidad de información que se maneja y la cantidad de cálculos matemáticos que se realizan, muchos de ellos complejos.

Se han realizado intentos para resolver o atenuar este problema digitalizando algunos métodos por separado. En la Universidad de las Ciencias Informáticas (UCI) se está desarrollando un sistema que contendrá la implementación de dos métodos de evaluación y decisión multicriterio: TOPSIS y los métodos PROMETHEE para elegir la mejor alternativa en cada uno de los problemas a resolver teniendo en cuenta los criterios arrojados por cada uno de estos métodos. Por lo que se plantea el siguiente **problema**: *La complejidad de la confección manual de los métodos Proceso de Análisis Jerárquico (AHP) y Proceso Analítico en Red (ANP), propicia que el proceso de aplicación de los mismos sea susceptible a errores y demora el proceso de toma de decisiones.*

Como **objeto de estudio** se define los *Sistemas de soporte a la toma de decisiones multicriterio* y como **campo de acción** los *métodos de toma de decisiones multicriterio*.

Se define como **objetivo general** de la presente investigación: *Desarrollar un módulo para el sistema de soporte a la toma de decisiones multicriterio que implemente los métodos Proceso de Análisis Jerárquico (AHP) y Proceso Analítico en Red (ANP) que elimine la complejidad de la confección manual de los mismos, evite que el proceso de aplicación sea susceptible a errores y agilice el proceso de toma de decisiones.*

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 multicriterio.*
2. *Modelar las funcionalidades de los métodos AHP y ANP como parte del sistema de ayuda a la toma de decisión multicriterio.*
3. *Diseñar e implementar las funcionalidades modeladas.*
4. *Validar la solución propuesta.*

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

1. *Caracterizar los sistemas de ayuda a la toma de decisión multicriterio.*
2. *Analizar los métodos para la toma de decisión multicriterio.*
3. *Caracterizar los métodos de toma de decisiones multicriterio AHP y ANP.*
4. *Seleccionar y caracterizar la metodología, las herramientas y las tecnologías a utilizar para el desarrollo de la solución propuesta.*
5. *Definir las funcionalidades del módulo AHP y ANP.*

## INTRODUCCIÓN

6. *Elaborar el diseño de la solución.*
7. *Realizar el modelo de datos.*
8. *Elaborar el diagrama de componentes.*
9. *Implementar las funcionalidades definidas.*
10. *Validar el diseño mediante la aplicación de métricas.*
11. *Validar las variables de la investigación.*
12. *Validar la solución 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 multicriterio que implemente los métodos Proceso de Análisis Jerárquico (AHP) y Proceso Analítico en Red (ANP) se eliminará la complejidad de la confección manual de los mismos, se evitará que el proceso de aplicación sea susceptible a errores y se agilizará el proceso de toma de decisiones.*

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 [3].

**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 [4]. En este caso se utilizó para el análisis de la información y para llegar a conclusiones sobre el estudio realizado sobre los métodos de toma de decisión multicriterio.

**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 [4]. En esta investigación se le dio utilidad para establecer la evolución y tendencia de los sistemas de apoyo a la decisión.

### ✓ **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 [5].

**Observación:** Es un método para reunir información visual sobre lo que ocurre, lo que el objeto de estudio hace o cómo se comporta [6]. En este caso se utilizó para entender cómo se han utilizado en la Universidad de las Ciencias Informáticas los métodos AHP y ANP en la solución de problemas de toma de decisiones multicriterio.

## INTRODUCCIÓN

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

### ✓ *Técnicas de recopilación de información*

La recopilación de datos se utiliza para verificar los métodos empleados en lo investigado, para llegar a la conclusión del suceso, teniendo las pruebas y una serie de pasos que se llevan a cabo para comprobar la hipótesis planteada [7].

**Entrevistas:** Es una técnica para obtener datos, que consiste en un diálogo entre dos personas [7]. 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 AHP y ANP en la solución de problemas.

El documento de tesis consta de una introducción, tres capítulos, las conclusiones, recomendaciones, las bibliografías utilizadas y el cuerpo de anexos. El Capítulo 1 establece el marco teórico referencial de la investigación, mientras que el Capítulo 2 presenta los resultados de la modelación y el análisis del sistema. Por su parte el Capítulo 3 describe la implementación del sistema y los resultados de la validación de la aplicación desarrollada.



Para el desarrollo de la presente investigación se hizo necesaria la revisión, estudio y 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 los sistemas de soporte a la decisión. De igual forma se realiza un análisis de los métodos a implementar para entender su funcionamiento y para definir en que circunstancias es más idóneo utilizarlos, así como la metodología de desarrollo y herramientas a utilizar.

### **Sistemas de Soporte a la Decisión.**

Según SlideShare [8] el proceso de tomar una decisión es una de las actividades que se realiza en el mundo de los negocios con mayor frecuencia. La misma se presenta en todos los niveles de la organización, ya sean asistentes, auxiliares, o directores generales de las empresas. En cualquier caso se tiene uno o varios objetivos a cumplir teniendo en cuenta un conjunto de restricciones. Los Sistemas de Soporte a la Decisión tienen como propósito apoyar y facilitar este proceso, a través del procesamiento de la información disponible para brindar criterios oportunos y confiables de relevancia para los decisores.

#### **1.1.1. Proceso de toma de decisión.**

Una investigación realizada por Soto [9] plantea que la decisión comienza al aparecer un problema que afecta a un individuo y teniendo en cuenta que, para resolverlo, dispone varias alternativas de solución. La decisión es un acto del presente cuyo resultado obtendrá un efecto en el futuro. El decisor adopta la decisión de acuerdo al enfoque que se posee de dicho problema y basándose en la experiencia del pasado. El acto de decidir es un trabajo intelectual, ya que es construido mentalmente por el decisor, basándose en sus experiencias, conocimientos y expectativas.

Otras definiciones de decisión son las planteadas por Gómez [10] que define “*una decisión como el proceso de elegir la solución para un problema suponiendo que existen varias alternativas*” y la planteada por Simon [1] “*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*”. Se puede destacar después de esta revisión, que de manera general las distintas definiciones tienen elementos comunes, entre los que se señalan los siguientes:

## CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

- Se elige o selecciona.
- En función de criterios.
- Entre un grupo de alternativas.
- Para lograr un objetivo.

El proceso de toma de decisiones según Hurtado [2] agrupa actividades que van desde la identificación de un problema hasta la solución del mismo haciendo uso de la alternativa seleccionada, por lo que se concluye que es un proceso enmarcado en problemas donde se deben encontrar las alternativas para su solución.

El mismo autor Hurtado [2] plantea que la toma de decisión se desarrolla bajo determinadas circunstancias, teniendo en cuenta las características de la información que se maneja y las posibles alternativas que se tienen, estableciendo las siguientes categorías:

- Toma de decisiones bajo certidumbre: En la que se conocen los datos de forma determinista, toda la información que se debe manejar en el proceso se tiene con total certeza, de fuentes confiables.
- Toma de decisiones bajo riesgo: En la que los datos se describen mediante distribuciones de probabilidad.
- Toma de decisiones bajo incertidumbre: En los que la información que se conoce no es precisa, existe vaguedad e incertidumbre en cuanto a la validez de la misma, no es posible representar la información de manera determinista.

En Hurtado [2] el proceso de resolución de problemas consta de siete fases, siendo la toma de decisiones el término que se asocia a las cinco primeras etapas de dicho proceso; comenzando con la identificación y definición del problema y concluyendo con la elección de una alternativa.

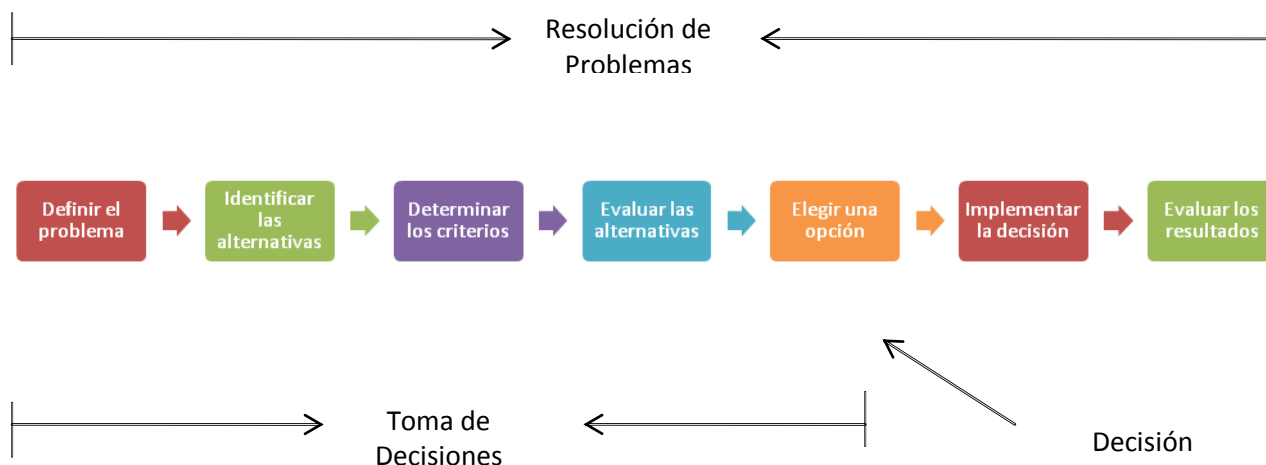


Figura 1: Actividades del Proceso de Resolución de Problemas según Hurtado [2].

Las tres primeras fases del proceso constituyen la “Estructuración del problema” y las dos últimas son el “Análisis del problema”. La fase de análisis del proceso de toma de decisiones puede asumir dos formas básicas: cualitativa y cuantitativa. El análisis cualitativo se basa primordialmente en el

## CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

razonamiento y la experiencia del decisor; a su vez la impresión intuitiva que este presenta del problema. Cuando se utiliza el enfoque cuantitativo, el analista se concentra en los hechos o datos asociados al problema y desarrolla expresiones matemáticas que describen los objetivos, las restricciones y las relaciones existentes en el problema. Después utilizando uno o más métodos cuantitativos, el analista ofrece una recomendación con base en los aspectos cuantitativos del problema.



Figura 2: Actividades del Proceso de Toma de Decisión según Hurtado [2].

Se establece en Hurtado [2] que a los problemas que no implican más de un criterio de decisión se le denomina problemas de decisión de criterio único y en el caso contrario se les denomina problemas de criterios múltiples o problemas de decisión multicriterio. En la investigación los problemas a resolver serán de decisión multicriterio.

### 1.1.2. Sistemas de soporte a la toma de decisiones.

El término “sistema de soporte a las decisiones” (SSD) ha sido utilizado de maneras diferentes, y definido de varias formas en dependencia del punto de vista de cada autor. Por ejemplo, Finlay lo define como “*un sistema basado en computadora que soporta el proceso de toma de decisiones*” [11]. Turban lo define más específicamente como “*un sistema de información basado en computadora, especialmente desarrollado para dar apoyo a la solución de un problema no estructurado de dirección para mejorar la toma de decisiones. Utiliza datos, proporciona una interfaz fácil de utilizar y permite considerar la propia visión del decisor*”. [12]

Para Little “*un SSD es un modelo basado en establecer procedimientos para el procesamiento de datos y juicios, con el fin de asistir a un director en la elaboración de una decisión*” [13]. Keen plantea que “*un SSD combina los recursos intelectuales de los individuos con las capacidades de una computadora para mejorar la calidad de las decisiones*” [14]. Moore los define como “*sistemas extensibles capaces de soportar análisis de conocimiento*” [15]. De acuerdo a Power, el término “sistema de soporte a las decisiones” permanece como un término inclusivo y útil para muchos tipos de sistemas de información que apoyan la toma de decisiones [16].

Al estudiar las definiciones planteadas anteriormente, se puede concluir que existe diversidad de definiciones de los SSD, sin embargo, todas ellas tienen elementos en común que los autores han

## CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

considerado característicos de los SSD y que son de importante conocimiento para el desarrollo de la investigación. Estos elementos son:

- Son sistemas informáticos.
- Utilizan datos e información que se procesa y analiza.
- Permiten combinar las capacidades de la computadora con las del intelecto humano.
- Ayudan a la solución de problemas.
- Asisten a los decisores brindando criterios para la decisión.

El objetivo del mismo es proporcionar la mayor cantidad de información relevante en el menor tiempo posible, con el fin de facilitar la toma de la decisión más adecuada. Para profundizar un poco en la teoría de los SSD se describen a continuación un grupo de clasificaciones y tipos de ayuda que pueden brindar estos sistemas. Estos elementos serán útiles para establecer las características que el sistema a desarrollar debe tener de acuerdo al problema tratado.

Es importante que se tenga en cuenta las formas en la que estos sistemas pueden apoyar a los decisores, según se plantea en [17]. Entre ellas:

- Alerta al usuario para una toma de decisiones oportuna.
- Reconoce los problemas que deban resolverse en el marco del proceso de toma de decisiones.
- Resuelve los problemas reconocidos por sí mismo o por el usuario.
- Se le facilita al usuario la capacidad para procesar el conocimiento.
- Se le ofrece al usuario asesoramiento, expectativas, evaluaciones, hechos, análisis y diseños.
- Simula la percepción, la imaginación o la visión creativa de los usuarios.
- Coordina o facilita las interacciones entre los participantes y los decisores.

El tipo de ayuda a brindar por el sistema desarrollado debe definirse a partir del conocimiento del problema y la información para su solución, así como las expectativas de los decisores respecto a lo que desean que se realice en la aplicación.

### 1.1.2.1. Clasificación

Al igual que las definiciones, existe diversidad de criterios para clasificar los SSD.

Según la relación con el usuario [18]:

- 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 cuando se genere una solución consolidada.
- Activos: presenta sugerencias y soluciones explícitas.

## CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

- Pasivos: que no puede presentar sugerencias o soluciones explícitas.

### Según la naturaleza de las decisiones [19]:

- Estructurados: Son aquellos en los que se pueden definir reglas y procedimientos de decisión preestablecidos, las variables que intervienen son conocidas y repetitivas.
- No estructurados: No son repetitivas y no poseen reglas predefinidas, no existen puntos de referencia idénticos en el pasado, las reglas deben ser creadas para cada ocasión. El papel del sistema de información en estos casos es de apoyo al decisor.

### Según la evolución que han experimentado [20]:

- Sistemas de soporte orientados al procesamiento de datos: EDP, OAS, MIS.
- Sistemas de soporte orientados al análisis de información: DSS, EIS, y los sistemas de soporte para la decisión en grupo (GDSS, Group decision support system).
- Sistemas de soporte orientados al conocimiento: DW, aplicaciones de exploración (Data Mining).

### **1.1.2.2. Características, usos, beneficios y limitaciones.**

Los SSD son aplicaciones informáticas que reúnen un grupo de características para ser clasificados como tal, según Delfos las características generales de este tipo de sistemas son:

- Decisiones en contexto semiestructurado o no estructurado.
- Brinda soporte a los encargados de las decisiones sin tratar de reemplazarlos.
- Respalda todas las etapas del proceso de toma de decisiones.
- Apunta hacia la eficacia más que hacia la eficiencia.
- Se encuentra bajo control del usuario encargado de la decisión.
- Usa modelos y datos básicos.
- Es interactivo y amigable al usuario.
- Está basado en un proceso iterativo.
- Proporciona soporte a todos los niveles administrativos.
- Respalda decisiones individuales y grupales [20].

Los beneficios que según Delfos tiene el 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 [20].

## CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

Mientras que entre las limitaciones que según Delfos tiene el 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 SSD no puede llegar a usarse de manera generalizada en múltiples contextos de toma de decisiones [20].

Una vez estudiada y analizada la teoría alrededor de los SSD, es importante la revisión de la metodología y explicación de cómo se aplican los métodos a implementar, cuáles son sus principales características, ventajas y desventajas.

### **1.2. Los métodos de toma de decisión multicriterio: Proceso de Análisis Jerárquico (AHP) y Proceso Analítico en Red (ANP).**

Los métodos de evaluación y decisión multicriterio comprenden la selección entre un conjunto de alternativas factibles, la optimización con varias funciones objetivos simultáneas, un agente decisor y procedimientos de evaluación racionales y consistentes. Sus principios se derivan de la Teoría de Matrices, Teoría de Grafos, Teoría de las Organizaciones, Teoría de la Medida, Teoría de las Decisiones Colectivas, Investigaciones de Operaciones y de Economía [2].

