

**Universidad de las Ciencias Informáticas**  
**Facultad #1**



***“Conjunto de Métricas para la evaluación de  
Bases de Datos Relacionales”***

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

**Autores:**

**Yaima Mederos Herrera  
Aniubis Rodríguez Batista**

**Tutor:**

**MSc. Edistio Yoel Verdecia Martínez**

**Mayo, 2009**

**“Año del 50 Aniversario del triunfo de la Revolución”**



*“La responsabilidad nuestra es luchar porque la calidad del producto que aquí se haga sea de las mejores y la mejor posible...”*

*Ernesto Che Guevara*

## DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del trabajo titulado:

*“Conjunto de Métricas para la evaluación de Bases de Datos Relacionales”*

y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

**Yaima Mederos Herrera**

**Aniubis Rodríguez Batista**

\_\_\_\_\_

Firma del Autor

\_\_\_\_\_

Firma del Autor

**MSc. Edistio Yoel Verdecia Martínez**

\_\_\_\_\_

Firma

## DATOS DE CONTACTO

**Nombre y Apellidos:** Edistio Yoel Verdecia Martínez.

**Institución:** Universidad de las Ciencias Informáticas (UCI).

**E-mail:** [edisitoyoel@uci.cu](mailto:edisitoyoel@uci.cu)

Graduado de Cibernética – Matemática, de la Universidad de la Habana. Año 1996. Profesor Auxiliar. 2008. Máster en Gestión de Proyectos Informáticos. 2008. Ha participado en diferentes proyectos productivos de la universidad, se dedica fundamentalmente a la docencia en el área de la programación y las BD. Actualmente es Jefe de Departamento de Técnicas de Programación en la Facultad 1. Ha obtenido diferentes reconocimientos tanto dentro y fuera de la Universidad. Sus investigaciones fundamentales están orientadas a la evaluación y certificación de competencias tema en el cual desarrolla su Doctorado en Ciencias Pedagógicas.

## DEDICATORIA

*A mis padres.. que han esperado este día tanto como yo, por todo el apoyo y el amor que me dan dado en cada instante de mi vida, por darme la fuerza necesaria para enfrentar la vida y seguir adelante.. sin ustedes esto no hubiese sido posible.*

Aniubis

*A mis dos madres por sus esfuerzos incansables por darme lo mejor, para ustedes será siempre todo lo bueno que alcance en mi vida.*

*A mi papá por todo el apoyo, cariño y dedicación que siempre me ha dado.*

*A mis hermanas queridas que son mi vida, esto es para ustedes.*

*A mi novio Antonio por su apoyo y amor incondicional.*

*A mis familiares y amigos que confiaron siempre en mí.*

Yaima

## AGRADECIMIENTOS

A mi mamita por todo su apoyo y cariño. Por estar ahí siempre que lo he necesitado, que siempre ha puesto a sus hijas por encima de todo y ha luchado tanto para sacarnos adelante y somos lo que somos por ella. Te quiero con la vida.

A mi abuelita linda que ha sido como madre para mí, por todo su cariño. Por todo lo que ha luchado por nosotras... por las peleas que tanto nos ha dado, es para ti todo lo bueno que haga porque sin ti esto no hubiera sido posible.

A mi papá por todo su cariño y comprensión, por estar ahí cuando lo he necesitado y apoyarme en todo momento. Muchos besos para ti mi flaco... te quiero.

A mi novio Antonio por su amor y comprensión, por estar a mi lado estos 5 años y ayudarme tanto, consolarme cuando las cosas no me han salido bien y su apoyo y darme las fuerzas para seguir adelante, por todas las malas noche ayudándome y enseñándome. Te Amo nene...

A mi hermana querida, Yamilka por su ayuda, comprensión y cariño. Por poder confiar en ti y por ayudarme tanto. Siempre podrás contar conmigo... te quiero mi negrita. jejeje... muass

A mi hermanita Yailena por ser la fuente de muchas de mis alegrías, que con su cariño y simpatía puede darle fuerza y esperanza a todos aquellos que la conocen. te quiero mucho mi chiquitica.

A mi abuela Selina por todo el cariño que me ha brindado, por ayudarme en los momentos difíciles. Besos.

A Alfredo por su apoyo y la ayuda incondicional que me ha brindado. Que aunque el no lo crea lo aprecio mucho.

A mis familiares que siempre me han apoyado mucho y han estado conmigo en los momentos más difíciles: mis tíos que son como padres para mí y a mis primos y primas que siempre he podido contar con ustedes. Los quiero mucho a todos.

A mi tío Rafaely mi madrina Gema por todo el apoyo que siempre me han brindado.

A la familia de mi novio por haberme acogido como una más de la familia, por el cariño y atención que me han dado.

A mis amigas Roxana y Daniuska porque siempre he podido contar con ustedes. Por apoyarme siempre y estar conmigo en los momentos más difíciles.

A Aniubis mi compañera de tesis por su apoyo, por todas las cosas y momentos que compartimos. Por poder contar con ella para todo. Me alegra mucho que hayamos sido compañeras para esta tarea tan importante en nuestras vidas porque no solo encontré una compañera de tesis sino una amiga.

En fin a toda aquella persona que ha ayudado de alguna forma en que este sueño se haya hecho realidad, y en especial a mis familiares que son todo para mi.

Yaima

## AGRADECIMIENTOS

A mis padres, por la eterna confianza, por el sacrificio y por ayudarme a ser la persona que soy.

A mi hermana por todo su cariño y por estar siempre ahí y mi cuñado por ser como otro hermano.

A Isuel Méndez, mi amigo y mi amor, que es lo más lindo que me ha pasado en la vida, por estar junto a mí dándome fuerzas y apoyo en los momentos más difíciles, por escucharme pacientemente y aconsejarme cuando más lo he necesitado, por todos los momentos buenos que hemos pasado, por enseñarme tantas cosas.. por todo tu amor...

A todos los que me brindaron su apoyo incondicional, A los que siempre estuvieron junto a mí sin importar la distancia, A todos los que me ayudaron, A los que me alentaron, A los que creyeron en mí, Sin ellos hubiera sido realmente difícil llegar hasta aquí.

A Yaisel, por saber que puedo contar contigo.

A Yaima, mi compañera de tesis, por todo el tiempo que pasamos juntas, por todo su optimismo, y porque el éxito no se alcanza sino se tiene con quien compartirlo.

A mis compañeros, amigos de la vieja guardia y de la UCI por estar siempre ahí, y por hacer que estos años sean inolvidables.

A todas las personas que hicieron posible la realización de este trabajo.

Aniubis

A nuestro tutor Edistio por toda la ayuda brindada y el tiempo dedicado. Por habernos guiado para que este trabajo tuviera éxito. Por la paciencia de haber soportado tanta insistencia de sus tesisistas mas quisquillosas. Gran parte de este éxito se lo debemos a él.

A la Revolución y a Fidel por darnos la oportunidad de cumplir nuestros sueños y ser partícipes de esta maravillosa idea.

A todas las personas que nos ayudaron en el desarrollo de este trabajo. A todos nuestros compañeros y amigos que estuvieron pendiente de todo.

Yaima y Aniubis

## **RESUMEN**

En la actualidad, las entidades dedicadas a la producción de software, tanto en Cuba como en el mundo, tienen entre sus principales objetivos desarrollar productos y servicios informáticos de alta calidad. Para poder obtener estos resultados y lograr un posicionamiento y reconocimiento en el mercado, es necesaria la implantación de estándares de calidad que garanticen la comprobación objetiva de la evaluación de la calidad de los productos de software, además de la certificación del mismo. El objetivo principal del presente trabajo es: realizar una propuesta de métricas que permitan evaluar la calidad de las BD relacionales desde una concepción lo más integral posible. Para lograr lo planteado anteriormente se realizó un estudio exhaustivo de diversos temas como: Calidad de BD, Bases de Datos(BD) y Métricas para BD; se analizaron un conjunto de métricas existentes; se realizó un proceso de prueba tomando como muestra algunos diseños de BD de proyectos reales de la Universidad de las Ciencias Informáticas (UCI), donde se registraron los resultados para un posterior proceso de refinamiento de las mismas conformándose así la propuesta que finalmente fue validada teórica y empíricamente. La propuesta es aplicable a toda Base de Datos independientemente del gestor en que se haya implementado y esto ayudará a mejorar la calidad de los sistemas informáticos o soluciones integrales que se desarrollan en la universidad.

## **PALABRAS CLAVE**

Calidad, Bases de Datos, Métricas



## Índice de Contenido:

Introducción .....	1
1.1. Introducción.....	5
1.2. Sistemas informáticos.....	5
1.2.1. Calidad en la producción de Sistemas Informáticos.....	8
1.2.1.1. ¿Qué es Calidad? .....	8
1.2.1.2. ¿Qué es calidad de software? .....	9
1.3. Bases de datos.....	9
1.3.1. Definiciones.....	9
1.3.2. Principales propiedades .....	10
1.3.3. Características de las bases de datos .....	10
1.3.4. Metodologías de diseño de base de datos.....	10
1.3.5. Arquitectura de las bases de datos .....	11
1.3.6. Ciclo de vida de las aplicaciones de bases de datos.....	12
1.3.7. Ventajas en el uso de base de datos .....	12
1.4. Sistemas gestores de bases de datos.....	13
1.4.1. Clasificación de los SGBD.....	14
1.4.2. Características de los principales SGBD .....	14
1.4.3. Tendencias tecnológicas y del mercado .....	15
1.5. Métricas en el proceso de análisis y diseño de sistemas informáticos.....	16
1.5.1. Definiciones .....	16
1.5.2. Clasificación de las métricas.....	17
1.5.3. Características de las métricas .....	18
1.5.4. ¿Por qué es importante medir? .....	18
1.5.5. Proceso de medición .....	19
1.5.6. Métricas y calidad .....	19
1.5.7. Las métricas en la industria cubana del software .....	20
1.5.8. Situación actual de la utilización de métricas de software en la UCI .....	20