Existen varios métodos multicriterios de ayuda a la toma de decisiones, el problema de seleccionar la mejor alternativa de las evaluadas, o mejor aún, el obtener una ordenación de “mejor” a “peor” (ranking) basada en un conjunto de criterios (cuantitativos o no) es la cuestión central que resuelven estos métodos.

Sobresalen, entre todas, dos formas de modelar las preferencias de decisiones multicriterios representados por las llamadas escuelas: norteamericana o Normativa y la escuela europea o Descriptiva.

Con el trabajo de Bernard Roy en los años 70 y la contribución de varios científicos europeos, la escuela Descriptiva fue la fundadora de la metodología de Ayuda a la Decisión Multicriterio. En la escuela Descriptiva se destacan la familia de los métodos ELECTRE (The ELimination Et Choix Traduisant la REalité) y de los métodos PROMETHEE (Preference Ranking Organization method of Enrichment Evaluations), métodos también llamados de superación [21].

La escuela Normativa propone varios métodos prácticos, como son los métodos MAUT (Teoría de la Utilidad Multiatributo), suma ponderada, el método SMART (The Simple Multi-attribute Rating Technique) y el método de las Jerarquías Analíticas o Proceso de Análisis Jerárquico (AHP) que según sus autores constituyen una teoría en sí mismo [21].

## CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

En esta corriente el método más popular es el AHP, introducido y desarrollado por Tomás Saaty a finales de la década de los 70 y consiste en formalizar la comprensión intuitiva de un problema multicriterio complejo, mediante la construcción de un modelo jerárquico, que le permite al agente decisor estructurar el problema de forma visual, el modelo jerárquico básicamente contiene tres niveles: meta u objetivo, criterios y alternativas [22] [2].

El AHP es un método matemático creado para evaluar alternativas cuando se tienen en consideración varios criterios y se basa en la idea de que la complejidad inherente a un problema de toma de decisión con criterios múltiples, se puede resolver mediante la jerarquización de los problemas planteados. Tiene como principio que la experiencia y el conocimiento de los actores son tan importantes como los datos utilizados en el proceso. Este método puede considerarse, según la orientación dada al mismo, de muy diversas maneras [23]. Su contribución es importante en niveles operativos, tácticos y estratégicos, sirviendo para mejorar el proceso de decisión debido a la gran información que aporta y a la mejora en el conocimiento del problema [2].

Las 8 etapas o pasos del método AHP son las siguientes:

1. Descomponer el Problema de Decisión en una jerarquía de elementos interrelacionados, identificando: la Meta u Objetivo General, los Criterios ( $i=1, 2, \dots, m$ ) y las alternativas posibles ( $j= 1, 2, \dots, n$ ).

Para cada uno de los “m” criterios, repetir las etapas 2 a 5.

2. Desarrollar la Matriz de Comparación por Pares (MCP) de alternativas para cada uno de los criterios estableciendo el peso de importancia relativa entre ambas alternativas consideradas, el peso se establece a partir de la siguiente escala.

1 = igualmente preferida.

3 = moderadamente preferida.

5 = fuertemente preferida.

7 = muy fuertemente preferida.

9 = extremadamente preferida.

Pueden asignarse los valores intermedios 2, 4, 6, 8. Un peso recíproco (ej.  $1/9, 1/7, 1/5, 1/3, \dots$ ) se aplica cuando la segunda alternativa es preferida a la primera. El valor 1 es siempre asignado a la comparación de una alternativa con sí misma.

3. Desarrollar la Matriz Normalizada (MN) dividiendo cada número de una columna de la Matriz de Comparación por Pares por la suma total de su columna.
4. Desarrollar el Vector de Prioridad para el Criterio calculando el promedio de cada fila de la Matriz Normalizada. Este promedio por fila representa el Valor de Prioridad de las alternativas con respecto al criterio seleccionado.

## CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

5. La Consistencia de las opiniones utilizadas en la Matriz de Comparación por pares puede ser determinada a través de la Razón de Consistencia (RC). Una RC inferior o igual a 0.10 es considerado aceptable. Para aquellos casos en que  $RC > 0.10$ , las opiniones y juicios deberán ser reconsiderados.
6. Luego de que la secuencia de 2-3-4-5 ha sido ejecutada para todos los criterios, los resultados obtenidos en 4 son resumidos en una Matriz de Prioridad (MP), listando las alternativas por fila y los criterios por columna.
7. Desarrollar una Matriz de Comparación de Criterios por pares de manera similar a lo que se hizo para las alternativas en 2-3-4.
8. Desarrollar un Vector de Prioridad Global multiplicando el Vector de Prioridad de los criterios 7, por la Matriz de Prioridad de las alternativas 6.

Para la determinación de la Razón de Consistencia:

1. Para cada fila de la Matriz de Comparación por pares, determinar una suma ponderada en base a la suma del producto de cada celda por la prioridad de cada alternativa correspondiente.
2. Para cada fila, dividir su suma ponderada por la prioridad de su alternativa correspondiente.
3. Determinar la media  $n_{max}$  del resultado de la etapa 2.
4. Calcular el Índice de Consistencia para cada alternativa

$$IC = \frac{n_{max} - n}{n - 1}$$

5. Determinar el Índice de Aleatoriedad por la fórmula

$$IA = \frac{1.98(n - 2)}{n}$$

6. Determinar la Razón de Consistencia

$$RC = \frac{IC}{IA}$$

Ventajas del AHP frente a otros métodos de Decisión Multicriterio:

- Presenta un sustento matemático.
- Permite desglosar y analizar un problema por partes.
- Permite medir criterios cuantitativos y cualitativos mediante una escala común.
- Incluye la participación de diferentes personas o grupos de interés y generar un consenso.
- Permite verificar el índice de consistencia y hacer las correcciones, si se da el caso.
- Genera una síntesis y da la posibilidad de realizar análisis de sensibilidad.
- Es de fácil uso y permite que su solución se pueda complementar con métodos matemáticos de optimización [24].



## CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

El método AHP es simple, flexible, de fácil e intuitiva comprensión, se puede integrar con otros métodos multicriterio y a pesar de que cuenta con varios softwares de apoyo para su aplicación como Hipre3+, Inpre, Criterium® y Expert Choice, creado con la colaboración de su creador, Tomás Saaty, presenta una serie de inconvenientes tales como:

- La interpretación de las prioridades totales obtenidas en el procedimiento.
- La introducción de una nueva alternativa puede hacer variar la estructura de preferencias del decisor, o poner de manifiesto alguna inconsistencia en los juicios.
- En algunos casos se verifican inconsistencias y carencia de robustez en los resultados (reversión de rangos) [22].

HIPRE 3+ presenta una sencilla herramienta que es utilizada por quienes toman las decisiones en sí. Está desarrollado en DOS y ofrece una interfaz gráfica visual y personalizable. Esto hace que la estructuración, priorización y análisis de problemas complejos es un proceso fácil e intuitivo. La versión demo de HIPRE 3+ se limita a ejecutar los modelos que tienen un máximo de tres niveles e igual número de elementos en cada uno. La versión demo no permite guardar los modelos. En la versión completa de HIPRE 3+ puede desplazar la pantalla y no puede haber un máximo de 50 ubicaciones de elementos en cada uno de los 20 niveles. Se necesita una copia con licencia para utilizar las funciones HIPRE 3+.

La aplicación de Criterium DecisionPlus ayuda a tomar decisiones precisas de manera rápida y eficaz. Criterium DecisionPlus 3.0 es la versión actual. Compensaciones directas, los modelos más grandes, gráficos potentes y amplias opciones significa que la CDP 3.0 es compatible con perspicaz, persuasivo, toma de decisiones más rápida y para los modelos más complejos que nunca. Se necesita licencia para su adquisición y uso.

Expert Choice asiste a los decisores organizando la información relacionada a la complejidad del problema en un modelo jerárquico consistente de un objetivo, escenarios posibles, criterios y alternativas. Usando el método, único de Expert Choice, de comparación par a par, se puede evaluar la importancia de los criterios, las preferencias de las alternativas, las probabilidades de los escenarios y sintetizar sus comparaciones para llegar a la mejor decisión. El Expert Choice está ampliamente difundido, sin embargo se necesita licencia para su adquisición y uso.

En Cuba se desarrolló el software Jerarquías que automatiza el método AHP, el mismo es una herramienta de apoyo a la toma de decisiones empresariales donde se pretende dar solución en muy corto tiempo a problemas de gran envergadura. En este software, la información es descompuesta en una jerarquía de criterios y alternativas que después es sintetizada para determinar un ranking de las mismas, criterios tanto cualitativos como cuantitativos pueden ser incorporados y comparados usando

## CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

información y juicios para derivar pesos y prioridades. Se usa la comparación a pares, donde se expresa la importancia relativa de un criterio sobre otro. A través de una interfaz gráfica es capaz de dar un ordenamiento de todas las alternativas analizadas.

Se descartó la utilización de software desarrollados en el mundo que implementan el método AHP debido a que algunos de ellos están desarrollados con un fin específico. La presente investigación desarrollará un módulo que será de utilización genérica, es decir, que se pueda utilizar para resolver cualquier problema de decisión multicriterio, además se necesitan licencias para la adquisición y uso de estos software. Existen versiones de algunos de ellos como el Hipre3+ que pueden ser descargadas de forma gratuita, pero esas versiones no cuentan con todas las funcionalidades del sistema original. Otra causa es que la mayoría de estos sistemas son desktop y el sistema a desarrollar será Web, ya que proporciona ventajas sobre estos como: compatibilidad multiplataforma, menos requerimientos de hardware, acceso inmediato, múltiples usuario concurrentes, información en línea y segura y facilidades de actualización.

El método ANP es una derivación del AHP y ambos son métodos propuestos por Saaty de apoyo a la toma de decisiones [25].

Aunque el AHP ha sido aplicado a diferentes situaciones con resultados razonables, no es capaz de abordar toda la complejidad inherente a muchos problemas del mundo real debido a su estricta estructura jerárquica. Con el objetivo de solucionar este problema, Saaty propuso el ANP. El modelo ANP representa los problemas de decisión como una red de criterios y alternativas (englobados todos bajo el término de elementos), agrupados en componentes. Todos los elementos de la red pueden relacionarse de cualquier forma posible, lo que significa que la red puede incorporar relaciones de interdependencia y realimentación. Esto proporciona un enfoque mucho más preciso para modelizar entornos complejos. La influencia de los elementos en la red puede representarse en lo que se denomina supermatriz. Este nuevo concepto es una matriz bidimensional de elementos por elementos que recoge los vectores de pesos de influencia calculados en matrices de comparación pareada [25].

El ANP comprende los siguientes pasos:

1. Modelizar el problema de decisión como una red, lo cual implica identificar los elementos de la red (criterios y alternativas), agruparlos en componentes y determinar las relaciones de interdependencia entre ellos.
2. Realizar comparaciones pareadas entre elementos.
3. Construir la denominada supermatriz no ponderada con los vectores de pesos de importancia relativa de los elementos.
4. Realizar comparaciones pareadas entre componentes.

## CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

5. Ponderar los bloques de la supermatriz no ponderada, mediante los pesos correspondientes de los componentes, para transformarla en la supermatriz ponderada.
6. Si es necesario, normalizar la supermatriz ponderada, dividiendo cada valor por la suma de las columnas. De esta forma se obtiene una matriz estocástica por columnas, es decir, cuyas columnas sumen la unidad (supermatriz ponderada estocástica).
7. Elevar la supermatriz ponderada estocástica a potencias sucesivas hasta que sus entradas converjan y permanezcan estables (supermatriz límite).

SuperDecisions-Q es un software que implementa el método ANP. Este programa ha sido elaborado por Juan Pascual Pastor Ferrando, profesor del Departamento de Proyecto de Ingeniería de la Universidad Politécnica de Valencia, está implementado en lenguaje Java, sin embargo se necesita licencia para su adquisición y uso.

En la Universidad de las Ciencias Informáticas en Cuba no existe ningún software que implemente alguno de estos métodos planteados anteriormente (AHP y ANP), solamente se ha utilizado el método AHP para la solución de problemas donde es necesario encontrar la mejor de varias alternativas teniendo en cuenta varios criterios.

### **1.3 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 referente al tiempo o la calidad del software. En Cuba existe una 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.

El estudio realizado está enfocado a la definición de lo necesario para cumplir con cada una de las partes del desarrollo de un sistema informático, es por ello que se presentan las metodologías de desarrollo y herramientas CASE, fundamentalmente para la descripción del análisis y modelación del sistema. Se describen además las herramientas de desarrollo, tanto para la implementación como el sistema gestor de base de datos.

#### **1.3.1. Metodologías para el desarrollo**

*“Desarrollar un buen software depende de un gran número de actividades y etapas”, así opina Figueroa [26] y continua su reflexión, “donde el impacto de elegir la mejor metodología para un equipo, en un determinado proyecto, es trascendental para el éxito del producto”. Este autor aclara en su investigación que el uso de las metodologías para el desarrollo de un software es esencial en un*

## CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

proyecto, la cual debe encajar en el equipo, guiar y organizar actividades que conlleven a las metas trazadas en el grupo.

La investigación realizada por Infante [27] se encuentra centrada en aquellas metodologías más tradicionales centralizadas especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, las herramientas y notaciones que se usarán. Este tipo de metodologías tienen a su favor que proveen al equipo de un alto grado de ordenamiento, de disciplina, pero al no adaptarse apropiadamente a los cambios no son métodos adecuados cuando se trabaja en un entorno donde los requisitos no pueden predecirse o pueden variar, estas metodologías se centran más en los procesos que en las personas, el cliente puede llegar a ser relegado. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en otros.

Por otro lado se presentan las metodologías ágiles centradas, según Canós [28], en el factor humano o en el producto del software. Las mismas están mostrando su efectividad fundamentalmente en proyectos que presentan requisitos cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo pero sin alterar la calidad. Las necesidades del cliente pueden variar desde el momento de contratación de un software al momento de su entrega; y es de mayor importancia satisfacer estas últimas que las primeras. Esto requiere procesos de software que en lugar de rechazar los cambios los incorpore. Las metodologías ágiles suministran una serie de normas y principios junto a técnicas que intentan hacer la entrega del proyecto menos complicada y más satisfactoria tanto para los clientes como para los equipos de entrega.

Teniendo en cuenta las características que se han mencionado de cada uno de los tipos de metodologías, se ha definido que se utilizarán metodologías ágiles para la realización del producto, debido a que permite principalmente disminuir costos, brinda flexibilidad a proyectos de software donde la incertidumbre está presente, como es el caso que se describe en esta investigación, y donde los requisitos pueden variar con el tiempo. Además de presentar un equipo de desarrollo pequeño y para el desarrollo de la aplicación es muy conveniente hacer al cliente parte del mismo.

### **1.3.1.1. Metodologías Ágiles.**

Dentro de las metodologías ágiles se estudiaron SCRUM (Gestión Ágil de Proyectos) y XP (Programación Extrema) debido a las características que presentan hacen que se ajusten a las necesidades para el desarrollo del software a implementar, además del estudio de una personalización de estas dos metodologías que se ha utilizado en proyectos de la Universidad de las Ciencias Informáticas con el nombre de SXP.

El estudio realizado por Davis [29] muestra a XP como una metodología de desarrollo ligera basada en una serie de prácticas que respaldan un aumento en la productividad al hacer software. Controla los

## CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

problemas de riesgo en los proyectos, permitiendo la participación de pequeños grupos de programadores. Esta metodología demanda un variado equipo de desarrollo. Teniendo como meta entregar el software requerido a tiempo. Una diferencia que la caracteriza, según Figueroa [26], sobre las metodologías tradicionales es que pone más énfasis en la adaptabilidad que en la previsibilidad.

El mismo autor [26], propone algunas características de este método:

- **Desarrollo iterativo e incremental:** pequeñas mejoras, unas tras otras.
- **Pruebas unitarias continuas:** frecuentemente repetidas y automatizadas, incluyendo regresión de pruebas. Se aconseja escribir el código de la prueba antes de la codificación.
- **Programación por parejas:** se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Esto supone mayor calidad del código escrito pues de esta manera el código es revisado y discutido mientras se escribe.
- **Frecuente interacción del equipo de programación con el cliente o usuario:** se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- **Corrección** de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
- **Refactorización del código:** es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- **Propiedad del código compartida:** en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- **Simplicidad en el código:** es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

Proponiendo [30] algunas desventajas:

- Es recomendable emplearlo solo en proyectos a corto plazo.
- Altas comisiones en caso de fallar.
- Conseguir su implantación en un equipo podría resultar dificultoso.