1.5.9.	Descripción de la encuesta.....	21
1.5.10.	Análisis de la encuesta .....	21
1.5.11.	Resultados de la investigación .....	23
1.6.	Métricas y bases de datos .....	24
1.6.1.	Características de un esquema conceptual con calidad .....	24
1.6.2.	Tendencias y Evolución .....	25
1.6.3.	Deficiencias de las métricas de calidad para bases de datos .....	25
1.7.	Conclusiones.....	25
Capítulo 2	.....	27
2.1.	Introducción.....	27
2.2.	Análisis de las métricas para modelos conceptuales.....	27
2.2.1.	Clasificación.....	27
2.2.2.	Caracterización .....	28
2.2.2.1.	Propuesta de Moody y Shanks .....	28
2.2.2.2.	Métricas de Kesh.....	29
2.2.2.3.	Métricas de Moody .....	31
2.2.2.4.	Métricas de Varas y Pradenas .....	32
2.2.2.5.	Métricas de Piattini .....	35
2.2.2.6.	Métricas de Pradenas .....	36
2.2.2.7.	Métricas de Abreu y Melo .....	36
2.2.2.8.	Métricas de Chindamber y Kemerer .....	38
2.2.2.9.	Métricas de Lorenz y Kid.....	41
2.2.2.10.	Métricas de Género.....	43
2.3.	Descripción del proceso de obtención de la propuesta .....	45
2.4.	Beneficios de la propuesta.....	46
2.5.	Roles y responsabilidades en el proceso de evaluación .....	46
2.6.	Conclusiones.....	47
Capítulo 3	.....	48
3.1.	Introducción.....	48
3.2.	Desarrollo del proceso de la aplicación de la propuesta.....	48

3.2.1.	Fase inicial .....	48
3.2.2.	Selección y ajuste .....	48
3.2.3.	Aplicación de métricas en proyectos de la universidad .....	49
3.2.3.1.	Aplicación de métrica en el proyecto Servicios Comunitarios .....	49
3.2.3.2.	Aplicación de métrica en el proyecto Akademos.....	52
3.2.4.	Evaluación de los resultados .....	54
3.2.5.	Principales deficiencias encontradas .....	55
3.2.6.	Refinamiento .....	55
3.2.7.	Definición de propuesta final .....	55
3.2.7.1.	Propuesta final .....	56
3.2.8.	Validación empírica de la propuesta .....	57
3.2.8.1.	Aplicación de propuesta final en Akademos.....	57
3.2.8.2.	Aplicación de propuesta final en Servicios Comunitarios .....	60
3.2.9.	Evaluación de los resultados .....	62
3.3.	Validación teórica .....	62
3.4.	Guía para la evaluación técnica de la propuesta .....	62
3.5.	Principales valoraciones y sugerencias de los expertos .....	67
3.6.	Conclusiones.....	68
Conclusiones Generales.....		69
Referencias Bibliográficas .....		71
Bibliografía Consultada.....		73
Glosario de Términos y Siglas .....		74
Anexos .....		78

## Índice de Tablas:

Tabla 1: Clasificación de las Bases de Datos.....	14
Tabla 2: Sistemas Gestores de BD más utilizados.....	15
Tabla 3: Clasificación de Métricas.....	18
Tabla 4: Métricas Propuestas.....	25
Tabla 5: Métricas de Moody y Shanks. ....	28
Tabla 6. Métricas de Moody .....	32
Tabla 7. Métricas de Piattini .....	35
Tabla 8: Métricas de Pradenas.....	36
Tabla 9. Métricas de Género .....	44
Tabla 10: Resumen de Modelos Conceptuales Tradicionales .....	44
Tabla 11: Tabla Resumen de Modelos Conceptuales Orientados a Objetos .....	45
Tabla 12: Prueba Servicios Comunitarios Conjunto 1. ....	49
Tabla 13 Prueba Servicios Comunitarios Conjunto 2. ....	51
Tabla 14. Prueba Akademos Conjunto 1. ....	52
Tabla 15. Prueba Akademos Conjunto 2. ....	54
Tabla 16. Propuesta Final.....	57
Tabla 17 Rango de Evaluación de Resultados .....	57
Tabla 18 Propuesta final en Akademos .....	58
Tabla 19. Evaluación de Akademos .....	59
Tabla 20. Propuesta final en Servicios Comunitarios .....	61
Tabla 21. Evaluación de Servicios Comunitarios .....	62
Tabla 22 Pesos Otorgados por los Expertos .....	64
Tabla 23. Cálculo de la Dispersión.....	65
Tabla 24. Cálculo de Índice de Aceptación .....	66
Tabla 25 Interpretación del Índice de Aceptación .....	67

## Índice de Figuras:

Fig 1. Ciclo de vida de un sistema informático.....	6
Fig 2. Concepto de métricas.....	17
Fig 3. Pregunta 1 de la encuesta.....	21
Fig 4. Pregunta 3 de la encuesta.....	21
Fig 5. Pregunta 4 de la encuesta.....	22
Fig 6. Pregunta 6 de la encuesta.....	22
Fig 7 Pregunta 7 de la encuesta.....	22
Fig 8. Pregunta 8 de la encuesta.....	23
Fig 9. Proceso de desarrollo de la propuesta de solución .....	46
Fig 10. Valoración de Expertos .....	68

# Introducción

En el desarrollo socio-económico actual la industria de software se ha convertido en un factor determinante. Lo que ha llamado la atención del mercado hacia esta industria han sido dos factores esenciales: la velocidad con que ha crecido y su alcance. El desarrollo de software se ha convertido en un tema crítico en la sociedad moderna, pues todos parecen necesitar mejores sistemas en menos tiempo y a menor costo. Al haber tanto interés, muchas personas empezaron a desarrollar diferentes tipos de aplicaciones informáticas y ahí nacieron las primeras grandes empresas.

Dentro del proceso de desarrollo de software la información ha sido siempre un recurso muy valorado. Desde las primeras civilizaciones se han archivado grandes cantidades de información en bibliotecas, monasterios o por determinadas personas. En esos lugares se atesoraban conocimientos científicos, de medicina, astronomía, matemática y física. Sin embargo, nunca como ahora, ha tenido tanto valor la información. La explosión de las tecnologías de redes, la aparición de Internet y todos los servicios que brinda, el acercamiento entre culturas y las necesidades de comunicación que estas generan, así como el fenómeno de la globalización; han hecho que en unos pocos años la información se haya convertido en un eslabón fundamental para el desarrollo de cualquier sociedad.

Millones de bits de información de todo tipo se generan y circulan cada día en todo el mundo e igual cantidad es consumida. Este enorme torrente requiere evidente organización y control, tanto para poder ofrecerla como administrarla por lo que uno de los principales objetivos del desarrollo de software es el procesamiento y gestión de la información, para esto el recurso más empleado son las BD. Estas constituyen el soporte fundamental de los datos de las empresas e instituciones en sus procesos de gestión, pues procesan gran cantidad de información valiosa y diversa, que se necesita organizar y tener disponible en el momento de tomar decisiones y brindar servicios.

En el mundo del software, los ingenieros han propuesto métricas para estimar productos, procesos y proyectos, enfocadas a diferentes aspectos en el desarrollo del software, sin embargo el uso de las métricas y el control de la Calidad de las BD han sido descuidadas y esto puede explicarse si se tiene en cuenta que, hasta hace poco, las BD se consideraban como algo secundario sin apenas repercusión en la complejidad total del sistema.

Sin embargo, las BD constituyen una parte esencial de todo software de gestión, pasando a ser el núcleo de los sistemas de información, debido a lo cual, resulta fundamental su medición. Esto pone en evidencia la importancia que tiene contar con métricas que permitan evaluar y controlar la calidad de las BD, así como poder brindar al cliente una garantía de la calidad del producto que está adquiriendo. (1) (2)

Diseñar métricas para medir los datos es importante debido a la relevancia que el tamaño de los datos y su naturaleza tienen en aspectos como el esfuerzo de desarrollo. Por tanto, medir datos puede ayudar a controlar y predecir aspectos del modelo de datos durante el proceso de desarrollo de software. (1)

Con el incremento de desarrolladores en distintos países y para distintas aplicaciones surgió una diversidad de estilos, por lo que varían mucho las características del producto final. Esto ha motivado que a escala mundial se esté generando un conjunto de modelos para estimar la calidad de los sistemas, los cuales responden a la necesidad de garantizar productos más eficientes en una competencia abierta y mundial.

A partir de la necesidad de informatizar el país y teniendo en cuenta los problemas existentes surgió la UCI con la misión de formar profesionales altamente calificados en la rama de la informática para producir software y brindar servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación.

La Universidad de las Ciencias Informáticas (UCI) es un centro educacional que a su vez está vinculado directamente con la producción donde se desarrollan múltiples proyectos con el fin de lograr la informatización de la sociedad cubana y la comercialización con otros países. Muchos de estos proyectos implementan BD a las cuales es necesario medir y certificar su calidad, lo que es difícil en este momento por lo que se puede definir la siguiente **Situación problémica**:

- La universidad no posee procesos bien definidos para la medición y certificación de la calidad de las BD relacionales.
- Carencia de métricas consistentes para la evaluación del diseño y la implementación de las BD relacionales.
- La falta de conocimientos y experiencia por parte de la mayoría de los evaluadores en el uso de métricas y normas de calidad trae consigo que se diseñen BD con errores que repercutan en el funcionamiento del sistema.

Luego de un profundo análisis de las necesidades reales existentes y con el fin de solucionar la situación anterior, se plantea el siguiente **problema científico**:

*La poca utilización, el desconocimiento y las deficiencias de las métricas dirigidas a la evaluación de la calidad del diseño y la implementación de BD relacionales provocan inconsistencias en el diseño de las mismas y mal funcionamientos de los sistemas al que están asociadas.*

Para alcanzar una solución satisfactoria que elimine total o parcialmente el problema planteado se necesita realizar un estudio exhaustivo del **objeto de estudio**: *El proceso de diseño e implementación de las BD Relacionales.*

En síntesis, el **campo de acción** que abarca la investigación es: *La medición de la calidad de los procesos de diseño e implementación de BD Relacionales en la UCI.*

Por tanto, el **objetivo general** de la investigación consiste en: *Diseñar un conjunto de métricas para la medición y evaluación del diseño e implementación de BD Relacionales.*

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

1. Realizar un estudio del estado del arte de los conceptos a utilizar.
2. Realizar la propuesta de métricas para la evaluación del diseño e implementación de las BD relacionales.
3. Aplicar las métricas obtenidas de forma piloto en algunos proyectos de la universidad y valorar los resultados.