La investigación de Figueroa [26] expone además que SCRUM es un proceso ágil y liviano utilizado para administrar y controlar el desarrollo de un software. Se emplea como marco para otras prácticas de ingeniería de software, como el Proceso Unificado de Desarrollo de Software (RUP, Rational Unified

## CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

Process, según las siglas en inglés) o XP. Focalizada en priorizar el trabajo, maximizando la utilidad de lo que se construye. Los requisitos y las prioridades se examinan y ajustan durante el proyecto en intervalos cortos y regulares. Se busca entregar software que realmente resuelva las necesidades, aumentando la satisfacción del cliente.

El mismo autor Figueroa [26] propone que en SCRUM, el equipo se enfoca en construir software de calidad. Teniendo en cuenta que la gestión de un proyecto SCRUM se concentra en definir qué características debe tener el producto a construir, transformando cualquier obstáculo que pudiera entorpecer la tarea del equipo de desarrollo. Este proceso busca que los equipos sean lo más prácticos y fructíferos posible.

El investigador [26] continua su estudio al proponer que SCRUM posee un conjunto de reglas muy simples, basado en los principios de inspección continua, adaptación, auto-gestión e innovación. Teniendo como objetivo que el cliente se entusiasme y se comprometa con el proyecto, debido a que ve crecer el producto iteración tras iteración, localizando las herramientas para alinear el desarrollo con las metas de negocio de su empresa. Por otro lado, el equipo encuentra un ámbito propicio para desarrollar sus capacidades profesionales y resultando un incremento en la motivación de los integrantes del mismo.

Infante [27] expone un grupo de características sobre esta metodología:

- SCRUM por su proceso iterativo incremental produce un grupo de funcionalidades en cada fin de iteración.
- Proceso ágil para el manejo y control del trabajo de desarrollo.
- Contenedor de prácticas de ingeniería existentes.
- Enfoque basado en equipos, incrementa el desarrollo cuando los requerimientos cambian rápidamente.
- Proceso que controla el caos entre los conflictos de interés y las necesidades.
- Camino para mejorar las comunicaciones y maximizar la cooperación.
- Camino para detectar la causa y solucionar cualquier problema en el desarrollo.
- Escalable desde proyectos simples a proyectos completos, Scrum ha controlado y organizado el desarrollo de productos y proyectos con miles de desarrolladores e implementadores.

Seguido por las siguientes ventajas:

- Se trabaja en iteraciones cortas, de alto enfoque y total transparencia.
- Se acepta que el cambio es una constante universal y se adapta el desarrollo para integrar los cambios que son importantes.
- Se incentiva la creatividad de los desarrolladores haciendo que el equipo sea auto administrado.

## CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

- Se mantiene la efectividad del equipo habilitando y protegiendo un entorno libre de interrupciones e interferencias.
- Permite producir software de una forma consistente, sostenida y competitiva.
- Las reuniones se dedican a inconvenientes recientes, evitando el estancamiento [31].

**SXP**, es un híbrido cubano de metodologías ágiles (XP y SCRUM), que ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, y ayudando al líder del proyecto a tener un mejor control del mismo. 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 al éxito del proyecto. Basada completamente en los valores y principios de las metodologías ágiles expuestos en el Manifiesto Ágil. Como método de estimación se utiliza la opinión de expertos y constan con métricas o indicadores para lograr una eficiente calidad [32].

La creación de SXP con la unión de las metodologías XP y SCRUM se centra principalmente en la utilización de SCRUM para la gestión, logrando 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. Siendo XP y no otra la metodología candidata para guiar el proceso ingenieril, caracterizándose por presentar una documentación discreta y de mayor dinamismo. La idea de las duplas para el desarrollo resultó interesante para los investigadores, pues en pequeñas iteraciones dos desarrolladores lograrían hacer lo mismo que un equipo grande de desarrollo (analista, arquitecto, diseñador, desarrollador, probador). SCRUM es entonces la metodología ideal para toda la gestión de proyectos, sirviendo de soporte para acelerar el dinamismo identificado en XP. La identificación de los pequeños sprint (iteraciones) y las reuniones diarias con el SCRUM Máster (líder del proyecto) se acercaban a la disciplina que se quería alcanzar por los desarrolladores [32].

Este híbrido consta de 4 fases principales: Planificación-Definición donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto; Desarrollo, es donde se realiza la implementación del sistema hasta que esté listo para ser entregado; Entrega, puesta en marcha; y por último Mantenimiento, donde se realiza el soporte para el cliente. De cada una de ellas se despliegan 7 flujos de trabajo: concepción inicial, captura de requisitos, diseño con metáforas, implementación, prueba, entrega de la documentación, soporte e investigación, utilizándose este último por el equipo de desarrollo cuando sea necesario, es decir, es un flujo que se puede mover y utilizarlo en cualquier parte del ciclo de vida del proyecto [32].

## CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

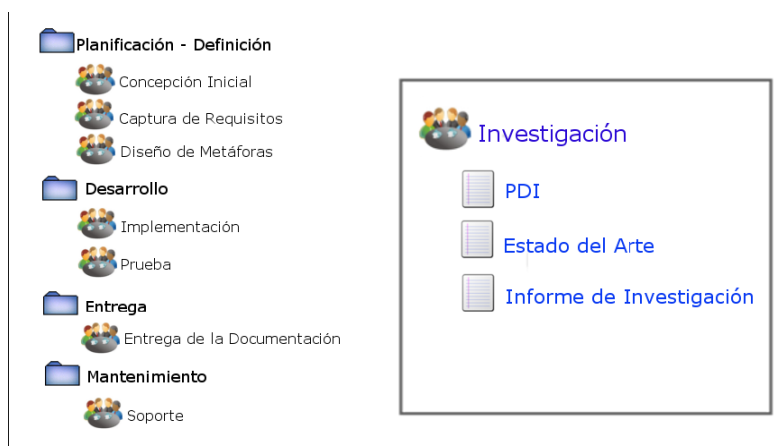


Figura 3: Fases y flujos de trabajo de SXP [27].

Los autores del presente trabajo elaboraron un esquema con estas fases, actividades y los artefactos que se generan en ellas, para lograr una mayor comprensión de la metodología a utilizar.

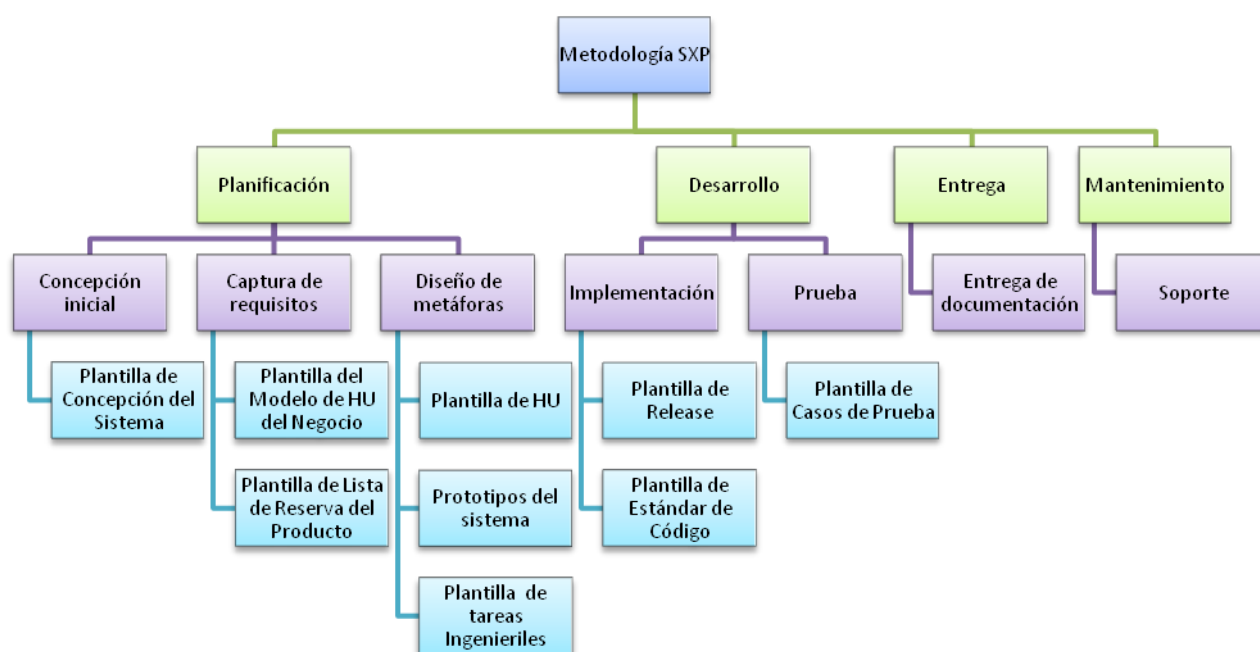


Figura 4: Esquema de la metodología SXP.

En la presente investigación se decidió usar esta personalización de las metodologías XP y SCRUM por las características que logran hacer, de un proceso de desarrollo, un proceso eficiente y eficaz. Ajustándose las mismas a las características propias del equipo de desarrollo entre las que se destacan principalmente: el equipo se encuentra formado por dos estudiantes y un líder, el mismo necesita reunirse diariamente para discutir avances y posibles cambios, es de vital importancia para el equipo ganar en tiempo para no retrasar la entrega.



## CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

### 1.3.2 Lenguaje de Modelado UML

Debido a la selección de la metodología de desarrollo de software se hace necesario utilizar como lenguaje unificado de modelado UML, por sus siglas en inglés, (Unified Modeling Language). UML es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para, especificar, construir y documentar un sistema. Especifica que el mismo brinda un estándar para describir un plano del software, incluye aspectos conceptuales tales como procesos de negocios, funciones del sistema, expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables [33] [34]. En concreto ayuda a comprender y a mantener de una mejor forma un sistema basado en un área que el analista o desarrollador puede desconocer.

La decisión de utilizar UML como notación para el desarrollo del software se debe a que se ha convertido en un estándar que tiene las siguientes características:

- Permite modelar sistemas haciendo uso de técnicas orientadas a objetos (OO).
- Permite especificar todas las decisiones de análisis y diseño, construyéndose así modelos precisos, no ambiguos y completos.
- Puede conectarse con lenguajes de programación (Ingeniería directa e inversa).
- Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, etc.).
- Es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas.
- Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.
- Es independiente del proceso, aunque para utilizarlo óptimamente se debería emplear en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.

### 1.3.3 Herramientas Case Visual Paradigm

Se tuvo en cuenta para la elección de la herramienta de modelo a usar, las características que posee Visual Paradigm y que presenta [35], entre las que se encuentran 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 diseñar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas, documentación, bases de datos, transforma diagramas de Entidad de Relación en tablas de una base de datos. También proporciona abundantes tutoriales de UML, demostraciones interactivas de modelado y proyectos

## CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

del mismo. Se puede adquirir mediante licencia gratuita y comercial. Es fácil de instalar y actualizar y es compatible entre ediciones. Permite la importación y exportación de ficheros XML [35]. Existe amplia experiencia en la universidad en el uso de esta herramienta CASE en los proyectos productivos.

### 1.3.4 Lenguaje de programación Java

**Java** es un lenguaje de Programación Orientado a Objetos (POO) el mismo se presenta bajo licencia GNU GPL, por tanto se puede considerar el lenguaje Java como de Software Libre. Actualmente 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. Este lenguaje permite aprovechar la flexibilidad de la POO en el diseño de sus aplicaciones. Con este lenguaje se puede acceder a bases de datos fácilmente con la conectividad de la Base de datos de Java (JDBC, Java Data Base Connectivity, según sus siglas en inglés), independientemente de la plataforma utilizada. El manejo de las bases de datos es uniforme, transparente y simple, ofrece una gran seguridad para desarrollar un sistema como el que se pretende en esta investigación. Después de analizar los elementos anteriores se considera que para el desarrollo de la aplicación se usará este lenguaje.

### 1.3.5 Marcos de trabajo

#### 1.3.5.1 Vaadin

Vaadin es un framework Java de código abierto para el desarrollo de aplicaciones web vistosas e interactivas sin requerir para ello de plugins en el servidor web ya que es un framework que trabaja en el lado del servidor. Funciona perfectamente con los siguientes navegadores: Mozilla Firefox 3+, Internet Explorer 6, 7, y 8, Safari 3+, Opera 10 y Google Chrome.

El desarrollo con Vaadin se centra en hacer aplicaciones solamente programando en Java; no es necesario manejar Javascript ni HTML para la mayoría de las cosas. Se pueden desarrollar componentes propios, basados en los componentes existentes de Vaadin, que van desde botones, etiquetas, ligas y campos de texto hasta formas completas, campos para captura de fecha, notificaciones de errores, árboles jerárquicos, etc.

La programación es similar a Swing, ya que el control de las acciones también se maneja por medio de eventos que son enviados a listeners. La plataforma Swing se basa primordialmente para el manejo de eventos en el concepto de escucha o listener; la teoría general establece que existe un objeto llamado disparador que realiza alguna acción sobre otro objeto reactivo. Esto simplifica un poco las cosas y

## CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

ofrece una buena abstracción al grado de que el desarrollador se puede olvidar de que está programando una aplicación web.

Servidores de aplicaciones admitidos:

- Apache Tomcat
- BEA WebLogic
- IBM WebSphere Application Server
- JBoss Application Server
- Jetty
- Glassfish

En el corazón de Vaadin existen, principalmente, tres tecnologías: AJAX para la interacción entre el usuario y la aplicación posibilitando la actualización de contenido mediante peticiones y respuestas asíncronas, Google Web Toolkit (GWT) aísla al programador de trabajar con JavaScript y otras tecnologías, JSON (JavaScript Object Notation) que permite crear y parsear información sobre los componentes para trabajar directamente con GWT.

La arquitectura de Vaadin se puede dividir en varias partes según su funcionalidad:

- Componentes de interfaz de usuario: que contiene todos los componentes Vaadin con los que el usuario puede interactuar.
- Motor del lado del cliente: administra el renderizado de componentes utilizando para ello el motor de GWT. Se comunica, además, con el lado del cliente (Terminal Adapter) mediante un lenguaje de definición de interfaz de usuario (UIDL), JSON y peticiones asíncronas de HTTP.
- Terminal Adapter: cuando el usuario hace algún cambio en su cliente web, sin darse cuenta, está comunicándose con el Terminal Adapter donde llegan esos cambios en forma de eventos. En este punto se comunican los cambios con la parte lógica de la aplicación y se responde al cliente con los cambios en la interfaz gráfica.
- Temas: la definición del aspecto de la interfaz de usuario está definida mediante CSS. Vaadin ofrece inicialmente una serie de esquemas CSS entre los que elegir.
- UIDL: es la definición de interfaz basada en JSON que permite el intercambio de información, siendo muy eficiente para la comunicación AJAX.
- Eventos: el usuario interactúa con los componentes de la interfaz generando eventos que se procesan inicialmente, en el lado del cliente, en JavaScript y posteriormente va pasando a través de las diferentes capas del lado del servidor: HTTP Server, Terminal Adapter, etc.
- Modelo de datos: el modelo de datos de Vaadin permite presentar la información en la interfaz de usuario y a su vez, que los componentes de la interfaz de usuario puedan actualizar los datos de modelo.

## CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

### 1.3.5.2 Spring

**Spring**, según Johnson [36] 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. El mismo interviene en todas las capas arquitectónicas de una aplicación Java. Teniendo en cuenta que su diseño brinda flexibilidad arquitectónica.

Areces [37] 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 este marco de trabajo 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, por las facilidades que brinda para el desarrollo de la solución.

### 1.3.5.4 Hibernate

**Hibernate**, según Relational Persistence for Java [38] 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 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*").

### 1.3.6 Gestor de Base de Datos PostgreSQL

PostgreSQL, [39], 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 da la posibilidad de que mientras un proceso es escrito en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases. Implementa el uso de retrocesos, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz. Posee la capacidad de

## CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos.

### 1.3.7 Entorno de Desarrollo Integrado (IDE) Eclipse

Eclipse es un IDE 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 "Cliente-liviano" que son basadas en navegadores [40]. La interfaz de usuario de Eclipse tiene una capa intermedia Interfaz Gráfica de Usuario, conocida también como GUI (del inglés Graphical User Interface) llamada JFace, la cual facilita la construcción de aplicaciones basadas en SWT (Standard Widget Toolkit) [41]. 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, las necesite el usuario o no. 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 [42].