Para dar cumplimiento al objetivo anteriormente expresado, se definió un conjunto de **tareas de investigación**:

1. Elaborar un estudio del estado del arte sobre los temas.
  - a. Ciclo de desarrollo de un software con sus etapas.
  - b. La medición de la calidad en un proyecto de software.
  - c. Las métricas y su evolución.
  - d. Métricas utilizadas para medir la correctitud de las BD.
  - e. El proceso de desarrollo de software de la UCI.
  - f. El diseño e implementación de las BD en los proyectos de la UCI.
2. Realizar una propuesta de métricas a utilizar en la validación del diseño e implementación de una base de datos relacional.
3. Realizar un muestreo de las métricas en un grupo de los proyectos productivos que se llevan a cabo en nuestra universidad.
4. Realizar un estudio del muestreo realizado y valorar los resultados.

**Idea a defender:**

*“Contando con un conjunto de métricas consistentes se podrá evaluar el proceso de diseño e implementación de una Base de Datos Relacional”.*

A lo largo de esta investigación se utilizarán un conjunto de métodos científicos que servirán de guía y facilitarán un mejor entendimiento de lo que está sucediendo, como es el caso de los métodos empíricos y los teóricos. A continuación se explica en detalles el por qué de su selección.

- 1- **Encuestas y entrevistas:** Se realizarán entrevistas a varios líderes de proyectos y diseñadores de BD con el objetivo de conocer las métricas de calidad que utilizan así como los principales parámetros a tener en cuenta para la obtención de BD con calidad.
- 2- **Histórico-lógico-tendencial:** Se llevará a cabo un análisis de la trayectoria hasta la actualidad del proceso de obtención de calidad en las BD y la utilización de métricas en el mismo.



- 3- **Analítico–sintético:** Se llevará a cabo un análisis de las teorías, documentos, permitiendo la extracción de los elementos más importantes que se relacionan con el objeto de estudio.
- 4- **Métodos matemáticos:** Como métodos matemáticos se usarán los **métodos estadísticos** para el análisis de las encuestas y la definición de las métricas.

Después de haberse cumplido las tareas planificadas los posibles resultados de la investigación serían:

1. Grupo de métricas para medir y evaluar el diseño e implementación de las BD relacionales.
2. Definición de calidad de una base de datos tomando en cuenta el diseño y la implementación de una base de datos relacional.
3. Mejora del proceso de diseño e implementación de las BD relacionales utilizando para ello las métricas definidas y que permitan valorar la calidad del mismo.

# Capítulo 1

## Fundamentación Teórica

### 1.1. Introducción

En la actualidad la calidad del software se ha convertido en un pilar de extrema importancia. En la universidad se está incursionando en la aplicación de métricas y procedimientos en aras de obtener software de mejor calidad con los que pueda insertarse en el mercado mundial. Por esta razón se ha realizado un análisis general sobre la situación actual de los procesos de obtención de calidad de los mismos específicamente en el desarrollo de las BD relacionales.

En el presente capítulo se exponen una serie de conceptos como: métricas, calidad, BD lo cual ayudará a una mejor comprensión de todo lo relacionado con los aspectos teóricos de nuestro tema de investigación. Además, brinda una pequeña síntesis de la situación actual de estos temas en el mundo.

### 1.2. Sistemas informáticos

Un sistema informático es un conjunto de elementos que hacen posible el tratamiento automático de la de información; es decir un conjunto de elementos interrelacionados entre sí y relacionados a su vez con el sistema global en que se encuentra que pretende conseguir unos fines determinados. Los elementos constitutivos de un sistema informático serán físicos, lógicos y humanos.

Conjunto de hardware, software y de un soporte humano. Un sistema informático típico emplea una computadora que usa dispositivos programables para capturar, almacenar y procesar datos. La computadora personal, junto con la persona que lo maneja y los periféricos que los envuelven, resultan de por sí un ejemplo de sistema informático. Incluso la computadora más sencilla se clasifica como un sistema informático, porque al menos dos componentes (hardware y software) tienen que trabajar unidos. (3) (4)

Un sistema de información está formado por los siguientes componentes:

- La base de datos (BD).
- El Sistema Gestor de Base Datos (SGBD).
- Los programas de aplicación.

- Los dispositivos físicos (ordenadores, dispositivos de almacenamiento).
- El personal que utiliza y que desarrolla el sistema.

La base de datos es un componente fundamental de un sistema de información. El ciclo de vida de un sistema de información está ligado al ciclo de vida del sistema de base de datos sobre el que se apoya. Al ciclo de vida de los sistemas de información también se le denomina ciclo de vida del desarrollo del software. Las principales etapas que lo compone son: planificación, recolección y análisis de los requisitos, diseño (incluyendo el diseño de la base de datos), creación de prototipos, implementación, prueba y mantenimiento. (4)

### 1.2.1. Ciclo de Vida de un Sistema Informático

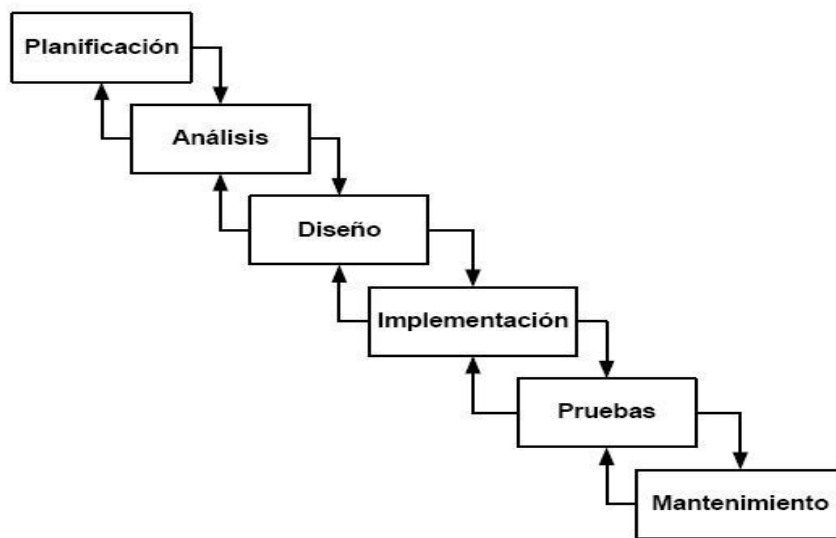


Fig 1. Ciclo de vida de un sistema informático.

#### **Identificación de problemas, oportunidades y objetivos.**

En la primera fase del ciclo de vida del desarrollo de sistemas informáticos el analista es el responsable de la identificación de problemas, oportunidades y objetivos. Esta etapa es crítica en el éxito del proyecto donde las personas involucradas son: usuarios, analista y administradores de sistemas que coordinan el proyecto. Las principales actividades que se realizan son: entrevistas a los administradores de los usuarios, centralización del conocimiento obtenido, estimación del alcance del proyecto y documentación de los resultados. La salida de esta fase es un estudio de factibilidad que contiene una definición del problema y los principales objetivos. Luego los administradores deben tomar una decisión para ver si continúan con el proyecto propuesto.

### **Determinación de los requerimientos de información.**

Entre las herramientas utilizadas para definir los requerimientos de información en el negocio se encuentran: muestreo e investigación de los datos relevantes, entrevistas, cuestionarios, el comportamiento de los tomadores de decisiones y su ambiente de oficina y hasta la elaboración de prototipos. En esta fase el analista está esforzándose por comprender qué información necesitan los usuarios para realizar su trabajo. Las personas involucradas en esta fase son los analistas y los usuarios.

### **Análisis de las necesidades del sistema.**

La siguiente fase que realiza el analista de sistemas involucra el análisis de las necesidades del sistema. Nuevamente, herramientas y técnicas especiales ayudan para que el analista haga las determinaciones de los requerimientos. Durante esta fase el analista de sistemas también examina las decisiones estructuradas que se hacen. Hay tres métodos principales para el análisis de decisiones estructurales: lenguaje estructurado, tablas de decisión y árboles de decisión.

### **Diseño del sistema recomendado.**

En esta fase del ciclo de vida del desarrollo de sistemas, el analista usa la información recolectada anteriormente para realizar el diseño lógico del sistema de información. El analista diseña procedimientos precisos para la captura de datos, a fin de que los datos que van a entrar al sistema de información sean correctos. Además, también proporciona entrada efectiva para el sistema de información mediante el uso de técnicas para el buen diseño de formas y pantallas.

### **Desarrollo y documentación del software.**

En la quinta fase del ciclo de vida del desarrollo de sistemas el analista trabaja con los programadores para desarrollar cualquier software original que se necesite, así como se relaciona con los usuarios para desarrollar documentación efectiva para el software, incluyendo manuales de procedimientos. La documentación le dice al usuario la manera de usar el software y también qué hacer si se suceden problemas con el software.

### **Pruebas y mantenimiento del sistema.**

Antes de que pueda ser usado, el sistema de información debe ser probado. Es mucho menos costoso encontrar problemas antes de que el sistema sea entregado a los usuarios. Algunas de las pruebas son realizadas por los programadores solos, y otras por los analistas de sistemas junto con los programadores. El mantenimiento del sistema y de su documentación comienza en esta fase y es efectuado rutinariamente a lo largo de la vida del sistema de información.

### **Implementación y evaluación del sistema.**

En esta fase del desarrollo del sistema el analista ayuda a implementar el sistema de información. Esto incluye el entrenamiento de los usuarios para que manejen el sistema. La

evaluación se muestra como parte de esta fase final de ciclo de vida del desarrollo del sistema. De hecho, la evaluación se realiza durante cada fase. (5)

### **1.2.1. Calidad en la producción de Sistemas Informáticos**

La calidad de los sistemas informáticos se ha convertido hoy en día en uno de los principales objetivos estratégicos de las organizaciones debido a que, cada vez más, dependen de los sistemas informáticos para su buen funcionamiento. En la evolución experimentada por la calidad en esta área se ha pasado de un tratamiento centrado fundamentalmente en la inspección y detección de errores en los programas, a una aproximación más sistemática, dada la importancia que ha adquirido la calidad en la ingeniería del software. En los últimos años, se han publicado diversos estándares y modelos que exponen los principios que se deben seguir para la mejora de la calidad de productos y procesos de software, entre ellos se encuentran: normas ISO 9000-3, ISO 9126, ISO 15504, modelos como CMM, CMMI, PSP, TSP. Además, trata aspectos muy importantes para conseguir sistemas de información de calidad, como pueden ser las métricas de software, la calidad de la información o la gestión del conocimiento. A lo largo de esta obra se ha combinado el rigor científico con la experiencia práctica, proporcionando una panorámica actual y completa sobre la problemática asociada a la calidad de los sistemas informáticos. (6)

#### **1.2.1.1. ¿Qué es Calidad?**

Según la Norma ISO 8402, se define calidad como la totalidad de las características de un producto o servicio que le confieren su aptitud para satisfacer unas necesidades expresadas o implícitas.

Se considera que calidad, resume las características, propiedades, cualidades y en general atributos propios de un producto, que determinan sobre este la ausencia de defectos y la conformidad de todo el personal que de una forma y otra se vinculan con él, (productores, clientes, usuarios, etc.); es además la asociación de varios conceptos que se mencionan a continuación:

**Sistema de Calidad:** Conjunto de las estructuras, responsabilidades, actividades, recursos y procedimientos de la organización de una empresa, que ésta establece para garantizar que lo que ofrece cumple con las especificaciones establecidas previamente por la empresa y el cliente, asegurando una calidad continua a lo largo del tiempo.

**Control de calidad:** El control está dirigido al cumplimiento de requisitos, no es más que un conjunto de actividades y técnicas operativas utilizadas para verificar los requerimientos relativos a la calidad del producto o servicio.

**Garantía de calidad:** Conjunto de acciones planificadas y sistemáticas necesarias para proporcionar la confianza adecuada de que un producto o servicio satisface los requerimientos

datos sobre calidad, o sea tiene como finalidad inspirar confianza en que se cumplirá el requisito pertinente.

**Gestión de la calidad:** La gestión de la calidad es un conjunto de actividades y medios necesarios para definir e implantar un sistema de la calidad, responsabilizarse de su control, aseguramiento y mejora continua.

### **1.2.1.2. ¿Qué es calidad de software?**

Se define la calidad de software como la ausencia de errores de funcionamiento, la adecuación a las necesidades del usuario, y el alcance de un desempeño apropiado (tiempo, volumen, espacio), además del cumplimiento de los estándares. Los objetivos que la calidad persigue son: La aceptación (utilización real por parte del usuario) y la mantenibilidad (posibilidad y facilidad de corrección, ajuste y modificación durante largo tiempo).

La calidad del software es definida como la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se esperan de todo software desarrollado profesionalmente.

Se considera que la calidad no es una simple característica del software que se aprecia en el producto final, sino el resultado de los esfuerzos de todos los miembros del equipo durante todas las etapas del proceso de desarrollo del software. Por lo general se tiene en cuenta la satisfacción del cliente, la entrega oportuna, la importancia de las mediciones y métricas, así como el desarrollo de un proceso bien definido. Uno de los elementos que permite dar garantía acerca de la calidad del software es la aplicación de métricas.

## **1.3. Bases de datos**

### **1.3.1. Definiciones**

Desde el punto de vista de la informática, la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos. (7)

Las BD, proporcionan unos datos organizados, en un entorno estático, según determinados criterios, y facilitan su exploración y consulta selectiva. Se pueden emplear en múltiples actividades como por ejemplo: seleccionar datos relevantes para resolver problemas, analizar y relacionar datos, extraer conclusiones, comprobar hipótesis. Una base de datos es una recopilación de datos o información relacionados entre sí, la misma está compuesta por filas que son los registros y por columnas que son los campos. (8)

Una base de datos es una colección de información que se encuentra agrupada o estructurada y organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite así como acceder a ello de forma rápida y segura.

### **1.3.2. Principales propiedades**

Una base de datos es un conjunto de datos que tiene las siguientes propiedades implícitas:

- Representa algún aspecto del mundo real, llamado minimundo o universo de discurso. Las modificaciones del minimundo se reflejan en la BD.
- Es un conjunto de datos lógicamente coherentes, con un cierto significado inherente. Una colección aleatoria de datos no puede considerarse propiamente una BD.
- Una BD se diseña, construye y puebla con datos para propósito específico. Está dirigida a un grupo de usuarios y tiene ciertas aplicaciones preconcebidas que interesan a distintos usuarios.

### **1.3.3. Características de las bases de datos**

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar. (9)

### **1.3.4. Metodologías de diseño de base de datos**

Solo con la tecnología no es suficiente para que los sistemas actuales de BD funcionen mejor que los de hace unos años. Vinculado a las tecnologías suelen asociarse unas metodologías que intentan sacar provecho de las primeras. Utilizar un sistema de gestión de BD relacional no es por sí solo una garantía de que la base de datos que se construya utilizándolo vaya a funcionar bien. De hecho, la simplicidad del modelo relacional y de algunos Sistemas Gestores de BD Relacionales (SGBDR), han llevado a profesionales y no profesionales del área informática a la creación de verdaderos desastres, causando casi más problemas de los que resuelven.

El diseño de BD es un proceso complejo para el cual es necesario tomar decisiones a distintos niveles. Para controlar esta complejidad se desglosa el problema en otros más pequeños, es decir, en subproblemas y se resuelven cada uno de estos problemas de forma independiente,

utilizando técnicas específicas. Así el diseño de una base de datos se descompone en diseño conceptual, diseño lógico y diseño físico.

Los requerimientos de usuario son el punto de partida para el diseño conceptual y el resultado es el esquema conceptual de la base de datos, cuyo objetivo es describir el contenido de información de la base de datos y no las estructuras de almacenamiento que se necesitarán para manejar dicha información.

Un esquema conceptual es una descripción de alto nivel de la estructura de la base de datos, independientemente del SGBD que se vaya a utilizar para manipularla. Un modelo conceptual es un lenguaje que se utiliza para describir esquemas conceptuales.

El diseño lógico parte del esquema conceptual y da como resultado un esquema lógico. Un esquema lógico es una descripción de la estructura de la base de datos en términos de las estructuras de datos que puede procesar un tipo de SGBD. Un modelo lógico es un lenguaje usado para especificar esquemas lógicos (modelo relacional, modelo de red). El diseño lógico depende del tipo de SGBD que se vaya a utilizar, no depende del producto concreto.

El diseño físico parte del esquema lógico y da como resultado un esquema físico. Un esquema físico es una descripción de la implementación de una base de datos en memoria secundaria: las estructuras de almacenamiento y los métodos utilizados para tener un acceso eficiente a los datos. Por ello, el diseño físico depende del SGBD concreto y el esquema físico se expresa mediante su lenguaje de definición de datos.

Es de vital importancia seguir adecuadamente la metodología de diseño de BD porque de su correcta aplicación dependen que se obtengan buenos resultados. Esta metodología proporciona una secuencia lógica de pasos que deben ser realizados en cada una de las etapas (diseño conceptual, diseño lógico, diseño físico). (24)

### **1.3.5. Arquitectura de las bases de datos**

Los sistemas de BD pueden ser estudiados desde 3 niveles distintos:

**Nivel Físico:** Es el nivel real de los datos almacenados. Es decir, cómo se almacenan los datos, ya sea en registros, o como sea. Este nivel es usado por muy pocas personas que deben estar capacitadas para ello. Este lleva asociada una representación de los datos, que es lo que se denomina esquema.

**Nivel Conceptual:** Es el correspondiente a una visión de la base de datos desde el punto de vista del mundo real. Es decir, se trata la entidad u objeto representado, sin importar como está representado o almacenado. Este nivel lleva asociado el esquema conceptual.

**Nivel Visión:** Son partes del esquema conceptual. El nivel visión es el encargado de dividir estas parcelas. Un ejemplo sería el caso del empleado que no tiene porqué tener acceso al



suelo de sus compañeros o de sus superiores. El esquema asociado a éste nivel es el esquema de visión.

Los 3 niveles vistos, componen lo que se conoce como arquitectura de base de datos a 3 niveles. Los SGBD basados en tres niveles permiten que cada usuario haga referencia exclusivamente a su propio esquema externo. Se realiza un proceso de transformación de un nivel a otro denominado correspondencia o transformación. (10)

### **1.3.6. Ciclo de vida de las aplicaciones de bases de datos**

Las etapas del ciclo de vida de una aplicación de BD son las siguientes:

- Planificación del proyecto.
- Definición del sistema.
- Recolección y análisis de los requisitos.
- Diseño de la base de datos.
- Selección del SGBD.
- Diseño de la aplicación.
- Prototipado.
- Implementación.
- Conversión y carga de datos.
- Prueba.
- Mantenimiento.

Estas etapas no son estrictamente secuenciales. De hecho, hay que repetir algunas de las etapas varias veces, haciendo lo que se conocen como ciclos de realimentación.

### **1.3.7. Ventajas en el uso de base de datos**

La utilización de BD como plataforma para el desarrollo de sistemas de aplicación en las organizaciones se ha incrementado notablemente en los últimos años, se debe a las ventajas que ofrece su utilización, algunas de las cuáles se comentarán a continuación:

**Globalización de la información:** permite a los diferentes usuarios considerar la información como un recurso corporativo que carece de dueños específicos.

**Eliminación de información inconsistente:** si existen dos o más archivos con la misma información, los cambios que se hagan a éstos deberán hacerse a todas las copias del archivo de facturas. Permite compartir información.

**Permite mantener la integridad en la información:** la integridad de la información es una de sus cualidades altamente deseable y tiene por objetivo que sólo se almacena la información correcta.

**Independencia de datos:** el concepto de independencia de datos es quizás el que más ha ayudado a la rápida proliferación del desarrollo de sistemas de BD. La independencia de datos implica un divorcio entre programas y datos. (9) (7)

#### **1.4. Sistemas gestores de bases de datos**

Un sistema gestor o manejador de BD es un conjunto de programas que permite a los usuarios crear y mantener una BD, por lo tanto, el SGBD es un software que facilita el proceso de definir, construir y manipular la BD para diversas aplicaciones. Pueden ser de propósito general o específico.

Existen distintos objetivos que deben cumplir los SGBD:

**Abstracción de la información.** Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos; da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.