### 1.3.8 Servidor de Aplicaciones Tomcat

**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. Al inicio de su desarrollo existía la idea de que su utilización era de manera autónoma, es decir sólo recomendable para entornos de desarrollo con mínimos requisitos de velocidad y gestión de transacciones, pero en la actualidad ya no es de esa forma, y es usado como servidor Web independiente en entornos con alto nivel de tráfico y alta disponibilidad. Es gratis, fácil de instalar, se ejecuta en máquinas más pequeñas y presenta compatibilidad con las API más recientes de Java. Permite la descarga, la instalación y la prueba en el iSeries en menos de una hora. 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. Presenta gran éxito como producto de código libre. El equipo de desarrollo decidió usar Tomcat porque además de estar escrito en Java, este satisface las necesidades para el desarrollo de la aplicación que se propone [43] [44].

### Conclusiones Parciales

- Los sistemas de soporte a la decisión permiten aumentar la capacidad de los decisores para tomar una decisión más acertada y con menor tiempo y costo.

## CAPÍTULO 1: MARCO TEÓRICO REFERENCIAL

- El AHP y el ANP son métodos que permiten solucionar problemas de decisión multicriterios que plantean un grupo de pasos matemáticos que se complejizan en la medida que sea mayor el número de criterios y de expertos.
- Para el desarrollo del sistema se empleará la metodología SXP y la herramienta CASE Visual Paradigm para la descripción y modelación del mismo.
- Como sistema gestor de base de datos se utilizará el PostgreSQL.
- La implementación del sistema se hará empleando IDE Eclipse STS 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 servidor de aplicaciones el Apache Tomcat.



Es de suma importancia para la construcción y desarrollo de un sistema informático la utilización de métodos y técnicas que permitan la solución de problemas existentes a lo largo del ciclo de vida de un software. La correcta captura de requisitos y el modelamiento del sistema permiten que se mitiguen las fallas que puedan aparecer durante su desarrollo, además de establecer un entendimiento común entre lo que se desea y lo que se elabora. La implementación de la solución propuesta se caracteriza por basarse en los principios y reglas de la metodología SXP, utilizando las tecnologías y herramientas definidas.

La metodología empleada para la descripción del sistema se divide en fases y actividades dentro de estas, generándose un grupo de artefactos que permiten especificar los elementos fundamentales del sistema. En el capítulo 1 se muestra el esquema de las fases, actividades y artefactos de la metodología, de ellos en este capítulo se describen los artefactos correspondientes a la fase de Planificación-Definición.

### **2.1. Descripción del problema**

La vida cotidiana esta llena de escenarios donde constantemente se deben tomar decisiones que involucran diferentes criterios de selección y alternativas a seleccionar, por lo general es complejo seleccionar la alternativa que satisfaga el mayor número de criterios definidos. Esta selección se torna mucho más compleja cuando se tienen numerosos criterios y alternativas. Existen varios métodos de toma de decisiones que ayudan a resolver los problemas de decisión, dentro de estos métodos se encuentran el AHP y ANP, los mismos se caracterizan por tener gran cantidad de cálculos matemáticos donde predominan las operaciones matriciales lo que provoca que su confección manual sea muy engorrosa y susceptible a errores, debido a esto existe una demora en los procesos de toma de decisiones.

### **2.2. Solución Propuesta**

Haciendo uso de sofisticadas herramientas y potentes marcos de trabajos se propone la implementación de un módulo que contenga los métodos AHP y ANP con el fin de agilizar el proceso de toma de decisiones. Este módulo cumplirá con una serie de requisitos funcionales y no funcionales que lograrán que la interacción de los decisores con el sistema sea sencilla y eficiente, garantizándose así que en un tiempo razonable todas las decisiones que se tomen sean correctas y eficaces.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

### **2.3 Concepción inicial del sistema**

En la etapa de concepción del sistema se genera la plantilla de concepción del sistema, en ella se especifican los aspectos generales organizativos y de concepción del sistema, su objetivo, principales involucrados y otros aspectos que permiten la posterior organización del desarrollo del proyecto. Esta plantilla se muestra en el anexo 1.

### **2.4 Captura de requisitos**

La ingeniería de requisitos es el conjunto de actividades implicadas en descubrir, documentar y mantener un conjunto de requisitos [32]. La captura de los mismos es un proceso en el cual los datos son extraídos de las personas pudiendo variar, dependiendo de la persona consultada.

Para la adquisición de estos requisitos se emplearon técnicas que permitieron hacer este proceso de forma más adecuada y segura. Entre las técnicas existentes para la captura de requerimientos se empleó la entrevista, específicamente la discusión, donde se sostuvo una discusión con el cliente sobre su problemática, para tratar de determinar en conjunto los requisitos del sistema, además de la arqueología de documentos donde se tratan de determinar posibles requerimientos sobre la base de inspeccionar la documentación utilizada por el cliente.

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, estos elementos 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 los desarrolladores y el cliente. Luego se sometieron a 2 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 se presenta un diagrama de historias de usuarios del negocio que permite ver la relación entre los usuarios y las actividades que se realizan, de ahí que se puedan obtener de ello los requisitos. Este diagrama se muestra a continuación:



## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA



Figura 5: Historias de Usuario del Negocio.

Los restantes elementos de esta plantilla se describen en el anexo 2.

### 2.4.2 Lista de Reserva del Producto (LRP)

En la lista de reserva, artefacto generado en la captura de requisitos, se describen los mismos como funcionalidades que el sistema debe cumplir en su desarrollo. Estos son descritos en la siguiente tabla:

Prioridad	Ítem	Descripción	Estimación en sprint	Estimado por	Asignado a
<b>Requisitos Funcionales Muy Alta</b>					
	1	Autenticación de usuarios	1	programador	Adrián Román González Landeiro
	2	Ponderar alternativas según los criterios	3	programador	Idel Jorge Sánchez González
	3	Ponderar los criterios según la importancia	3	programador	Adrián Román González Landeiro
	4	Ponderar los subcriterios	3	programador	Idel Jorge Sánchez González
	5	Elegir mejor alternativa según AHP	3	programador	Adrián Román González Landeiro
	6	Llenar matriz de dominación interfactorial	3	programador	Idel Jorge Sánchez González
	7	Ponderar los componentes según dominancia	3	programador	Idel Jorge Sánchez González
	8	Elegir mejor alternativa según ANP	3	programador	Adrián Román González Landeiro
<b>Alta</b>					
	9	Verificar consistencia de los datos según AHP	3	programador	Idel Jorge Sánchez González
	10	Obtener ranking de alternativas	3	programador	Adrián Román

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

	11	Verificar consistencia de los datos según ANP	3	programador	González Landeiro Adrián Román González Landeiro
<b>Media</b>					
	12	Gestionar decisión	1	programador	Idel Jorge Sánchez González
	13	Seleccionar decisión	3	programador	Adrián Román González Landeiro
	14	Gestionar alternativas	1	programador	Idel Jorge Sánchez González
	15	Gestionar criterios	1	programador	Adrián Román González Landeiro
	16	Gestionar sub criterios	1	programador	Idel Jorge Sánchez González
	17	Gestionar componentes	1	programador	Adrián Román González Landeiro
<b>Requisitos No Funcionales</b>					
<b>Usabilidad</b>					
	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	Idel Jorge Sánchez González
	2	El tiempo de entrenamiento requerido para que usuarios normales y avanzados sean productivos operando el sistema es de 2 días.		Programador	Adrián Román González Landeiro
	3	Debe poseer una interfaz agradable para el cliente.		Diseñador	Idel Jorge Sánchez González
<b>Fiabilidad</b>					
	4	El sistema estará disponible 24 horas del día, los siete días de la semana.		Programador	Adrián Román González Landeiro
	5	La herramienta de implementación a utilizar tiene soporte para recuperación ante fallos y errores.		Programador	Idel Jorge Sánchez González
<b>Seguridad</b>					
	6	Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.		Programador	Idel Jorge Sánchez González
	7	El sistema debe mantener en todo momento la seguridad de la información asegurando la autenticidad de la misma.		Programador	Adrián Román González Landeiro
	8	El sistema debe garantizar la confidencialidad, integridad y disponibilidad de la información que se procese en el sistema.		Programador	Idel Jorge Sánchez González
	9	El control de acceso se establecerá por roles que se le asignarán a los usuarios que interactúen con el sistema.		Programador	Adrián Román González Landeiro
<b>Soporte</b>					
	10	Soporte para grandes volúmenes de datos y velocidad de procesamiento.		Programador	Idel Jorge Sánchez González
	11	Tiempo de respuesta rápido en accesos concurrentes.		Programador	Adrián Román González Landeiro
<b>Restricciones de diseño</b>					
	12	El lenguaje de programación es Java		Programador	Idel Jorge Sánchez González
	13	Los marcos de trabajo de desarrollo son Vaadin, Spring, Hibernate.		Programador	Adrián Román González Landeiro
	14	La herramienta IDE de desarrollo utilizada será Eclipse STS.		Programador	Idel Jorge Sánchez González
	15	La herramienta case utilizada es Visual Paradigm para modelado.		Programador	Adrián Román González Landeiro
	16	La herramienta gestor de base de datos es el PostgreSQL.		Programador	Adrián Román González Landeiro

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Interfaz				
17	El sistema tiene que ofrecer una interfaz amigable, fácil de operar.	Diseñador	Idel Jorge Sánchez González	
18	Diseño sencillo, con pocas entradas, permitiendo que no sea necesario mucho entrenamiento para que los usuarios puedan utilizar el sistema.	Diseñador	Idel Jorge Sánchez González	

Tabla 1: Lista de Reserva del Producto

### 2.5 Diseño de metáforas

A partir de la definición de las funcionalidades descritas en la 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. Estas historias de usuario se especifican en el anexo 3. En el documento se presentan cuatro de ellas para dar continuidad en la explicación de las tareas ingenieriles definidas.

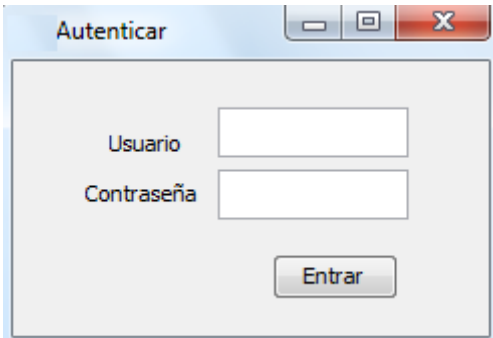
Historia de Usuario	
<b>Número:</b> HU_1	<b>Nombre Historia de Usuario:</b> Autenticación de usuarios
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Adrián Román González Landeiro	<b>Iteración Asignada:</b> Sprint 2
<b>Prioridad en Negocio:</b> <i>Muy Alta</i>	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 1
<b>Descripción:</b> <i>El usuario accede al sistema insertando un nombre de usuario y una contraseña, el sistema comprobará que estos datos sean correctos, al serlo, el sistema le dará al usuario acceso a las funcionalidades a las cuales tiene permisos.</i>	
<b>Observaciones:</b> <i>El usuario o decisor debe introducir un usuario y una contraseña al sistema que deben ser correctos, de otra manera no tendrá acceso al mismo.</i>	
<b>Prototipo de interfaces:</b>	
	

Tabla 2: HU Autenticación de usuarios

Historia de Usuario	
<b>Número:</b> HU_2	<b>Nombre Historia de Usuario:</b> Ponderar alternativas según los criterios
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Idel Jorge Sánchez González	<b>Iteración Asignada:</b> Sprint 2
<b>Prioridad en Negocio:</b> <i>Muy Alta</i>	<b>Puntos Estimados:</b> 3
<b>Riesgo en Desarrollo:</b> Alto	<b>Puntos Reales:</b> 3
<b>Descripción:</b> <i>Después que el usuario ha añadido las alternativas y los criterios que va a tener en la decisión, pondera las alternativas en base a los criterios de acuerdo a la escala definida por el método. Esa ponderación será guardada por el sistema para su posterior utilización.</i>	
<b>Observaciones:</b>	

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Prototipo de interfaces:

	Criterio 1	Criterio 2	Criterio 3	Criterio n
Alternativa 1	1	1	1	1
Alternativa 2	1	1	1	1
Alternativa 3	1	1	1	1
Alternativa n	1	1	1	1

Enviar

Tabla 3: HU Ponderar alternativas según los criterios

### Historia de Usuario

**Número:** HU\_3 **Nombre Historia de Usuario:** Ponderar los criterios según la importancia

**Modificación de Historia de Usuario Número:** Ninguna

**Usuario:** Adrián Román González Landeiro

**Iteración Asignada:** Sprint 2

**Prioridad en Negocio:** Muy Alta

**Puntos Estimados:** 3

**Riesgo en Desarrollo:** Alto

**Puntos Reales:** 3

**Descripción:** El usuario entra la ponderación de los criterios entrados al sistema pertenecientes a la decisión, el sistema almacena los pesos y los utiliza en los pasos posteriores del método

**Observaciones:**

Prototipo de interfaces:

	Criterio 1	Criterio 2	Criterio 3	Criterio n
Pesos	1	1	1	1

Enviar

Tabla 4: Historia de Usuario Ponderar los criterios según la importancia.

### Historia de Usuario

**Número:** HU\_10 **Nombre Historia de Usuario:** Obtener ranking de alternativas.

**Modificación de Historia de Usuario Número:** Ninguna

**Usuario:** Adrián Román González Landeiro

**Iteración Asignada:** Sprint 3

**Prioridad en Negocio:** Alta

**Puntos Estimados:** 1

**Riesgo en Desarrollo:** Medio

**Puntos Reales:** 1

**Descripción:** Luego de entrar las alternativas, los criterios y ponderar a cada uno de ellos el usuario puede ver el ranking de alternativas que se generó según el método seleccionado.

**Observaciones:**

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Prototipo de interfaces:

Nro.	Descripción	Peso
1	B	60.3
2	C	33.8
3	A	5.9

Tabla 5: HU Obtener ranking de alternativas

### 2.5.2 Tareas Ingenieriles

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

Algunas de las tareas ingenieriles están relacionadas con todas las historias de usuario preparando el ambiente de implementación, estas son:

Tarea de Ingeniería	
<b>Número Tarea:</b> T_1	<b>Número Historia de Usuario:</b> todas
<b>Nombre Tarea:</b> Generación de la base de datos	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 15-enero-2013	<b>Fecha Fin:</b> 20-enero-2013
<b>Programador Responsable:</b> Adrián Román González Landeiro	
<b>Descripción:</b> <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 6: Tarea Ingenieril 1

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

Tabla 7: Tarea Ingenieril 2

Tarea de Ingeniería	
<b>Número Tarea:</b> T_3	<b>Número Historia de Usuario:</b> todas
<b>Nombre Tarea:</b> Realizar el mapeo de la base de datos en el marco de trabajo Hibernate.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 5-febrero-2013	<b>Fecha Fin:</b> 20-febrero-2013
<b>Programador Responsable:</b> Idel Jorge Sánchez González	
<b>Descripción:</b> <i>Realizar el mapeo de la base de datos en el framework de acceso a datos seleccionado.</i>	

Tabla 8: Tarea Ingenieril 3

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Otras 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\_6 y la HU\_10 son:

Tarea de Ingeniería	
Número Tarea: T_13	Número Historia de Usuario: HU_6
Nombre Tarea: Diseñar las interfaces requeridas para la funcionalidad llenar matriz de dominación interfactorial.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 11-abril-2013	Fecha Fin: 15-abril 2013
Programador Responsable: Idel Jorge Sánchez González	
Descripción: <i>Se hace un diseño de las interfaces que se van a utilizar para llenar la matriz, las cuales deben cumplir con las necesidades del método, para hacer un buen uso de él.</i>	

Tabla 9: Tarea de Ingeniería 13.

Tarea de Ingeniería	
Número Tarea: T_14	Número Historia de Usuario: HU_6
Nombre Tarea: Implementar funcionalidad llenar matriz de dominación interfactorial.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 16-abril-2013	Fecha Fin: 25-abril 2013
Programador Responsable: Idel Jorge Sánchez González	
Descripción: <i>Se implementa la funcionalidad de llenar la matriz, los pesos entrados por los usuarios son guardados y con ellos se procede a realizar los demás pasos del método.</i>	

Tabla 10: Tarea de Ingeniería 14.

Tarea de Ingeniería	
Número Tarea: T_18	Número Historia de Usuario: HU_10
Nombre Tarea: Diseñar las interfaces requeridas para la funcionalidad obtener ranking de alternativas.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 5-mayo -2013	Fecha Fin: 10-mayo 2013
Programador Responsable: Adrián Román González Landeiro	
Descripción: <i>Se diseñan las interfaces necesarias para la implementación de las funcionalidades relacionadas con la HU, teniendo en cuenta los prototipos diseñados en las HU.</i>	

Tabla 11: Tarea de Ingeniería 18.