**Independencia.** La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.

**Redundancia mínima.** Un buen diseño de una base de datos logrará evitar la aparición de información repetida o redundante. De entrada, lo ideal es lograr una redundancia nula; no obstante, en algunos casos la complejidad de los cálculos hace necesaria la aparición de redundancias.

**Consistencia.** En aquellos casos en los que no se ha logrado esta redundancia nula, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.

**Seguridad.** La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentre asegurada frente a usuarios malintencionados que intenten leer información privilegiada; frente a ataques que manipulen o destruyan la información o simplemente ante las torpezas de algún usuario autorizado pero despistado. Normalmente, los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios que permiten otorgar diversas categorías de permisos.

**Integridad.** Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados; es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados o cualquier otra circunstancia capaz de corromper la información almacenada.

**Respaldo y recuperación.** Los SGBD deben proporcionar una forma eficiente de realizar copias de seguridad de la información almacenada en ellos y de restaurar a partir de estas copias los datos que se hayan podido perder.

**Control de la concurrencia.** En la mayoría de entornos lo más habitual es que sean muchas las personas que acceden a una base de datos, bien para recuperar información, bien para almacenarla, es frecuente que dichos accesos se realicen de forma simultánea. Así pues, un SGBD debe controlar este acceso concurrente a la información, que podría derivar en inconsistencias.

**Tiempo de respuesta.** Lógicamente, es deseable minimizar el tiempo que el SGBD tarda en darnos la información solicitada y en almacenar los cambios realizados. (2)

### 1.4.1. Clasificación de los SGBD

Criterio	Clasificación de las bases de datos
<b>Según modelos conceptuales</b>	<ul style="list-style-type: none"> <li>• Relacional</li> <li>• Redes</li> <li>• Jerárquico</li> <li>• Orientado a Objetos</li> </ul>
<b>Según el número de usuarios</b>	<ul style="list-style-type: none"> <li>• Monousuario</li> <li>• Multiusuario</li> </ul>
<b>Según el número de sitios</b>	<ul style="list-style-type: none"> <li>• Centralizado.</li> <li>• Distribuido.</li> </ul>
<b>Según el uso de Software</b>	<ul style="list-style-type: none"> <li>• Homogéneo.</li> <li>• Heterogéneo.</li> </ul>
<b>Según el costo</b>	
<b>En cuanto al tipo de caminos de acceso</b>	
<b>En cuanto al propósito</b>	<ul style="list-style-type: none"> <li>• De propósito general.</li> <li>• De propósito específico</li> </ul>

Tabla 1: Clasificación de las bases de datos.

### 1.4.2. Características de los principales SGBD

En la actualidad existen numerosos SGBD entre los más utilizados mundialmente se encuentran *SQL Server* y *Oracle* ambos de código cerrado que ofrece buenas prestaciones a los usuarios pero tienen precios elevados, fundamentalmente *Oracle*.

En nuestro país existe la imperiosa necesidad de utilizar las tecnologías de software libre por cuestiones políticas, sociales y económicas. En algunas instituciones como la UCI se están utilizando en algunas facultades los SGBD *MySQL* y *PostgreSQL* ambos de código abierto. A continuación se ofrece una breve descripción de algunos de ellos con sus características fundamentales. (2)

Sistemas Gestores de Bases de Datos	Características
<b>SQL Server</b>	<ul style="list-style-type: none"> <li>• Soporte de transacciones.</li> <li>• Escalabilidad, estabilidad y seguridad.</li> <li>• Soporta procedimientos almacenados.</li> <li>• Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.</li> <li>• Permite trabajar en modo cliente-servidor, donde la información y los datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.</li> <li>• Permite administrar información de otros servidores de datos.</li> </ul>
<b>Oracle</b>	<p>Se considera a Oracle como uno de los sistemas de BD más completos, destacando su:</p> <ul style="list-style-type: none"> <li>• Soporte de transacciones.</li> <li>• Estabilidad.</li> <li>• Escalabilidad.</li> <li>• Soporte multiplataforma.</li> </ul> <p>Sus últimas versiones han sido certificadas para poder trabajar bajo Linux.</p>
<b>MySQL</b>	<ul style="list-style-type: none"> <li>• Sistema de gestión de base de datos relacional, multihilo y multiusuario.</li> <li>• Esquema de licenciamiento dual.</li> <li>• Soporte a multiplataforma.</li> <li>• Procedimientos almacenados.</li> <li>• <i>Triggers</i>.</li> <li>• Cursores.</li> <li>• Vistas actualizables.</li> <li>• Extremadamente rápido y fácil de personalizar.</li> <li>• Consume muy pocos recursos.</li> </ul> <p>Ofrece estabilidad y facilidad de despliegue.</p>
<b>PostgreSQL</b>	<ul style="list-style-type: none"> <li>• Alta concurrencia.</li> <li>• Amplia variedad de tipos nativos.</li> <li>• Claves ajenas también denominadas Llaves ajenas o Claves Foráneas (<i>foreign keys</i>).</li> <li>• Disparadores (<i>triggers</i>).</li> <li>• Vistas.</li> <li>• Integridad transaccional.</li> <li>• Herencia de tablas.</li> <li>• Tipos de datos y operaciones geométricas.</li> </ul> <p>Licencia BSD, se permite la utilización del código para ser comercializado.</p>

Tabla 2: Sistemas gestores de BD más utilizados.

### 1.4.3. Tendencias tecnológicas y del mercado

En la tecnología actual de SGBD relacionales se observan las siguientes tendencias:

- Los datos siguen estando centralizados en cuanto a modelado y administración mientras que el desarrollo de aplicaciones se descentraliza o se distribuye a través de Internet.

- Los usuarios finales acceden a las BD con mayor facilidad (existencia de productos en el mercado), incrementándose la seguridad en el acceso, sobre todo si es a través de servicios de Internet.
- El ordenador central pasa de soportar los procesos de manera centralizada a ser un servidor de datos. Los procesos pasan a máquinas clientes, o se distribuyen a los usuarios finales vía Internet/Intranet.
- Nuevas herramientas para gestión y soporte de las nuevas tendencias respecto al almacenamiento de datos (*Data Mining, Data Warehousing*) o respecto al procesamiento de los datos (*OLAP, OnLine Transaction Processing*).

En cuanto a los SGBD distribuidos, aunque no hay estándares definidos, existen en el mercado algunos productos que incorporan características de estos SGBD.

### **Los SGBD relacionales orientados a objetos**

Una vez ya desarrollados los SGBD orientados a objetos, surge la necesidad de un SGBD que, aparte de soportar la orientación a objetos, tenga la robustez y capacidad de procesamiento de información de los SGBD relacionales. De esta manera, aparecen los SGBD relacionales orientados a objetos, que todavía se encuentran en fase de expansión, tanto en su uso como en sus posibilidades de integrar distintos paradigmas pertenecientes a la orientación a objetos. Por otra parte, entre los temas en los que se está trabajando para la próxima versión de SQL (SQL3) se encuentra el soporte de BD orientadas a objetos. También existen varios fabricantes de SGBD relacionales que están incorporando, lentamente, capacidades de orientación a objetos en sus SGBD, abriendo así otra vía al desarrollo de SGBD orientados a objetos que parece muy prometedora en un futuro muy próximo.

## **1.5. Métricas en el proceso de análisis y diseño de sistemas informáticos**

### **1.5.1. Definiciones**

Las métricas son un buen medio para entender, monitorizar, controlar, predecir y probar el desarrollo software y los proyectos de mantenimiento (1).

Una métrica es un valor numérico o nominal asignado a características o atributos de un ente computado a partir de un conjunto de datos observables y consistentes con la intuición. Una métrica puede ser directa o indirecta, interna o externa, objetiva o subjetiva. (12)

Una métrica es una medida efectuada sobre algún aspecto del sistema en desarrollo o del proceso empleado que permite, previa comparación con unos valores (medidas) de referencia, obtener conclusiones sobre el aspecto medido con el fin de adoptar las decisiones necesarias.

La definición y aplicación de una métrica no es un objetivo en sí mismo sino un medio para controlar el desarrollo de un sistema de software.

Una métrica es un valor numérico o nominal asignado a características o atributo que permite comparar con los valores de referencia y obtener conclusiones sobre el aspecto medido con el fin de entender, monitorizar, controlar, predecir y probar el desarrollo de un proyecto.

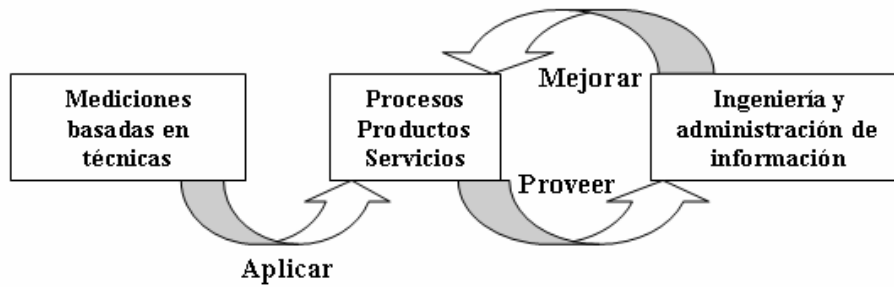


Fig 2. Concepto de métricas

### 1.5.2. Clasificación de las métricas

Existen innumerables métricas con propósitos diferentes que reflejan o describen la conducta del software, estas pueden medir entre otros aspectos la competencia, calidad, desempeño y la complejidad del software contribuyendo a establecer de una manera sistemática y objetiva una visión interna del trabajo mejorando así la calidad del producto. (13)

Clasificación de Métricas	Descripción	Ejemplos
<b>Métricas de complejidad</b>	Son todas las métricas de software que definen la medición de la complejidad	<ul style="list-style-type: none"> <li>• Volumen</li> <li>• Tamaño</li> <li>• Anidaciones</li> <li>• Costo (estimación)</li> <li>• Configuración</li> </ul>
<b>Métricas de calidad</b>	Son todas las métricas de software que definen de una u otra forma la calidad del software	<ul style="list-style-type: none"> <li>• Exactitud</li> <li>• Estructuración o modularidad</li> <li>• Pruebas</li> <li>• Mantenimiento reusabilidad</li> </ul>
<b>Métricas de competencia</b>	Son todas las métricas que intentan valorar o medir las actividades de productividad de los programadores o practicantes	<ul style="list-style-type: none"> <li>• Certeza</li> <li>• Rapidez</li> <li>• Eficiencia</li> <li>• Competencia.</li> </ul>

<b>Métricas de desempeño</b>	Corresponden a las métricas que miden la conducta de módulos y sistemas de un software, bajo la supervisión del sistema operativo o hardware	<ul style="list-style-type: none"> <li>• Eficiencia de ejecución</li> <li>• Tiempo</li> <li>• Almacenamiento</li> <li>• Complejidad de algoritmos computacionales.</li> </ul>
<b>Métricas estilizadas</b>	Son las métricas de experimentación y de preferencia	<ul style="list-style-type: none"> <li>• Estilo de código</li> <li>• Las convenciones denominando de datos</li> <li>• Las limitaciones</li> </ul>
<b>Variedad de métricas</b>		<ul style="list-style-type: none"> <li>• Portabilidad</li> <li>• Facilidad de localización</li> <li>• Consistencia</li> </ul>

Tabla 3: Clasificación de Métricas.

### 1.5.3. Características de las métricas

Para que sea útil en el contexto del mundo real, una métrica del software debe ser objetiva, simple y calculable, consistente en el empleo de unidades y tamaños, persuasiva, además debería ser independiente del lenguaje de programación y proporcionar una realimentación eficaz para el desarrollador de software.

¿Por qué asegurarnos de que las métricas cumplen estas condiciones?

Las métricas deben ser un instrumento que ayude a mejorar el proceso, producto o proyecto de software, no tiene mucho sentido aplicar métricas que lejos de ayudar a los desarrolladores constituyan un problema; bien por ser demasiado complejas, porque no se entiendan correctamente los objetivos que persiguen o porque arrojen resultados imprecisos que no puedan ser interpretados por los ingenieros de software.

Es importante entonces que una métrica pueda obtenerse fácilmente, que se entienda por qué y para qué se utiliza, que los cálculos no produzcan resultados ambiguos o en los que existan extrañas combinaciones de unidades, y que la interpretación de valores obtenidos esté acorde a las nociones intuitivas del ingeniero de software. Por otra parte, las métricas no deben ser específicas para ningún lenguaje de programación o metodología de desarrollo. (13)

### 1.5.4. ¿Por qué es importante medir?

Hay cuatro razones para medir los procesos del software, los productos y los recursos:

**Caracterizar:** Para comprender mejor los procesos, los productos, los recursos y los entornos y para establecer las líneas bases para las comparaciones con evaluaciones futuras.

**Evaluar:** Para determinar el estado con respecto al diseño. Las medidas permiten conocer cuándo los proyectos y procesos están perdiendo la pista, de modo que puedan ponerse bajo control. Además para valorar si se cumplen o no los objetivos de calidad trazados y para evaluar el impacto de la tecnología y las mejoras en los productos y procesos.

**Predecir:** Para poder planificar. Los valores que se observan para algunos atributos pueden ser utilizados para predecir otros, lo que contribuye a establecer objetivos alcanzables para el coste, planificación y calidad, de manera que se puedan aplicar los recursos apropiados, además permite analizar los riesgos y realizar intercambios diseño coste.

**Mejorar:** Se mide para mejorar cuando se recoge la información cuantitativa que ayuda a identificar obstáculos, problemas de raíz, ineficiencias y otras oportunidades para mejorar la calidad del producto y el rendimiento del proceso.

Las métricas de software ayudan a los desarrolladores a valorar el trabajo desarrollado, proveen la información necesaria para la toma de decisiones técnicas, proporcionan datos objetivos que pueden ser usados en la planificación de futuros proyectos, ayudan a la evaluación de los modelos de análisis y de diseño, y a la formulación de casos de prueba, valoran la productividad de los desarrolladores, ayudan a evaluar la calidad de los productos o sistemas que se construyen así como también a entender que ocurre durante el desarrollo y el mantenimiento. (25)

### **1.5.5. Proceso de medición**

Todo proceso de medición del software tiene como objetivo fundamental satisfacer necesidades de información a partir de las cuales se deben identificar las entidades y los atributos que deben ser medidos.

El proceso de medición, se caracteriza en cinco actividades:

**Formulación:** Obtención de medidas y métricas del software apropiadas para la presentación del software en cuestión.

**Colección:** Mecanismo empleado para acumular datos necesarios para obtener las métricas formuladas.

**Análisis:** Cálculo de las métricas y la aplicación de herramientas matemáticas.

**Interpretación:** La evaluación de los resultados de las métricas en un esfuerzo por conseguir una visión interna de la calidad de la presentación.

**Realimentación:** Recomendaciones obtenidas de la interpretación de métricas y técnicas transmitidas al equipo de desarrollo de software. (14)

### **1.5.6. Métricas y calidad**

El principal objetivo de los ingenieros del software es producir un sistema, aplicación o producto competitivo en el mercado mundial, para lo cual emplean métodos y herramientas efectivas dentro del contexto de un proceso maduro de desarrollo del software y además deben desarrollar mediciones que den como resultado sistemas de alta calidad. Para obtener esta evaluación, el ingeniero debe utilizar medidas técnicas, que evalúan la calidad con objetividad, no con subjetividad.



A pesar del avance en el desarrollo de software y las tecnologías, con el paso de los años los atributos que proporcionan una indicación de la calidad del software siguen siendo los mismos. La norma ISO 9126 es un ejemplo de ello, muchas características y subcaracterísticas definidas en la misma hacen referencia a la operación, transición y revisión del software y aunque no las dividen en estos tres grupos, señalan entre otras cosas la necesidad de lograr que el software opere correctamente y con el grado de exactitud requerido, que los usuarios sean capaces de entenderlo y usarlo, es decir, que sea amigable con quienes interactúen con él, que sea capaz de responder correctamente ante fallos o cambios del entorno.(15)

### **1.5.7. Las métricas en la industria cubana del software**

La Industria Cubana del Software (ICSW) está llamada a convertirse en una significativa fuente de ingresos nacional, como resultado del correcto aprovechamiento de las ventajas del considerable capital humano disponible. La preparación de estos recursos humanos especializados para las Tecnologías de la Informática y las Comunicaciones (TIC) es un factor clave de la estrategia cubana de informatización. La Universidad de las Ciencias Informáticas y el sistema de empresas cubanas vinculadas a este trabajo jugarán un papel importante en el desarrollo de la ICSW, y en la materialización de los proyectos asociados al programa cubano de informatización.

Existen empresas que tienen definidas métricas basadas fundamentalmente en la norma ISO 9126, pero no se están aplicando en todos los proyectos pues depende en gran medida de la decisión de los líderes y desarrolladores. Se guardan registros históricos que se utilizan para realizar estimaciones, sin embargo en otras empresas no se aplican métricas en los procesos de desarrollo de software pues se considera que es necesario lograr una mayor organización y definición de los procesos de la empresa antes de comenzar a utilizar un conjunto de métricas o al menos una base sólida que justifique la aplicación de las mismas, sin que esto constituya una pérdida de tiempo. (16)

### **1.5.8. Situación actual de la utilización de métricas de software en la UCI**

La UCI no está aislada del problema de las mediciones. Hoy la mayoría de los desarrolladores de software: líderes de proyectos, programadores, diseñadores, no planean ni registran su trabajo y no miden y administran la calidad de los productos; a pesar, de que las mediciones pueden ayudarlos a lograr que su trabajo cada día sea mejor y con una mayor calidad. El desconocimiento de las métricas es un factor que pesa en la evaluación del software, lo cual hace que establezcan un rechazo hacia ellas.

La situación que dio origen a esta investigación fue el tema de la calidad de las BD relacionales la cual ha sido descuidada por la mayoría de los desarrolladores, siendo la inexistencia de

métricas para la evaluación de las mismas un problema que atenta contra la calidad de los productos de la universidad.

### 1.5.9. Descripción de la encuesta

Para el desarrollo de la propuesta fue necesario aplicar una encuesta, con el propósito de definir el nivel de importancia que poseen los factores de calidad de las BD, a través del criterio de diversos especialistas en este tema. En la encuesta se utilizó una combinación de preguntas abiertas y cerradas, con el objetivo de que los encuestados contaran con una guía para seguir la línea de la investigación. Esta dirigida fundamentalmente al jefe de proyecto y al diseñador de BD. (Ver Anexo #1)

### 1.5.10. Análisis de la encuesta

La encuesta se le aplicó a 9 especialistas de los proyectos productivos de la facultad 1, entre ellos 4 jefes de proyecto y 5 diseñadores de BD.

Una vez procesada se obtuvieron diversos resultados, en dependencia de la importancia que cada encuestado le concedió a la medición de la calidad de las BD, así como al establecimiento de métricas que evalúen la calidad de un diseño de BD.

Pregunta #1: ¿Tienen definido algún plan de pruebas para el diseño de las BD?

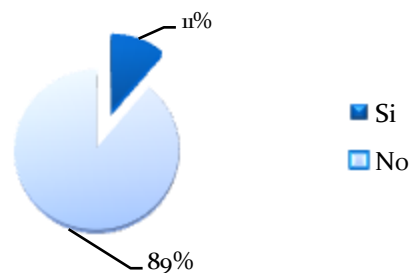


Fig 3. Pregunta 1 de la encuesta

Pregunta #2: ¿Se utiliza algún estándar para el plan de pruebas?

El 100 % de los encuestados respondieron que no a esta pregunta.

Pregunta #3: ¿Se utiliza algún estándar para el diseño de las BD?

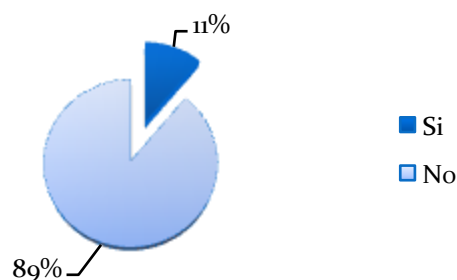


Fig 4. Pregunta 3 de la encuesta

Pregunta #5: ¿Conoce alguna métrica para medir la calidad de un diseño de BD?