Tarea de Ingeniería	
Número Tarea: T_19	Número Historia de Usuario: HU_10
Nombre Tarea: Implementar funcionalidad obtener ranking de alternativas.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio: 10-mayo -2013	Fecha Fin: 15-mayo 2013
Programador Responsable: Adrián Román González Landeiro	
Descripción: <i>Se implementa la funcionalidad vinculada con la HU.</i>	

Tabla 12: Tarea de Ingeniería 19.

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

## 2.6 Arquitectura seleccionada

### Patrón en Capas

Los sistemas o arquitecturas en capas constituyen uno de los patrones que aparecen con mayor frecuencia. El patrón en capas es definido como una organización jerárquica tal que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior [45].

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

En algunos ejemplares, las capas internas están ocultas a todas las demás, menos para las capas externas adyacentes, y excepto para funciones puntuales de exportación; en estos sistemas, los componentes implementan máquinas virtuales en alguna de las capas de la jerarquía. En otros sistemas, las capas pueden ser sólo parcialmente opacas. En la práctica, las capas suelen ser entidades complejas, compuestas de varios paquetes o subsistemas. El uso de arquitecturas en capas, explícitas o implícitas, es muy frecuente.

En un patrón en capas, los conectores se definen mediante los protocolos que determinan las formas de la interacción. Los diagramas de sistemas clásicos dibujan las capas en adyacencia, sin conectores, flechas ni interfaces; en algunos casos se suele representar la naturaleza jerárquica del sistema en forma de círculos concéntricos. Las restricciones topológicas del patrón pueden incluir una limitación, más o menos rigurosa, que exige a cada capa operar sólo con capas adyacentes, y a los elementos de una capa entenderse sólo con otros elementos de la misma; se supone que si esta exigencia se relaja, el patrón deja de ser puro y pierde algo de su capacidad heurística [46].

A veces se argumenta que el cruce recargado de muchos niveles involucra eventuales degradaciones de rendimiento; pero muchas más veces se sacrifica la pureza de la arquitectura en capas precisamente para mejorarla: colocando, por ejemplo, reglas de negocios en los procedimientos almacenados de las bases de datos, o articulando instrucciones de consulta en la capa de la interfaz de usuario.

El número mínimo de capas es obviamente dos, en ese umbral la literatura arquitectónica sitúa a veces al sub-patrón cliente-servidor como el modelo arquetípico del patrón de capas y el que se encuentra con mayor frecuencia en las aplicaciones en red. Un componente servidor, que ofrece ciertos servicios, escucha que algún otro componente requiera uno; un componente cliente solicita ese servicio al servidor a través de un conector.

El servidor ejecuta el requerimiento (o lo rechaza) y le devuelve una respuesta [46].

Las ventajas del patrón en capas son:

- Soporta un diseño basado en niveles de abstracción crecientes, lo cual a su vez permite a los implementadores la partición de un problema complejo en una secuencia de pasos incrementales.
- Admite muy naturalmente optimizaciones y refinamientos.
- Proporciona amplia reutilización. Al igual que los tipos de datos abstractos, se pueden utilizar diferentes implementaciones o versiones de una misma capa en la medida que soporten las mismas interfaces de cara a las capas adyacentes. Esto conduce a la posibilidad de definir interfaces de capa estándar, a partir de las cuales se pueden construir extensiones o prestaciones específicas [46].

### 2.7 Patrones de diseño

Para la definición de las clases del sistema, el diseño del sistema a implementar, es importante la revisión de algunos patrones que permiten realizar un diseño adecuado y consistente. La asignación de responsabilidades es la habilidad más importante en el análisis y diseño orientado por objetos, para ello tiene suma importancia la utilización de los patrones GRASP.

En términos generales, un patrón es un conjunto de información que proporciona respuesta a un conjunto de problemas similares, es decir, un patrón es una solución a un problema en un contexto [47].

#### 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.”* Estos patrones se describen a continuación [37]:

**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 [37].

**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 [37].

**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 [36].

**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 [37].

**Controlador:** Es un evento generado por actores externos. Se asocian con operaciones del sistema, operaciones del sistema como respuestas a los eventos del sistema, tal como se relacionan los mensajes y los métodos [37]. Este patrón se pone de manifiesto con la utilización de una clase que coordina o controla las actividades que son necesarias realizar con las demás clases, en este caso AHP\_Controller y ANP\_Controller que son quienes hacen uso de este patrón.



## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Otros patrones utilizados:

**Patrón Fachada:** 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 [48].

**Patrón DAO:** El problema que viene a resolver este patrón es el de contar con diversas fuentes de datos (base de datos, archivos, servicios externos, etc.). De tal forma que se encapsula la forma de acceder a la fuente de datos. Este patrón surge históricamente de la necesidad de gestionar una diversidad de fuentes de datos, aunque su uso se extiende al problema de encapsular no sólo la fuente de datos, sino además ocultar la forma de acceder a los datos. Se trata de que el software cliente se centre en los datos que necesita y se olvide de cómo se realiza el acceso a los datos o de cuál es la fuente de almacenamiento [49].

En la descripción del diagrama de clases se especifica cómo se materializa el uso de estos patrones en el diseño realizado.

### 2.8 Diagrama de Clases

El propósito de este diagrama es el de representar los objetos fundamentales del sistema, es decir, los que percibe el usuario y con los que espera tratar para completar su tarea en vez de objetos del sistema o de un modelo de programación [50].

El diagrama de clases del diseño representa los métodos y atributos de cada una de las clases del sistema, para mostrar de forma simple la colaboración y las tareas de cada una de ellas en relación al sistema que conforman.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

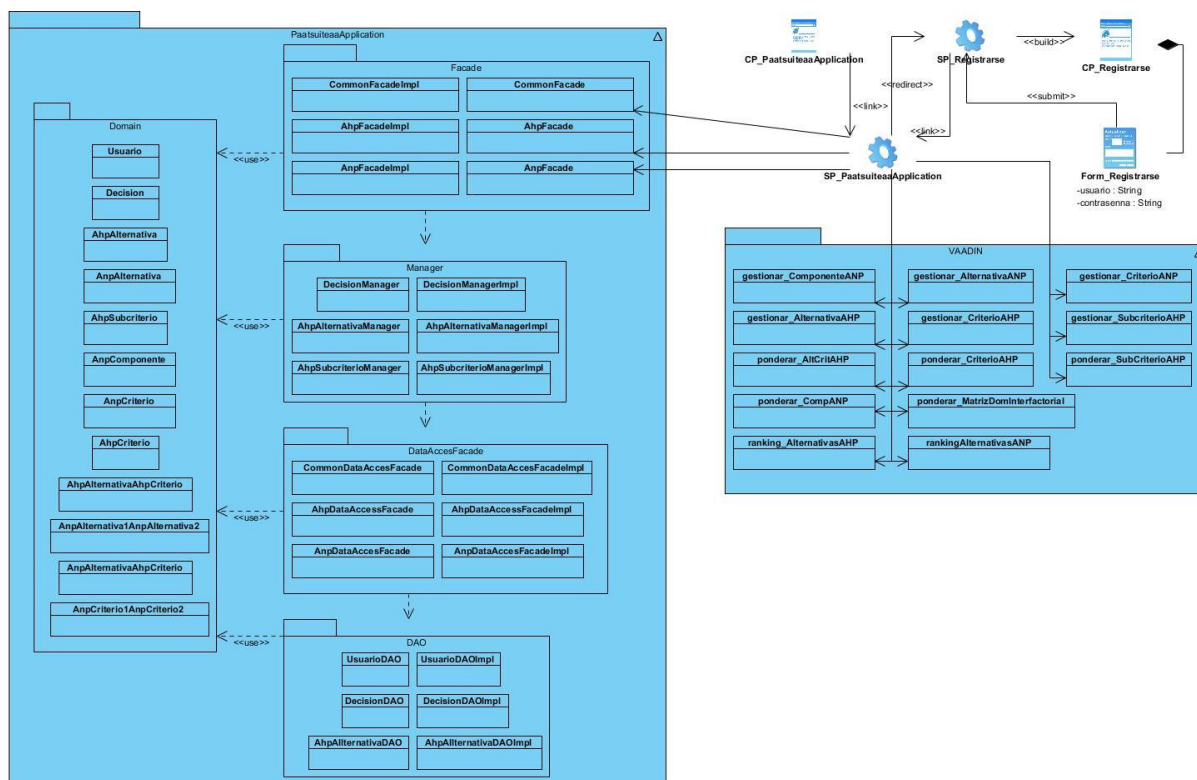


Figura 6: Diagrama de Clases.

En el diagrama de clases se pone de manifiesto el uso del patrón arquitectónico en capas. La clase PaatsuiteaaApplication es la clase principal de nuestro sistema, a través de la cual se va a acceder a todas las funcionalidades. Lo primero que hace el sistema al acceder a él es redireccionar para la clase interfaz Registrarse, en la cual el usuario debe introducir su usuario y contraseña para poder acceder al mismo. Una vez dentro el usuario puede acceder a cualquier clase interfaz, cada una está respondiendo a los requerimientos funcionales del sistema, estas clases están en el paquete VAADIN. Al seleccionar una de las clases interfaz para realizar una operación, se debe acceder a las clases del negocio, que es donde está la lógica de nuestra aplicación, para esto primero se debe acceder a las clases del paquete Facade, estas se necesitan para implementar el patrón fachada el cual se utilizó para brindar una interfaz de comunicación entre varias de ellas que se utilizan en el sistema. Mediante las clases que define este patrón se accede a las del negocio que están en el paquete Manager, donde van a estar todas las clases correspondientes al negocio de nuestro sistema. Haciendo uso de otras clases del patrón fachada que se encuentran en el paquete DataAccesFacade, se puede acudir a las clases de acceso a dato ubicadas en el paquete DAO, ahí se van a encontrar todas las encargadas de acceder a la base de datos del sistema. El patrón DAO se empleó para crear el conjunto de clases que permiten el manejo de los datos persistentes independizando esta gestión del sistema gestor de base de datos que se utilice. Los paquetes Facade, Manager, DataAccesFacade y DAO usan las clases del dominio que se encuentran en el paquete Domain.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Los patrones GRASP utilizados se ponen de manifiesto en el diagrama de clases, asociado a las relaciones entre estas y las responsabilidades asociadas a cada una de ellas para dar cumplimiento a los requisitos. En el caso del patrón creador es utilizado para dar la responsabilidad de la creación de los objetos a las clases que corresponde, el patrón experto se utilizó para asignar las responsabilidades relacionadas con las principales funcionalidades de la aplicación. Por ejemplo para asignar la responsabilidad de la funcionalidad de adicionar una decisión se designó la clase `CommonDataAccesFacade`, esta nos remite a las clases de acceso a datos, para adicionar la decisión a la base de datos de nuestro sistema. Relacionado con el diagrama de clases también está el patrón alta cohesión, este posibilita que las clases tengan solo las responsabilidades que le corresponden de acuerdo al objetivo e información que maneja.

Para analizar el uso del patrón bajo acoplamiento se vincula con otro patrón empleado, el patrón Fachada, haciendo que las relaciones entre las clases sean optimizadas en función de hacer que el sistema sea menos resistente a los cambios que se puedan producir. En este caso las clases de las fachadas `CommonFacade`, `AhpFacade` y `AnpFacade`.

Para más entendimiento de cómo funciona la lógica de la aplicación se realizaron algunos diagramas de clases con estereotipos web de varios requisitos, estos diagramas se pueden apreciar en el anexo 6.

### 2.9 Modelo de Datos

Es un conjunto de conceptos que nos permiten describir los datos, las relaciones entre ellos, la semántica y las restricciones de consistencia [51].

Existen 3 tipos de modelos de datos [51]:

- **Modelos externos o lógicos basados en objetos:** permiten representar los datos que necesita cada usuario con las estructuras propias del lenguaje de programación que se vaya a usar.
- **Modelos globales o lógicos basados en registros:** ayuda a escribir los datos para el conjunto de usuarios.
- **Modelos físicos o de datos:** orientado a la máquina.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

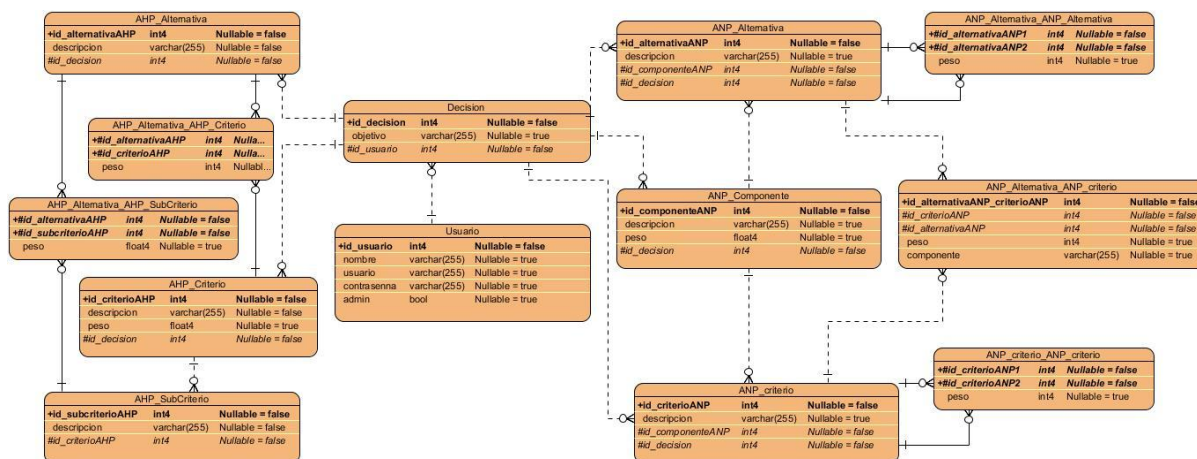


Figura 7: Modelo de datos

Se describe el modelo de datos en el anexo 5, cada una de las tablas y sus atributos, algunos ejemplos de la descripción de estos modelos son los siguientes:

Nombre Tabla: usuario		
<b>Descripción:</b> Esta es la tabla donde se va a almacenar toda la información de los usuarios que pueden acceder al sistema.		
Atributos	Tipo	Descripción
id_usuario	Int	Es el identificador del usuario. Es la llave primaria de la tabla.
nombre	Varchar	Es el nombre del usuario.
usuario	Varchar	Este es el usuario con que va a acceder al sistema.
contraseña	Varchar	La contraseña con que va a acceder al sistema.
admin	Bool	Este es el atributo que define si el usuario es administrador o no.

Tabla 13: Tabla usuario.

Nombre Tabla: Decisión		
<b>Descripción:</b> Es la tabla donde se describe cada decisión que se va a tomar.		
Atributos	Tipo	Descripción
Id_decisión	Int	Es el identificador de la decisión. Es la llave primaria de la tabla.
objetivo	Varchar	Se especifica cuál es el objetivo de la decisión que se quiere tomar.
id_usuario	Integer	Pertenece al identificador del usuario que va a tomar la decisión

Tabla 14: Tabla Decisión.

Nombre Tabla: AHP_Alternativa		
<b>Descripción:</b> Es la tabla donde se describen las alternativas en el caso en que se va a utilizar el método AHP.		
Atributos	Tipo	Descripción
Id_alternativaAHP	Int	Es el identificador de la alternativa cuando se usa el método AHP. Es la llave primaria de la tabla.
descripción	Varchar	Es la descripción de la alternativa.
Id_decisión	Int	Es el identificador de la decisión. Es una llave foránea en la tabla.

Tabla 15: Tabla AHP\_Alternativa.

Nombre Tabla: AHP_Criterio		
<b>Descripción:</b> Es la tabla donde se describen los criterios en el caso en que se va a utilizar el método AHP.		
Atributos	Tipo	Descripción
Id_criterioAHP	Int	Es el identificador del criterio cuando se usa el método AHP. Es la llave primaria de la tabla.
descripción	Varchar	Es la descripción del criterio.
Id_decisión	Int	Es el identificador de la decisión. Es una llave foránea en la tabla.

Tabla 16: Tabla AHP\_Criterio.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

### **Conclusiones Parciales**

- Al 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 de desarrollo.
- Mediante las historias de usuarios, se sentaron las bases para las restantes fases del proceso.
- Con el diseño de un Modelo de Datos se representaron las entidades relevantes del sistema y por medio del Diagrama de Clases se visualizaron las relaciones entre las clases que involucran dicho sistema, permitiendo una mejor comprensión del mismo.
- Estos resultados son utilizados como entrada en la siguiente etapa de trabajo.