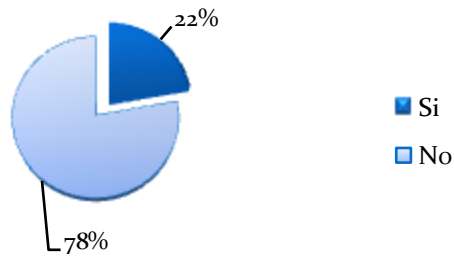


Fig 5. Pregunta 4 de la encuesta

Pregunta #6: ¿Qué herramienta de modelado se utilizan en el proyecto para modelar las BD?  
¿Qué SGBD?

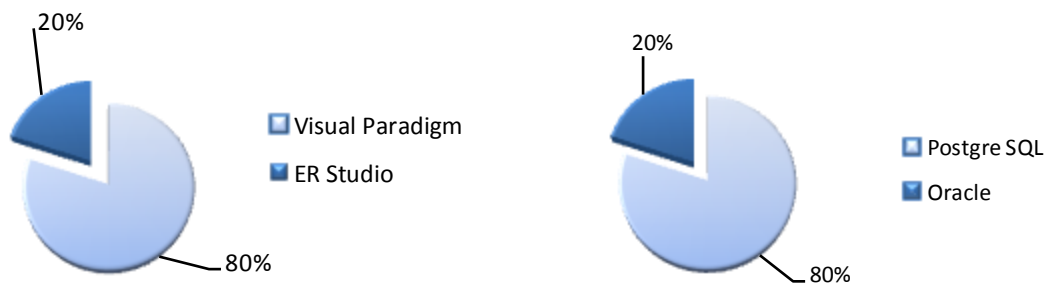


Fig 6. Pregunta 6 de la encuesta

Pregunta #7: ¿Se aplica alguna métrica para la medición de la calidad del producto obtenido o al proceso de la obtención del mismo?

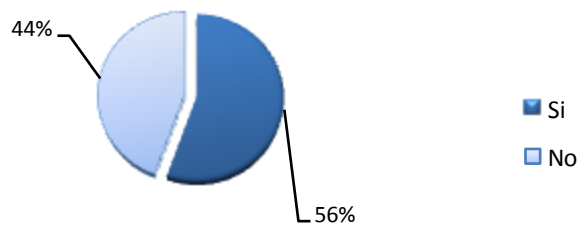


Fig 7 Pregunta 7 de la encuesta

Pregunta #8: ¿Qué factores cree que influyen en la calidad de una base de datos?

Se le aplicó la encuesta a 5 proyectos con el objetivo de determinar los factores con más importancia para los diseñadores según su experiencia en el diseño de BD.

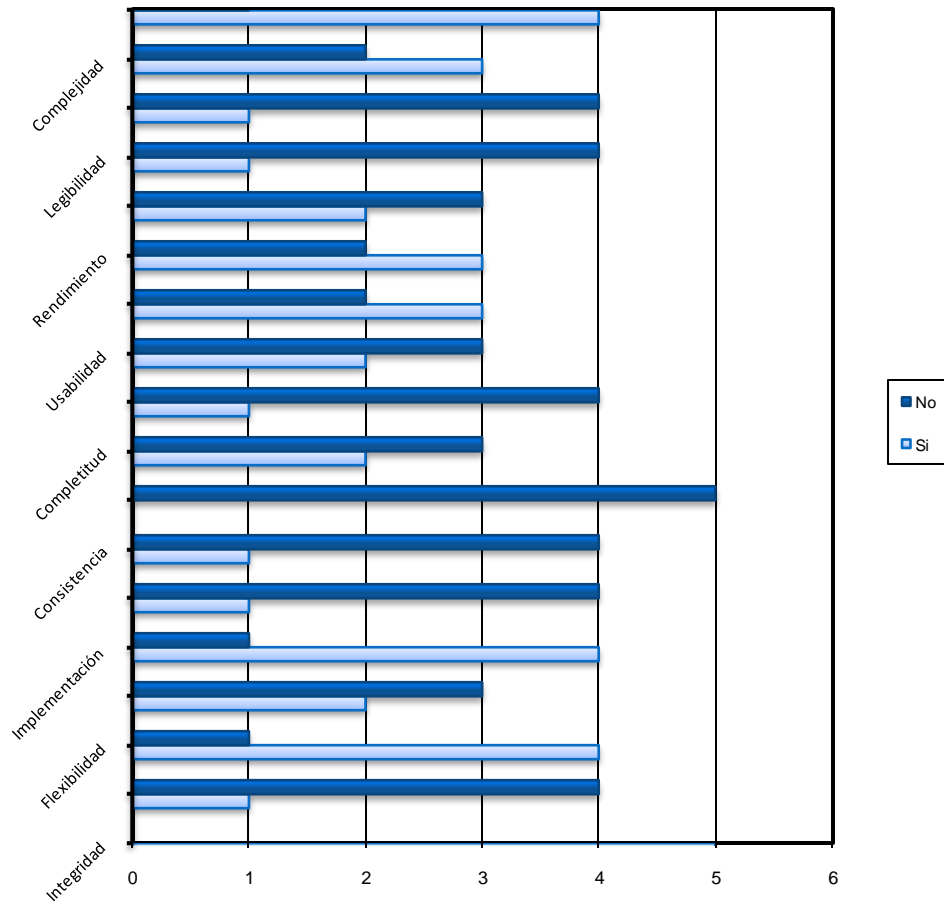


Fig 8. Pregunta 8 de la encuesta

### 1.5.11. Resultados de la investigación

Después de un análisis de las encuestas realizadas se detectaron los siguientes problemas:

- La UCI no tiene definido un procedimiento para la evaluación de las base de datos relacionales.
- El personal disponible no conoce sobre la existencia de métricas dirigidas a las BD.
- No existe un procedimiento definido que estandarice el proceso de diseño de las BD.
- No existen una cultura de aplicación de métricas de software.
- Algunos desarrolladores no le dan importancia a la aplicación de métricas para BD pues creen que con lo que se realiza actualmente se garantiza la calidad de las mismas.

## 1.6. Métricas y bases de datos

Las bases de datos suponen una parte fundamental de los sistemas de información. Aún cuando la etapa de modelado de datos únicamente representa una pequeña proporción del esfuerzo total del desarrollo de sistemas, probablemente el impacto sobre el resultado final es mayor que el de cualquier otra etapa, por lo que es importante el control de la calidad de los procesos de diseño e implementación de las BD.

Una forma de realizar este control es mediante la utilización de métricas sometidas a procesos de verificación formal y de validación empírica encaminadas específicamente a las BD, campo bastante descuidado dentro de la ingeniería del software.

### 1.6.1. Características de un esquema conceptual con calidad

**Legibilidad:** Se refiere a la estética del esquema. Un esquema posee esta cualidad cuando puede leerse con facilidad, reconociendo tipos de entidad, tipos de interrelación y atributos, deben existir mínimos cruces de líneas siendo estas horizontales y verticales, que deje en sus debidos niveles las jerarquías, agregando además que si existen estructuras simétricas estas deben aparecer como tales. Se recalca que se trata de una fácil lectura no incluyendo la sencillez en la “comprensión” del esquema.

**Compleción:** Se trata del esquema que incluye la representación total del mundo o la realidad diseñada, esto a través de los requerimientos impuestos y restricciones, es decir, que exista una correspondencia uno a uno de los requerimientos con el esquema y también viceversa, además no deben existir requerimientos no formulados (inexistentes).

**Corrección:** Un esquema cumple con esta característica cuando usa correctamente los conceptos y definiciones del lenguaje ya sea sintáctica como semánticamente.

**Minimalidad:** La característica de minimalidad se refiere a la existencia de información redundante o duplicada en el esquema, es decir, este no es mínimo si existe redundancia. Dicho de otra forma, en un esquema mínimo no se puede borrar del esquema un elemento sin perder alguna información.

**Expresividad:** La expresividad se enfoca a la naturalidad con que se expresan los requerimientos. Se diferencia de la legibilidad en que aquí el esquema es de fácil comprensión, pero desde el punto de vista de lo que representa.

**Autoexplicación:** Un esquema se autoexplica si todos los requerimientos pueden ser modelados a través de los conceptos del modelo entidad-relación (MER) sin la ayuda de otros formalismos (ej.: lógica, lenguaje natural). Se puede tener acceso al mundo modelado solamente a través del esquema.

**Extensibilidad:** Se refiere a la capacidad de un esquema para poder tolerar cambios en los requerimientos. Se consideran extensibles aquellos esquemas que pueden descomponerse en

módulos o partes. Un esquema "modular", en el sentido recién expuesto, es más escalable, pudiendo incorporarse nuevos elementos alterando levemente los existentes.

**Consistencia:** Un esquema de BD es consistente, si existe una instanciación permitida (no vacía) para cada elemento del esquema. (17)

### 1.6.2. Tendencias y Evolución

A lo largo de los años, algunos autores han propuesto métricas dirigidas a medir la calidad de los diseños de BD relacionales, que a pesar de no ser lo suficientemente consistentes, sirven de base para el futuro desarrollo. (18)

<b>Autores</b>	<b>Propiedades</b>
<b>Batini (1992)</b>	compleción, corrección, minimalidad.
<b>Reingruber M y Gregori W (1994)</b>	corrección conceptual, corrección sintáctica.
<b>Kesh (1995)</b>	validez, consistencia, compleción
<b>Boman (1997)</b>	facilidad de comprensión, correctitud semántica.
<b>Moody (1998)</b>	integración, implementabilidad, comprensibilidad.
<b>Piatini (2000)</b>	número de atributos(NA), número de entidades (NE)

Tabla 4: Métricas Propuestas.

### 1.6.3. Deficiencias de las métricas de calidad para bases de datos

A modo de resumen las principales deficiencias que se detectaron en las métricas estudiadas fueron las siguientes:

- Las definiciones no son precisas, a veces se solapan entre sí con objetivos no realistas.
- Definen propiedades deseables y algunas son imprecisas y ambiguas.
- Proponen algunos criterios que no necesariamente tienen relación con los objetivos que se persiguen.
- No existe un criterio general aceptado, es decir un estándar, todos abundan distintos criterios de calidad propios de cada autor.
- No han sido validadas ni teórica ni empíricamente además de no aportar ninguna herramienta automática.
- Tienen medidas objetivas y estimaciones subjetivas. (18)

## 1.7. Conclusiones

En el capítulo se abordan los principales temas asociados a nuestro objeto de estudio, así como los aspectos necesarios para definir la propuesta de solución, en este caso la creación de un conjunto de métricas para la evaluación de la calidad de BD relacionales. Además se desarrolló un estudio del estado actual de las métricas para BD lo cual da una visión de los temas principales en los que debemos trabajar para darle solución a las deficiencias existentes.