Con la realización de este capítulo se describe cómo fue implementada la aplicación en términos de componentes. Se detalla mediante el diagrama de despliegue cómo quedará distribuida la aplicación. Se presenta la validación de las pruebas de calidad desarrolladas. Al final de este capítulo se realizará un análisis de la solución obtenida.

### 3.1 Implementación

#### 3.1.1. Diagrama de componentes

Un diagrama de componentes muestra las dependencias lógicas entre componentes software, sean estos últimos fuentes, binarios o ejecutables, los cuales ilustran las piezas del software. Los diagramas de Componentes prevalecen en el campo de la arquitectura de software pero pueden ser usados para modelar y documentar cualquier arquitectura del sistema, es decir para describir la vista de implementación estática de un sistema. Los diagramas de componentes se relacionan con los diagramas de clases, ya que un componente normalmente se corresponde con una o más clases, interfaces o colaboraciones, pero un diagrama de Componentes tiene un nivel más alto de abstracción que un diagrama de clase. Usualmente un componente se implementa por una o varias clases (u objetos) en tiempo de ejecución. Estos son bloques de construcción, como eventualmente un componente puede comprender una gran porción de un sistema [52].

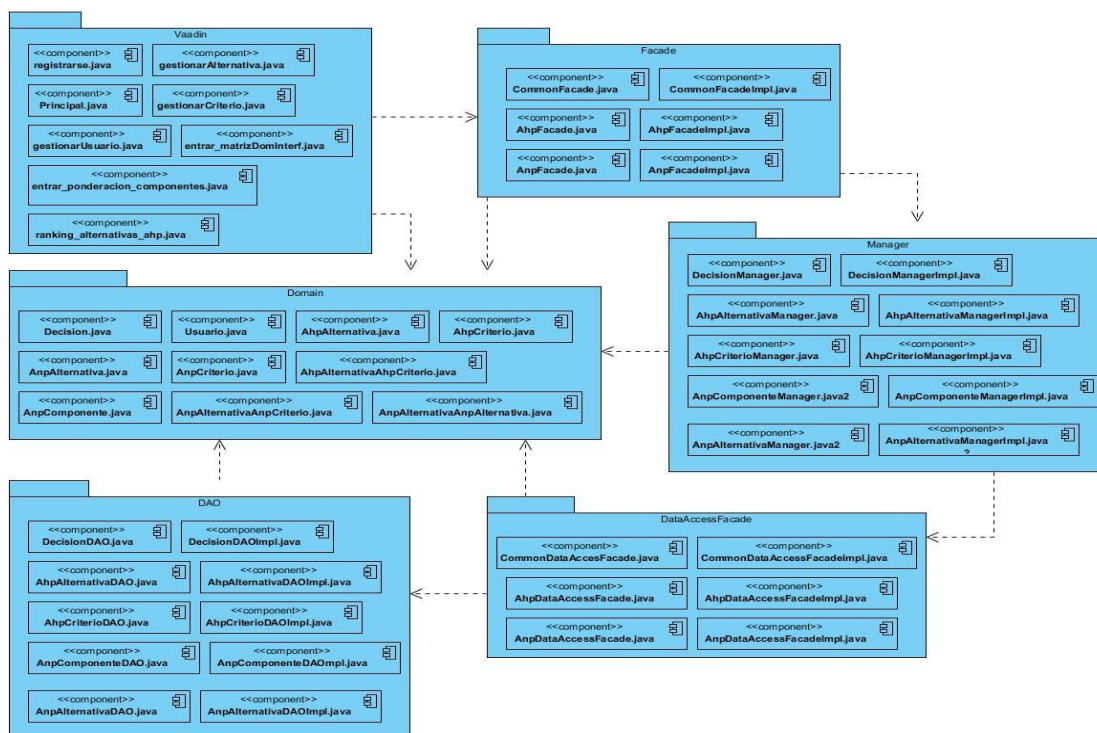


Figura 8: Diagrama de Componentes.

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

### 3.1.2. Diagrama de Despliegue

Para comprender como se ejecutará a nivel de hardware un sistema desarrollado y tener una visión clara de la estructura del sistema en ejecución y las relaciones entre los componentes que interactúan en el mismo, se realiza el diagrama de despliegue. Dicho diagrama tiene como objetivo reflejar lo mencionado anteriormente, representando la disposición de las instancias de los componentes de ejecución, en instancias de nodos conectados por enlaces de comunicación.

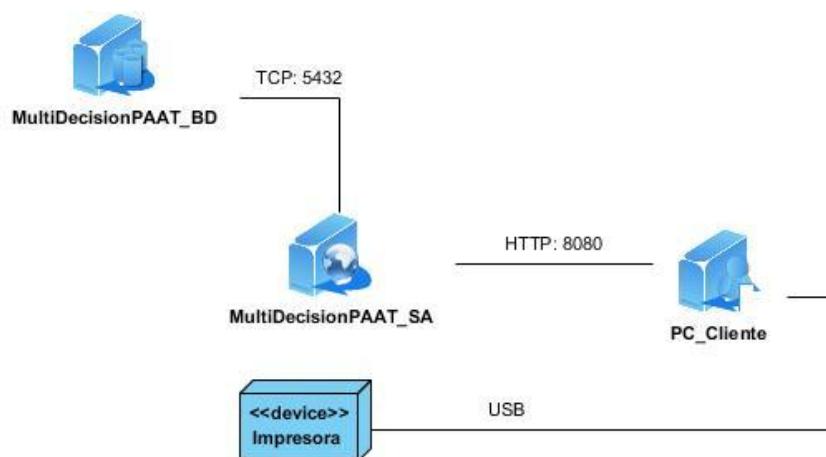


Figura 9: Diagrama de Despliegue.

Para el despliegue del sistema a implementar se tiene que la PC\_Cliente se encuentra conectada por USB a una impresora donde se podrá obtener de forma física los resultados y por HTTP a través del puerto 8080 al servidor de aplicación. Este último mediante el protocolo TCP y puerto 5432 se conectará a la base de datos del sistema.

### 3.2 Diseño de los casos de prueba

Las pruebas que se le realizan al sistema para la validación de sus funcionalidades son pruebas de caja negra, estas pruebas se llevan a cabo sobre la interfaz del software, obviando el comportamiento interno y la estructura del sistema [53].

Los casos de prueba de caja negra pretenden demostrar que [53]:

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

A continuación se derivan conjuntos de condiciones de entrada que utilizan todos los requisitos funcionales de una aplicación.

Las pruebas de caja negra pretenden encontrar errores como [53]:

- Funciones incorrectas o ausentes.

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

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

Algunos ejemplos de casos de prueba, para la aplicación, se muestran seguidamente:

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> M-1-1	<b>Nombre Historia de Usuario:</b> Autenticación de usuarios
<b>Nombre de la persona que realiza la prueba:</b> Adrián Román González Landeiro	
<b>Descripción de la Prueba:</b> Se ejecuta la prueba y se verifica que el usuario se autentique correctamente.	
<b>Condiciones de Ejecución:</b> <ul style="list-style-type: none"><li>• Que se inserte correctamente el usuario y la contraseña.</li></ul>	
<b>Entrada / Pasos de ejecución:</b> Se inserta un usuario y una contraseña y se verifican que sean los correctos.	
<b>Resultado Esperado:</b> Que el usuario acceda al sistema y se activen las funcionalidades correspondientes.	
<b>Evaluación de la Prueba:</b> Satisfactoria	

Tabla 17: Caso de Prueba de Autenticación de usuarios

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> M-2-2	<b>Nombre Historia de Usuario:</b> Ponderar alternativas según los criterios
<b>Nombre de la persona que realiza la prueba:</b> Idel Jorge Sánchez González	
<b>Descripción de la Prueba:</b> Después que el usuario ha añadido las alternativas y los criterios que va a tener en la decisión, pondera las alternativas en base a los criterios de acuerdo a la escala definida por el método. Esa ponderación será guardada por el sistema para su posterior utilización.	
<b>Condiciones de Ejecución:</b> <ul style="list-style-type: none"><li>• Que se ponderen las alternativas en base a los criterios de acuerdo a la escala definida por el método.</li></ul>	
<b>Entrada / Pasos de ejecución:</b> Se ponderan las alternativas en base a los criterios y se verifica que la ponderación sea guardada por el sistema para su posterior utilización.	
<b>Resultado Esperado:</b> Que se ponderen las alternativas en base a los criterios para dar continuidad al proceso de toma de decisión.	
<b>Evaluación de la Prueba:</b> Satisfactoria	

Tabla 18: Caso de Prueba de Ponderar alternativas según los criterios

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> M-8-8	<b>Nombre Historia de Usuario:</b> Elegir mejor alternativa según ANP
<b>Nombre de la persona que realiza la prueba:</b> Adrián Román González Landeiro	
<b>Descripción de la Prueba:</b> Después de entradas cada uno de los elementos, ponderamos cada uno de ellos, el decisor puede ver mediante la funcionalidad, cuál es la mejor o las mejores alternativas según el método.	
<b>Condiciones de Ejecución:</b> <ul style="list-style-type: none"><li>• Que se haya realizado correctamente la entrada de los componentes, las alternativas, los criterios y que se hayan ponderado correctamente cada uno de ellos.</li></ul>	
<b>Entrada / Pasos de ejecución:</b> Se entran cada uno de los elementos, se ejecuta la funcionalidad y finalmente se obtiene el resultado deseado.	
<b>Resultado Esperado:</b> Que se obtenga la mejor o las mejores alternativas según el método ANP.	
<b>Evaluación de la Prueba:</b> Satisfactoria	

Tabla 19: Caso de Prueba de Elegir mejor alternativa según ANP

El resto de los casos de pruebas definidos para la aplicación se encuentran en la planilla del anexo 7.

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

Las métricas de software son una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado. Permiten averiguar cuán bien están definidas las clases y el sistema,



## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

lo cual tiene un impacto directo en el mantenimiento del mismo, tanto por la comprensión de lo desarrollado como por la dificultad de modificarlo con éxito. Estas métricas tienen como propósito entender y mejorar la calidad del producto, evaluar la efectividad del proceso y mejorar la calidad del trabajo llevado a cabo a nivel del proyecto [54].

Para la evaluación de la calidad del diseño propuesto se hará uso de las métricas Tamaño Operacional de Clase (TOC) [54] [55] y Relaciones entre Clases (RC) [54] [55].

TOC y RC permiten medir los siguientes atributos de calidad:

Las métricas empleadas están diseñadas para evaluar los siguientes atributos de calidad [56]:

- **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 implementado 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 (Unidad) del producto (componente, módulo, clase, conjunto de clases, etc.) diseñado.

### 3.3.1 Métrica TOC

El TOC está dado por el número de métodos asignados a una clase y evalúa los siguientes atributos de calidad [57]:

- **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 del grado de reutilización de la clase.

Atributo de Calidad	Categoría	Criterio
Responsabilidad	Baja	< =Promedio.
	Media	Entre Promedio y 2* Promedio

### CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

	Alta	> 2* Promedio
<b>Complejidad de implementación</b>	Baja	< =Promedio
	Media	Entre Promedio y 2* Promedio
	Alta	> 2* Promedio
<b>Reutilización</b>	Baja	> 2*Promedio
	Media	Entre Promedio y 2* Promedio
	Alta	<= Promedio.

Tabla 20: Rango de valores para la evaluación técnica de los atributos de calidad relacionados con la métrica TOC [58].

A continuación se muestran las clases del sistema evaluadas en los atributos de calidad Responsabilidad, Complejidad de implementación y Reutilización.

Clase	Cantidad de Procedimientos	Responsabilidad	Complejidad	Reutilización
<b>CommonDataAccesFacade</b>	6	Media	Media	Media
<b>DecisionDAO</b>	2	Baja	Baja	Alta
<b>UsuarioDAO</b>	2	Baja	Baja	Alta
<b>CommonDataAccesFacadeImpl</b>	5	Baja	Baja	Alta
<b>DecisionDAOImpl</b>	2	Baja	Baja	Alta
<b>UsuarioDAOImpl</b>	2	Baja	Baja	Alta
<b>CommonFacade</b>	6	Media	Media	Media
<b>CommonFacadeImpl</b>	5	Baja	Baja	Alta
<b>DecisionManager</b>	4	Baja	Baja	Alta
<b>UsuarioManager</b>	2	Baja	Baja	Alta
<b>DecisionManagerImpl</b>	2	Baja	Baja	Alta
<b>UsuarioManagerImpl</b>	2	Baja	Baja	Alta
<b>AhpAlternativaAhpCriterioDAO</b>	2	Baja	Baja	Alta
<b>AhpAlternativaAhpSubcriterioDAO</b>	2	Baja	Baja	Alta
<b>AhpAlternativaDAO</b>	2	Baja	Baja	Alta
<b>AhpCriterioDAO</b>	2	Baja	Baja	Alta
<b>AhpDataAccesFacade</b>	13	Alta	Alta	Baja
<b>AhpSubcriterioDAO</b>	1	Baja	Baja	Alta
<b>AhpAlternativaAhpCriterioDAOImpl</b>	2	Baja	Baja	Alta
<b>AhpAlternativaAhpSubcriterioDAOImpl</b>	2	Baja	Baja	Alta
<b>AhpAlternativaDAOImpl</b>	2	Baja	Baja	Alta
<b>AhpCriterioDAOImpl</b>	2	Baja	Baja	Alta
<b>AhpDataAccesFacadeImpl</b>	12	Alta	Alta	Baja
<b>AhpSubcriterioDAOImpl</b>	1	Baja	Baja	Alta
<b>AhpFacade</b>	13	Alta	Alta	Baja
<b>AhpFacadeImpl</b>	11	Alta	Alta	Baja
<b>AHP_Controller</b>	43	Alta	Alta	Baja
<b>AHP_MatrizGenerada</b>	7	Media	Media	Media
<b>AhpAlternativaAhpCriterioManager</b>	2	Baja	Baja	Alta
<b>AhpAlternativaAhpSubcriterioManager</b>	2	Baja	Baja	Alta
<b>AhpAlternativaManager</b>	4	Baja	Baja	Alta
<b>AhpCriterioManager</b>	4	Baja	Baja	Alta
<b>AhpSubcriterioManager</b>	3	Baja	Baja	Alta
<b>AhpAlternativaAhpCriterioManagerImpl</b>	2	Baja	Baja	Alta
<b>AhpAlternativaAhpSubcriterioManagerImpl</b>	2	Baja	Baja	Alta
<b>AhpAlternativaManagerImpl</b>	3	Baja	Baja	Alta
<b>AhpCriterioManagerImpl</b>	3	Baja	Baja	Alta
<b>AhpSubcriterioManagerImpl</b>	2	Baja	Baja	Alta

### CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

AnpAlternativaAnpAlternativaDAO	2	Baja	Baja	Alta
AnpAlternativaAnpCriterioDAO	2	Baja	Baja	Alta
AnpAlternativaDAO	3	Baja	Baja	Alta
AnpComponenteDAO	3	Baja	Baja	Alta
AnpCriterioAnpCriterioDAO	2	Baja	Baja	Alta
AnpCriterioDAO	3	Baja	Baja	Alta
AnpDataAccessFacade	19	Alta	Alta	Baja
AnpAlternativaAnpAlternativaDAOImpl	2	Baja	Baja	Alta
AnpAlternativaAnpCriterioDAOImpl	2	Baja	Baja	Alta
AnpAlternativaDAOImpl	3	Baja	Baja	Alta
AnpComponenteDAOImpl	3	Baja	Baja	Alta
AnpCriterioAnpCriterioDAOImpl	2	Baja	Baja	Alta
AnpCriterioDAOImpl	3	Baja	Baja	Alta
AnpDataAccessFacadeImpl	17	Alta	Alta	Baja
AnpFacade	16	Alta	Alta	Baja
AnpFacadeImpl	18	Alta	Alta	Baja
ANP_Controller	37	Alta	Alta	Baja
ANP_MatrizGenerada	7	Media	Media	Media
AnpAlternativaAnpAlternativaManager	3	Baja	Baja	Alta
AnpAlternativaAnpCriterioManager	3	Baja	Baja	Alta
AnpAlternativaManager	5	Baja	Baja	Alta
AnpComponenteManager	5	Baja	Baja	Alta
AnpCriterioAnpCriterioManager	3	Baja	Baja	Alta
AnpCriterioManager	5	Baja	Baja	Alta
AnpAlternativaAnpAlternativaManagerImpl	2	Baja	Baja	Alta
AnpAlternativaAnpCriterioManagerImpl	2	Baja	Baja	Alta
AnpAlternativaManagerImpl	4	Baja	Baja	Alta
AnpComponenteManagerImpl	4	Baja	Baja	Alta
AnpCriterioAnpCriterioManagerImpl	2	Baja	Baja	Alta
AnpCriterioManagerImpl	4	Baja	Baja	Alta

Tabla 21: Evaluación de las clases del sistema mediante la métrica TOC.

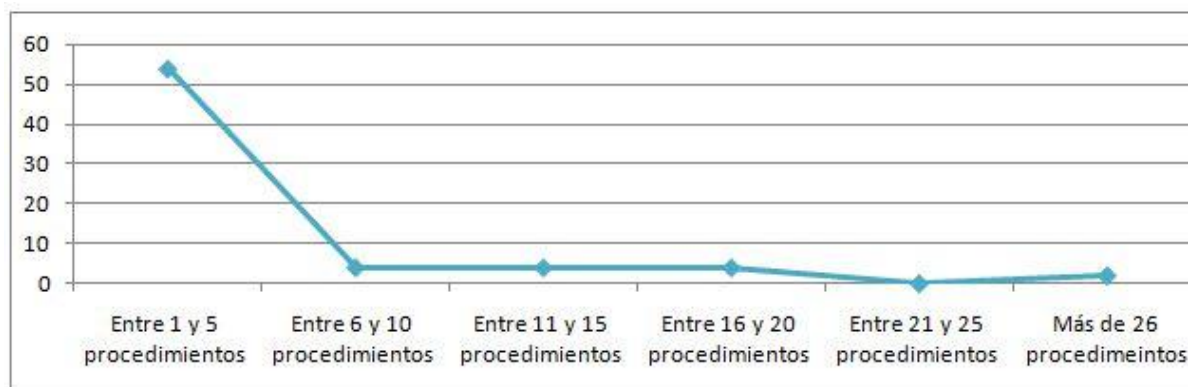


Figura 10: Representación de la cantidad de clases agrupadas en intervalos según la cantidad de procedimientos.

Las gráficas que corresponden a los resultados obtenidos se presentan en las siguientes figuras.

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

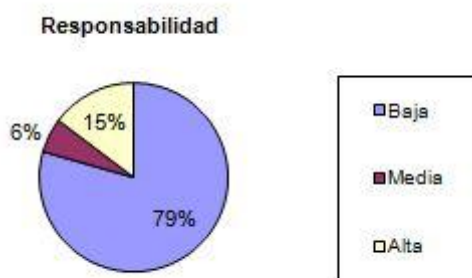


Figura 11: Representación en por ciento (%) de los resultados obtenidos en el atributo Responsabilidad.



Figura 12: Representación en por ciento (%) de los resultados obtenidos en el atributo Complejidad de implementación.

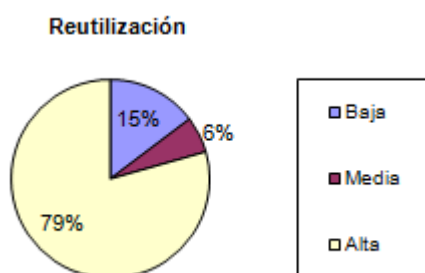


Figura 13: Representación en por ciento (%) de los resultados obtenidos en el atributo Reutilización.

Como resultado de la aplicación de la métrica TOC se evidencia que las clases del sistema poseen una alta reutilización, baja responsabilidad y baja complejidad de implementación por lo que el diseño de las clases en cuanto a cantidad de funcionalidades por cada una es bueno.

### 3.3.2 Métrica RC

Está dado por el número de relaciones de uso de una clase con otra y evalúa los siguientes atributos de calidad [57].

#### Atributos de Calidad [56]:

- **Acoplamiento:** Un aumento del RC implica un aumento del Acoplamiento de la clase.
- **Complejidad de mantenimiento:** Un aumento del RC implica un aumento de la complejidad del mantenimiento de la clase.

### CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

- **Cantidad de pruebas:** Un aumento del RC implica un aumento de la Cantidad de pruebas de unidad necesarias para probar una clase.
- **Reutilización:** Un aumento del RC implica una disminución en el grado de reutilización de la clase.

	Categoría	Criterio
<b>Acoplamiento</b>	Ninguno	0
	Bajo	1
	Medio	2
	Alto	>2
<b>Complejidad de Mantenimiento</b>	Baja	<= Promedio
	Media	Entre Promedio y 2* Promedio.
	Alta	> 2* Promedio.
<b>Reutilización</b>	Baja	>2* Promedio.
	Media	Entre Promedio y 2* Promedio.
	Alta	<= Promedio.
<b>Cantidad de Pruebas</b>	Baja	<= Promedio.
	Media	Entre Promedio y 2* Promedio.
	Alta	> 2* Promedio.

Tabla 22: Rangos de valores para la evaluación técnica de los atributos de calidad relacionados con la métrica RC [58].

A continuación se muestran las clases del sistema evaluadas en los atributos de calidad Acoplamiento, Complejidad del mantenimiento, Cantidad de pruebas y Reutilización.

Clase	Cantidad de Relaciones de Uso	Acoplamiento	Comp. Mant.	Reutilización	Cantidad de Pruebas
<b>CommonDataAccesFacade</b>	2	Medio	Media	Media	Media
<b>DecisionDAO</b>	1	Bajo	Baja	Alta	Baja
<b>UsuarioDAO</b>	1	Bajo	Baja	Alta	Baja
<b>CommonDataAccesFacadeImpl</b>	1	Bajo	Baja	Alta	Baja
<b>DecisionDAOImpl</b>	1	Bajo	Baja	Alta	Baja
<b>UsuarioDAOImpl</b>	1	Bajo	Baja	Alta	Baja
<b>CommonFacade</b>	5	Alto	Alta	Baja	Alta
<b>CommonFacadeImpl</b>	1	Bajo	Baja	Alta	Baja
<b>DecisionManager</b>	1	Bajo	Baja	Alta	Baja
<b>UsuarioManager</b>	1	Bajo	Baja	Alta	Baja
<b>DecisionManagerImpl</b>	1	Bajo	Baja	Alta	Baja
<b>UsuarioManagerImpl</b>	1	Bajo	Baja	Alta	Baja
<b>AhpAlternativaAhpCriterioDAO</b>	1	Bajo	Baja	Alta	Baja
<b>AhpAlternativaAhpSubcriterioDAO</b>	1	Bajo	Baja	Alta	Baja
<b>AhpAlternativaDAO</b>	1	Bajo	Baja	Alta	Baja
<b>AhpCriterioDAO</b>	1	Bajo	Baja	Alta	Baja
<b>AhpDataAccesFacade</b>	6	Alto	Alta	Baja	Alta
<b>AhpSubcriterioDAO</b>	1	Bajo	Baja	Alta	Baja
<b>AhpAlternativaAhpCriterioDAOImpl</b>	1	Bajo	Baja	Alta	Baja
<b>AhpAlternativaAhpSubcriterioDAOImpl</b>	1	Bajo	Baja	Alta	Baja
<b>AhpAlternativaDAOImpl</b>	1	Bajo	Baja	Alta	Baja
<b>AhpCriterioDAOImpl</b>	1	Bajo	Baja	Alta	Baja
<b>AhpDataAccesFacadeImpl</b>	1	Bajo	Baja	Alta	Baja
<b>AhpSubcriterioDAOImpl</b>	1	Bajo	Baja	Alta	Baja

### CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

AhpFacade	12	Alto	Alta	Baja	Alta
AhpFacadelmpl	1	Bajo	Baja	Alta	Baja
AHP_Controller	12	Alto	Alta	Baja	Alta
AHP_MatrizGenerada	1	Bajo	Baja	Alta	Baja
AhpAlternativaAhpCriterioManager	1	Bajo	Baja	Alta	Baja
AhpAlternativaAhpSubcriterioManager	1	Bajo	Baja	Alta	Baja
AhpAlternativaManager	1	Bajo	Baja	Alta	Baja
AhpCriterioManager	1	Bajo	Baja	Alta	Baja
AhpSubcriterioManager	1	Bajo	Baja	Alta	Baja
AhpAlternativaAhpCriterioManagerImpl	1	Bajo	Baja	Alta	Baja
AhpAlternativaAhpSubcriterioManagerImpl	1	Bajo	Baja	Alta	Baja
AhpAlternativaManagerImpl	1	Bajo	Baja	Alta	Baja
AhpCriterioManagerImpl	1	Bajo	Baja	Alta	Baja
AhpSubcriterioManagerImpl	1	Bajo	Baja	Alta	Baja
AnpAlternativaAnpAlternativaDAO	1	Bajo	Baja	Alta	Baja
AnpAlternativaAnpCriterioDAO	1	Bajo	Baja	Alta	Baja
AnpAlternativaDAO	1	Bajo	Baja	Alta	Baja
AnpComponenteDAO	1	Bajo	Baja	Alta	Baja
AnpCriterioAnpCriterioDAO	1	Bajo	Baja	Alta	Baja
AnpCriterioDAO	1	Bajo	Baja	Alta	Baja
AnpDataAccessFacade	7	Alto	Alta	Baja	Alta
AnpAlternativaAnpAlternativaDAOImpl	1	Bajo	Baja	Alta	Baja
AnpAlternativaAnpCriterioDAOImpl	1	Bajo	Baja	Alta	Baja
AnpAlternativaDAOImpl	1	Bajo	Baja	Alta	Baja
AnpComponenteDAOImpl	1	Bajo	Baja	Alta	Baja
AnpCriterioAnpCriterioDAOImpl	1	Bajo	Baja	Alta	Baja
AnpCriterioDAOImpl	1	Bajo	Baja	Alta	Baja
AnpDataAccessFacadelmpl	1	Bajo	Baja	Alta	Baja
AnpFacade	15	Alto	Alta	Baja	Alta
AnpFacadelmpl	1	Bajo	Baja	Alta	Baja
ANP_Controller	8	Alto	Alta	Baja	Alta
ANP_MatrizGenerada	1	Bajo	Baja	Alta	Baja
AnpAlternativaAnpAlternativaManager	1	Bajo	Baja	Alta	Baja
AnpAlternativaAnpCriterioManager	1	Bajo	Baja	Alta	Baja
AnpAlternativaManager	1	Bajo	Baja	Alta	Baja
AnpComponenteManager	1	Bajo	Baja	Alta	Baja
AnpCriterioAnpCriterioManager	1	Bajo	Baja	Alta	Baja
AnpCriterioManager	1	Bajo	Baja	Alta	Baja
AnpAlternativaAnpAlternativaManagerImpl	1	Bajo	Baja	Alta	Baja
AnpAlternativaAnpCriterioManagerImpl	1	Bajo	Baja	Alta	Baja
AnpAlternativaManagerImpl	1	Bajo	Baja	Alta	Baja
AnpComponenteManagerImpl	1	Bajo	Baja	Alta	Baja
AnpCriterioAnpCriterioManagerImpl	1	Bajo	Baja	Alta	Baja
AnpCriterioManagerImpl	1	Bajo	Baja	Alta	Baja

Tabla 23: Clases del Sistema evaluadas en los atributos de calidad según la métrica RC.

A continuación se muestra una gráfica con agrupaciones por intervalos de la cantidad de clases según las dependencias entre ellas.

### CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

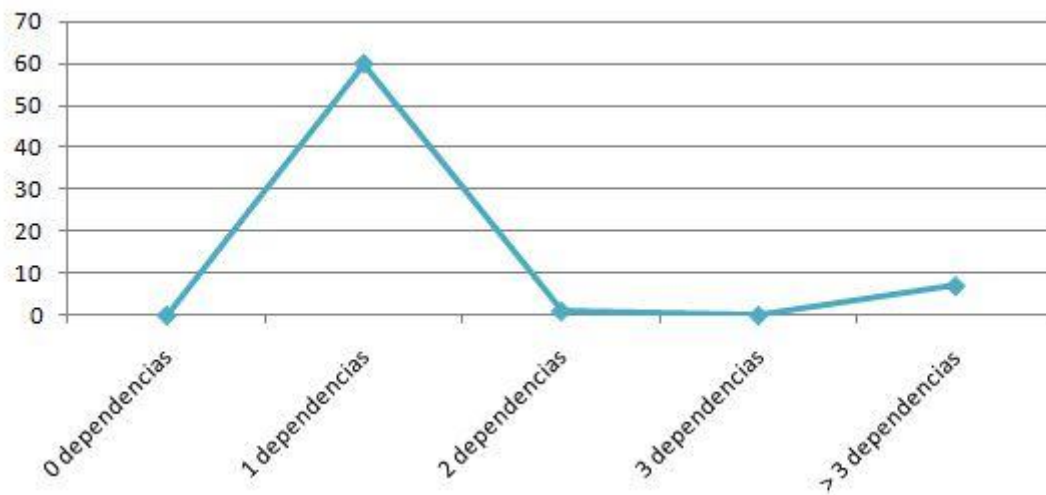


Figura 14: Intervalos de las clases agrupadas según las dependencias entre ellas.

Las gráficas que corresponden a los resultados obtenidos se presentan en las siguientes figuras.

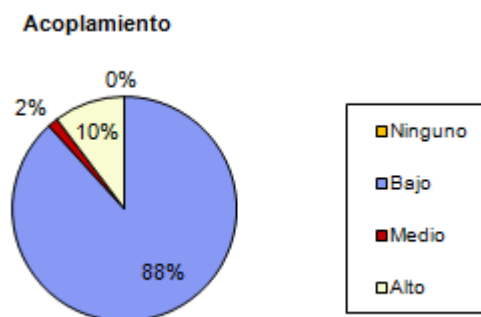


Figura 15: Representación en porcentos (%) de los atributos obtenidos en el atributo Acoplamiento.



Figura 16: Representación en porcentos (%) de los atributos obtenidos en el atributo Complejidad de Mantenimiento.



Figura 17: Representación en porcentos (%) de los atributos obtenidos en el atributo Cantidad de Pruebas.

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA



Figura 18: Representación en porcentajes (%) de los atributos obtenidos en el atributo Reutilización.

Como resultado de la aplicación de la métrica anteriormente expuesta se evidencia que las clases del sistema poseen una alta reutilización, baja complejidad de mantenimiento, bajo acoplamiento y la cantidad de pruebas a realizar es mínima, por lo que se puede concluir que el diseño de las clases en cuanto a cantidad de relaciones de cada una en el sistema es bueno.

### 3.4 Validación de las variables de la investigación

Para realizar la validación de las variables de la investigación se desarrolló un caso de estudio aplicando los métodos AHP y ANP de forma manual y utilizando el software MultiDecision PAAT, además en el caso del método AHP se utilizó el software Jerarquías para comparar los resultados obtenidos en cada una de las variantes anteriores. Las variables que se evaluaron fueron: demora en el proceso de decisión, susceptibilidad a errores y complejidad de la confección. Los resultados se muestran en las siguientes gráficas:

#### Utilizando el método AHP



Figura 19: Demora en el proceso de decisión utilizando AHP.



### CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA



Figura 20: Susceptibilidad a errores utilizando AHP.

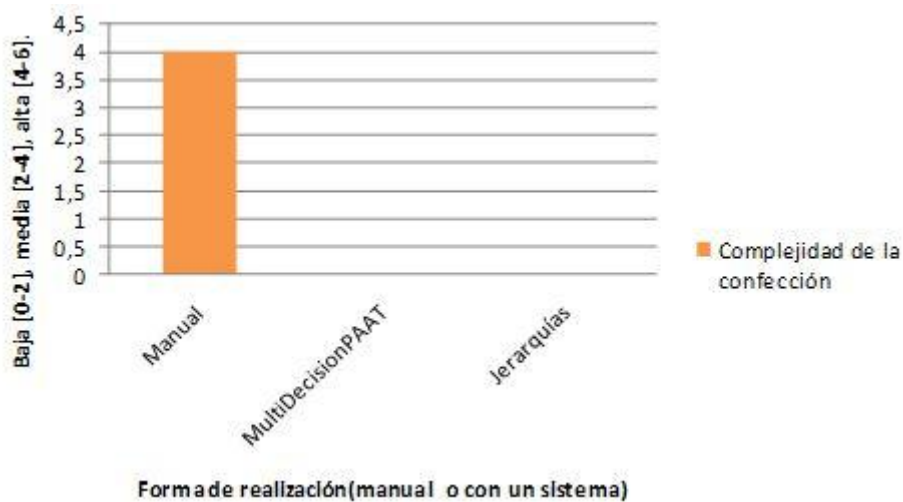


Figura 21: Complejidad de la confección utilizando AHP.

### Utilizando el método ANP

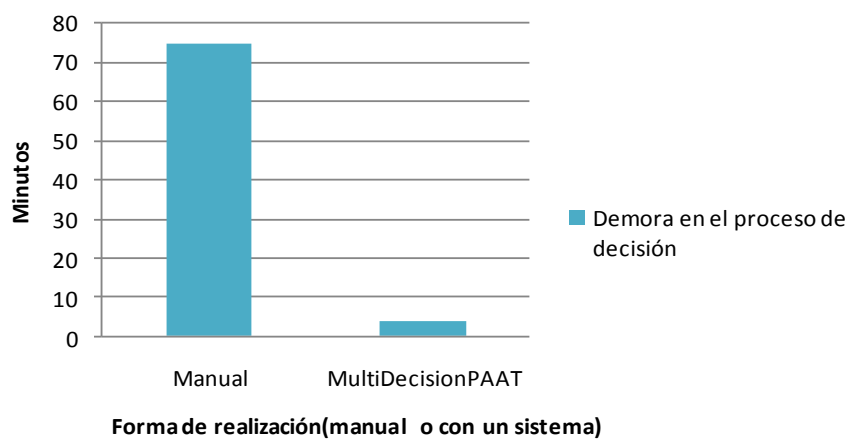


Figura 22: Demora en el proceso de decisión utilizando ANP.

## CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

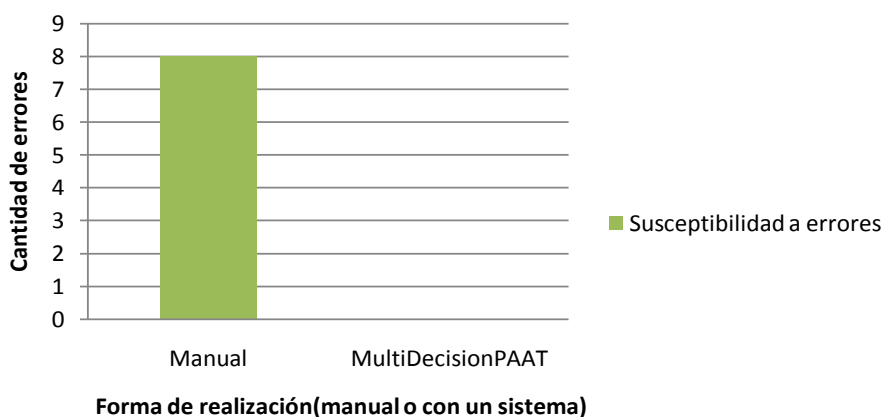


Figura 23: Susceptibilidad a errores utilizando ANP.

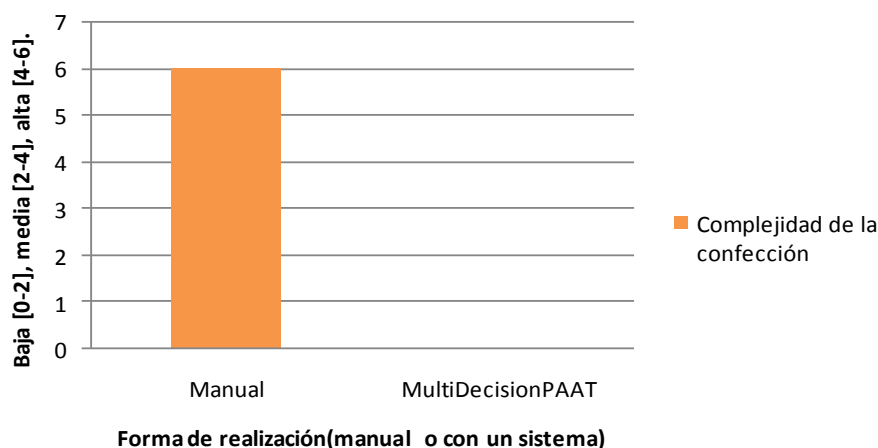


Figura 24: Complejidad de la confección utilizando ANP.

A modo de conclusión se puede afirmar que el sistema MultiDecision PAAT ofrece los mejores resultados para las variables analizadas en comparación con su implementación de forma manual, en el caso software Jerarquías se obtienen resultados similares en cuanto a la susceptibilidad a errores y a la complejidad de la confección, sin embargo MultiDecision PAAT ofrece mejores resultados en cuanto al tiempo de demora en el proceso de decisión.

En el anexo 8 se muestra la realización del caso de estudio analizado.

### Conclusiones parciales

- Se realizaron el diagrama de componentes y de despliegue para mostrar una vista de cómo quedó la aplicación a nivel de componentes y su distribución.
- Se describieron los diseños de casos de pruebas para realizar pruebas funcionales a la aplicación los cuales arrojaron en todos los casos resultados satisfactorios.

### CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

- Se aplicaron las métricas TOC y RC que arrojaron resultados positivos en la valoración de la validación de la aplicación propuesta.
- Se validaron las variables de la investigación.
- Con las pruebas y validaciones realizadas se puede concluir que la herramienta desarrollada cumple con las especificaciones y requisitos definidos por los clientes en la etapa de concepción del sistema.

1. El análisis de los métodos AHP y ANP permitió identificar los requisitos y determinar los que se implementan en el sistema basado en los pasos descritos por Saaty.
2. Para el desarrollo del sistema se empleó la metodología SXP y la herramienta CASE Visual Paradigm para la descripción y modelación del mismo, como sistema gestor de base de datos PostgreSQL, los marcos de trabajo Vaadin, Spring e Hibernate y el entorno integrado de desarrollo Eclipse STS.
3. El módulo implementa 17 funcionalidades que permiten la ejecución de los pasos planteados por los métodos.
4. La aplicación de pruebas de caja negra permitió validar las funcionalidades de la aplicación y los resultados de las métricas TOC y RC validan el diseño realizado. Estas demuestran que la herramienta desarrollada cumple con las especificaciones y requisitos definidos en la etapa de concepción del sistema.
5. Se compara mediante un caso de estudio las variables definidas en la investigación, a partir de desarrollar un caso manual, utilizando el software Jerarquías y el sistema desarrollado MultiDecision PAAT; las mejoras obtenidas demuestran que se cumple el objetivo de la investigación y se valida la idea a defender.

1. Implementar las variantes difusas de los métodos AHP y ANP.
2. Exportar los resultados del análisis en distintos formatos .pdf, .doc, .xls.

1. Simon, Hebert, "The New Science of Management Decision", in Harper and Row. 1960: New York.
2. Hurtado et al, T., Bruno, Gérard, *El proceso de análisis jerárquico (AHP) como herramienta para la toma de decisiones en la selección de proveedores*, Tesis Digitales UNMSM. Lima, 2005.
3. *Métodos Teóricos*. Available from: <http://www.buenastareas.com/ensayos/Metodos-Teoricos/136411.html>.
4. Zayas, A.P.M., *El rombo de las Investigaciones de las Ciencias Sociales*.
5. *Métodos empíricos*. Available from: <http://es.scribd.com/doc/21229743/METODOS-EMPIRICOS>.
6. *Observación Descriptiva y Experimento*. Available from: <HTTP://WWW2.UIAH.FI/PROJECTS/METODI/262.HTM>.
7. Custodio, R.A. *Métodos y Técnicas*. 2008; Available from: <http://www.gestiopolis.com/economia/tecnicas-y-metodos-de-investigacion.htm>.
8. SlideShare. *Tipos de Sistemas de Información*. 2009; Available from: <http://www.slideshare.net/about>.
9. Soto, C.E., *Decisiones en ambientes de incertidumbre*, Revista de Servicio Civil: San José, Costa Rica.
10. Gómez, A.; Available from: [http://www.investigacion-operaciones.com/Curso\\_inv-Oper\\_carpeta/Clase21\\_II.pdf](http://www.investigacion-operaciones.com/Curso_inv-Oper_carpeta/Clase21_II.pdf).
11. Finlay, P.N., *Introducing decision support systems*. 1994: Oxford, UK Cambridge.
12. Turban, E., *Decision support and expert systems: management support systems*. 1995.
13. Little, J.D.C., "Models and Managers: The Concept of a Decision Calculus.", in *Management Science*. 1970.
14. Keen, P.G.W., *Decision support systems: an organizational perspective*. 1978, ISBN 0-201-03667-3.
15. Moore, J.H., Chang, M.G., *Design of Decision Support Systems*, in *Data Base*. 1980.
16. Power, D.J., *What is a DSS? The On-Line Executive Journal for Data-Intensive Decision Support* 1. 1997.
17. Holsapple et al, C.W., Whinston, Andrew B., *Decision Support System: aknowledge- based approach*. 1996, Los Angeles.
18. Haettenschwiler, P., *Neues anwenderfreundliches Konzept der Entscheidungsunterstützung.*, in *Gutes Entscheiden in Wirtschaft*. 1999, 189-208.
19. Dominguez , A.J., Medina, Garrido Jose Aurelio, *El sistema de información como soporte a las decisiones*, in *La gestión de los sistemas de información en la empresa*. 2002, Ediciones Pirámide: Madrid.
20. Delfos, C.E., *Software para la implementación de los Sistemas de Soporte a las Decisiones (SSD) a nivel Internacional*. 2008.
21. Alfonso, O. *Procedimiento multicriterio basado en herramientas de toma de decisiones para la selección de proveedores*. 2009.
22. Font, Elena. *Gestión de Información en el Proceso Analítico Jerárquico para la toma de decisiones de nuevos productos*. Metodología de Implantación. Tesis Doctoral. Dpto

## REFERENCIAS

- Economía. Facultad de Economía. Universidad de la Habana, Ciudad de la Habana, Cuba. 2000.
23. Osorio Gómez, Juan Carlos; *El Proceso de Análisis Jerárquico (AHP) y la Toma de Decisiones Multicriterio. Ejemplo de Aplicación*. Universidad Tecnológica de Pereira. Septiembre de 2008.
  24. Ávila Mogollón, Ruth Maritza. RMAM, *El AHP (Proceso Analítico Jerárquico) y su aplicación para determinar los usos de las tierras, el caso de Brasil, Proyecto regional "Información sobre tierras y aguas para un desarrollo agrícola sostenible"*. Santiago Chile. Diciembre 2000.
  25. Rodríguez Bello, Sandra; *Toma Decisión Multicriterios con AHP, ANP y Lógica Difusa*; Maestría en Ingeniería de Sistemas y Computación, Universidad Nacional de Colombia.
  26. Figueroa, R.G., Solís, Camilo J. , Cabrera, Armando A. , *Metodologías tradicionales vs metodologías ágiles*.
  27. Infante, L. *Metodología Agil - Scrum*. 2009.
  28. Canós, J.H., Letelier, Patricio, Panadés, M. Carmen. *Metodologías ágiles en el desarrollo de software*. Available from: <http://www.willydev.net/descargas/prev/TodoAgil.Pdf>.
  29. Davis, A., "A History of Agile Methods", in *JISBD 2002 Libertarios Fundamentalistas Tendencia global*. 2002.
  30. *Rompiendo Paradigmas*. Available from: <http://paradigmas14.blogspot.com/p/metodologia-utilizar.html>.
  31. *Ventajas de la metodología Scrum*. Available from: <http://www.clubdesarrolladores.com/articulos/mostrar/63-metodologia-scrum/2>.
  32. Peñalver et al, G.M., A. García, S., *SXP, Metodología Ágil para el Desarrollo de Software*, in *1er Congreso Iberoamericano de Ingeniería de proyectos*. 2010: Chile.
  33. Carmina Lizeth Torres Flores, Germán Harvey Alférez Salinas Navoja . *Establecimiento de una Metodología de Desarrollo de Software para la Universidad de Navoja usando OpenUP*. s.n., 2008.
  34. Rumbaugh, James, Ivar . *El Lenguaje Modificado de Modelado*. 2000. pág. 19.
  35. *Visual Paradigm*. Available from: <http://www.visual-paradigm.com>.
  36. Johnson, R., *Professional Java Development with the Spring Framework*. 2005.
  37. Areces, G.A., Diaz, Marquez Iskael, "Diseño e implementación de las capas de negocio y acceso a datos de los módulos Planificación y Ejecución de Visitas Familiares". 2008.
  38. *Relational Persistence for Java*.
  39. Cameron, N., *PostgreSQL affiliates .ORG domain*. 2003.
  40. *Netbeans User FAQ*. 2007.
  41. Fernando, D.M. *Patrones de Diseño de Arquitecturas de Software Enterprise*. 2005; Available from: <http://materias.fi.uba.ar/7500/montaldo-tesisdegradoingenieriainformatica.pdf>.
  42. *The Eclipse Foundation open source community website*.
  43. *Elección del servidor de aplicaciones web*. 2003; Available from: <http://www.help400.es/asp/scripts/nwart.asp?Num=131&Pag=10&Tip=T>.
  44. *Servidor Web Tomcat*. 2011; Available from: [http://www.ecured.cu/index.php/Servidor\\_Tomcat](http://www.ecured.cu/index.php/Servidor_Tomcat).
  45. Shaw, D.G.y.M., *An introduction to software architecture*. 2000.
  46. *Microsoft Patterns & Practices*. 2004.
  47. *Patrones Grasp*. 2011; Available from: <http://www.buenastareas.com/ensayos/Patrones-Grasp/1896730.html>.

## REFERENCIAS

48. Nieto et al , D.J., Ramos, Fernández Pablo *El Patrón Fachada*: Universidad de Salamanca.
49. Lago, R. *Patrón "Data Access Object"*. 2007; Available from: <http://www.proactiva-calidad.com/java/patrones/DAO.html>.
50. *Diagrama de clases*. 2005; Available from: [http://www-2.dc.uba.ar/materias/isoft1/is1-2005\\_2/apuntes/SlidesDC.pdf](http://www-2.dc.uba.ar/materias/isoft1/is1-2005_2/apuntes/SlidesDC.pdf).
51. Fernández, R.J., *Modelo de Datos*.
52. Arizaca, R.E., *Artefacto: Diagrama de Componentes*. 2009: La Paz - Bolivia.
53. *Técnicas de Pruebas*. Available from: [https://docs.google.com/viewer?a=v&q=cache:L0\\_moMWzYTYJ:indalog.ual.es/mtorres/LP/Prueba.pdf+pruebas+de+caja+negra&hl=es&gl=cu&pid=bl&srcid=ADGEEShyq3ux79zqPpID1O92UU\\_M3GdGAlm2GwqSjSr2t - ZPK92JW5\\_hBCcumq1FcxUBeqrpgyuQ6QzRHEujVuzZ1T961tR-Omog9-fZtsh1JqlqFEOkiBLEYCPjgT-GyyMPmFmgdJ9&sig=AHIEtbQ\\_GHqnVZF5zNLQ7I7Epp26iQ3Bpg](https://docs.google.com/viewer?a=v&q=cache:L0_moMWzYTYJ:indalog.ual.es/mtorres/LP/Prueba.pdf+pruebas+de+caja+negra&hl=es&gl=cu&pid=bl&srcid=ADGEEShyq3ux79zqPpID1O92UU_M3GdGAlm2GwqSjSr2t - ZPK92JW5_hBCcumq1FcxUBeqrpgyuQ6QzRHEujVuzZ1T961tR-Omog9-fZtsh1JqlqFEOkiBLEYCPjgT-GyyMPmFmgdJ9&sig=AHIEtbQ_GHqnVZF5zNLQ7I7Epp26iQ3Bpg).
54. PRESSMAN, R. *Ingeniería del Software: Un Enfoque Práctico* Mc Graw, 2005.
55. LORENZ, M. and J. KIDD *Object-oriented software metrics* *Journal of Systems and Software*, 1994.
56. Driggs Vélez, J.C., *Diseño e Implementación de la nueva versión del Módulo Banco del Sistema Integral de Gestión Cedrux*. 2011.
57. EcuRed. *Métricas de diseño*. 2012; Available from: [http://www.ecured.cu/index.php/M%C3%A9trica\\_de\\_dise%C3%B1o](http://www.ecured.cu/index.php/M%C3%A9trica_de_dise%C3%B1o).
58. Romero, D.B., *Sistema para gestionar la actividad científica del Departamento de informática de la universidad de Guantánamo*, U.d.G.G. Departamento de informática Editor. 2010.