Como resultado de la investigación realizada se obtuvo los conocimientos teóricos necesarios para la continuación el desarrollo de la investigación. Se identificaron las principales deficiencias de las métricas dirigidas a la evaluación de las BD relacionales con lo cual se ratificó la importancia que tiene la medición y evaluación de las BD.

# Capítulo 2

## Propuesta de Solución

### 2.1. Introducción

En el presente capítulo se realizará un análisis de las métricas dirigidas a la evaluación de las BD relacionales que existen hasta el momento, además de una breve descripción del proceso de desarrollo de la propuesta de solución así como sus principales fases. Se hará un análisis matemático de los resultados obtenidos a partir de las encuestas realizadas a los líderes de proyecto así como a los diseñadores de BD con el objetivo de definir los principales parámetros para BD con calidad.

### 2.2. Análisis de las métricas para modelos conceptuales

#### 2.2.1. Clasificación

Los modelos conceptuales pueden clasificarse en dos grandes grupos, los tradicionales y los orientados a objetos:

- Los modelos conceptuales tradicionales, como el de entidad-relación desarrollado en 1976 por Chen, y modificado posteriormente por otros autores, todavía pueden describir fácilmente los requisitos de datos de un sistema de información con independencia de criterio de la gestión y organización de los datos.
- Los modelos conceptuales orientados a objetos representan, además de los datos, el comportamiento y funcionalidad del sistema de información, mediante diagramas de clases, de actividad, de transición de estados. (19)



## 2.2.2. Caracterización

### 2.2.2.1. Propuesta de Moody y Shanks

<b>Nombre</b>	Métricas de Moody y Shanks. (19)
<b>Autores</b>	Graeme G. Shanks y Daniel L. Moody.
<b>Año</b>	1994
<b>Clasificación</b>	Métricas para modelos conceptuales tradicionales.
<b>Fuente</b>	Calidad y Medición de Sistemas de Información por Grupo Alarcos 2008/2009.
<b>Descripción</b>	Este modelo se basa en un conjunto de métricas, algunas objetivas y otras subjetivas, para evaluar algunos factores de calidad de los modelos de datos.
<b>Complejidad</b>	Baja.

#### Métricas:

Factor de calidad	Métricas
Compleción	Nº de elementos del modelo de datos que no corresponden con requisitos de usuario Nº de requisitos de usuario no representados en el modelo de datos Nº de elementos de datos que corresponden a requisitos de usuario pero definidos de forma inexacta. Nº de inconsistencias con el modelo de procesos
Integridad	Nº de reglas del negocio que no se hacen cumplir por el modelo de datos Nº de restricciones de integridad incluidas en el modelo de datos que no corresponden a políticas del negocio
Flexibilidad	Nº de elementos en el modelo que están sujetos a cambios en el futuro costes estimados de los cambios Importancia estratégica de los cambios
Comprensibilidad	Valoración de los usuarios sobre la comprensibilidad del modelo Capacidad de los usuarios de interpretar el modelo correctamente Valoración de los desarrolladores de aplicaciones sobre la comprensibilidad del modelo
Corrección	Nº de violaciones de las convenciones de modelado de datos Nº de violaciones a las formas normales Nº de instancias de redundancia en el modelo
Simplicidad	Nº de entidades Nº de entidades e interrelaciones Nº de constructores (aNE + bNR + cNA)
Integración	Nº de conflictos con el modelo de datos corporativo Nº de conflictos con los sistemas existentes Valoración de los representantes de todas las áreas de negocio
Implementabilidad	Valoración de riesgo técnico Valoración de riesgo de planificación Estimación del coste de desarrollo Nº de elementos físicos incluidos en el modelo de datos

Tabla 5: Métricas de Moody y Shanks.

**Significado de los valores:**

Los valores obtenidos constituyen indicadores para realizar comparaciones.

**2.2.2.2. Métricas de Kesh**

<b>Nombre</b>	Métricas Q.
<b>Autores</b>	Kesh.
<b>Año</b>	1995
<b>Clasificación</b>	Métricas para modelos conceptuales tradicionales.
<b>Fuente</b>	Calidad de software por Jorge Salamanca Escorial.
<b>Descripción</b>	El profesor Kesh publicó en 1995 el método que había desarrollado para el aseguramiento de la calidad de las BD. Este método se basa en que la calidad en estos modelos de datos se determina por dos tipos de componentes los ontológicos y los de comportamiento.
<b>Complejidad</b>	Alta

**El método se compone de tres pasos:**

**Cálculo del valor de cada uno de los componentes ontológicos:** Se calcula individualmente el valor de los componentes estructurales (las relaciones entre los elementos que forman el modelo: adecuación al problema:  $o_1$ , validez:  $o_2$ , consistencia:  $o_3$  y concisión  $o_4$ ) y de los componentes de contenido (los atributos de las entidades: completitud:  $o_5$ , cohesión:  $o_6$  y validez:  $o_7$ )

**Cálculo de los valores de los componentes de comportamiento:** Este cálculo se hace a partir de los valores de los componentes ontológicos relevantes para cada uno de los componentes de comportamiento. Los componentes de comportamiento a tener en cuenta son: facilidad de uso desde el punto de vista del usuario:  $s_1$ , usabilidad desde el punto de vista del diseñador:  $s_2$ , facilidad de mantenimiento:  $s_3$ , precisión:  $s_4$  y rendimiento:  $s_5$ .

**Cálculo de la calidad del modelo:** Este cálculo se hace a partir de los valores de los componentes de comportamiento de acuerdo con la fórmula:  $Q = \sum w_i s_i$  (con  $i$  de 1 a 5) donde  $w_i$  son los pesos de los factores de comportamiento y  $s_i$  los valores de dichos factores. Los pesos son determinados por la organización en función de la importancia que tengan para la misma.

Las formulas para el cálculo de  $s_i$  son las siguientes:

$$s_1 = (o_1 + o_3 + o_4 + o_5) / 4$$

$$s_2 = (o_2 + o_3 + o_5 + o_6 + o_7) / 5$$

$$s_3 = (o_2 + o_4 + o_6) / 3$$

$$s_4 = (o_3 + o_5) / 2$$

$$s_5 = (o_4 + o_5) / 2$$

Los valores de los factores ontológicos son, en algunos casos, estimados por los usuarios, y en otros calculados mediante fórmulas.

Los procedimientos son los siguientes:

**Adecuación del modelo al problema ( $o_1$ ):** Valor entre 1 y 5, determinado mediante entrevista con los usuarios.

**Validez del modelo ( $o_2$ ):** Valor entre 1 y 5, obtenido mediante entrevistas a un equipo técnico que no esté involucrado en el proyecto.

**Consistencia del modelo ( $o_3$ ):** Se calcula mediante la fórmula:  $o_3 = 5 - D1$  estando  $D1$  basado en el ratio  $R = (\text{número de inconsistencias}) / 4n$ , donde  $n$  es el número de relaciones en el modelo ( $4n$  representa el número de implicaciones)

**Concisión del modelo ( $o_4$ ):** Si un modelo tiene  $n$  entidades, el número mínimo de relaciones es  $n-1$ . A un modelo con  $(n-1)$  entidades se le atribuye un  $o_4$  de 5. El valor de 0 constituye el peor de los casos, cuando todas las entidades están relacionadas entre sí. En los demás casos el valor (entre 0 y 5) se obtiene, para un modelo con  $n1$  relaciones, mediante una fórmula específica:

$$o_4 = M((Cn2 - n1) / (Cn2 - (n-1)))$$

**Compleción del contenido ( $o_5$ ):** Se compara el modelo con la lista de consultas e informes que se desean obtener de la BD y por cada fallo que se observe se resta de 5 una cantidad proporcional a la importancia de la consulta o informe.

**Cohesión del contenido ( $o_6$ ):** La cohesión para cada entidad es el tamaño del identificador primario. Si éste está formado por un solo atributo, la cohesión es máxima y, por lo tanto,  $o_{6i}$  ( $i$  es el número de la entidad) es 5. Si el identificador primario lo constituyen todos los atributos de la entidad,  $o_{6i} = 0$ . Si  $ne$  es el número de atributos de la entidad y  $np$  es el número de atributos del identificador primario, entonces:

$$o_{6i} = M((ne - np) / (ne - 1))$$

Así el valor total de cohesión para todo el modelo será:  $o_6 = \sum o_{6i} / N$ .

**Validez del contenido ( $o_7$ ):** Si todos los atributos para todas las entidades son válidos  $o_7$  vale 5 ( $M=5$ ). El valor 0 se le asigna si todos los atributos **ne** se consideran no válidos. Si  $n_i$  es el número de entidades no válidas, entonces  $o_7$  se calcula como:  $o_7 = M(1 - n_i / \sum ne)$

El modelo está poco experimentado, por eso se necesita mucha interacción entre los diseñadores y los usuarios para su retroalimentación.

El propio Kesh considera que el valor de Q no es una estimación precisa, sino un indicador de la calidad del modelo y que, por consiguiente, habría que seguir trabajando sobre éste.

**Complejidad:** Alta. (19)

### 2.2.2.3. Métricas de Moody

<b>Nombre</b>	Métricas de Moody.
<b>Autores</b>	Daniel L. Moody.
<b>Año</b>	1998
<b>Clasificación</b>	Métricas para modelos conceptuales tradicionales.
<b>Fuente</b>	Calidad de software por Jorge Salamanca Escorial.
<b>Descripción</b>	Este modelo se basa en un conjunto de métricas, algunas objetivas y otras subjetivas, para evaluar algunos factores de calidad de los modelos de datos.
<b>Complejidad</b>	Baja

#### Métricas:

<b>Compleción</b>	<ul style="list-style-type: none"> <li>• Número de elementos del modelo de datos que no corresponden con los requisitos de usuario.</li> <li>• Número de elementos del modelo de datos que corresponden con los requisitos de usuario, pero definidos incorrectamente.</li> <li>• Número de requisitos del usuario no representados en el modelo.</li> <li>• Número de inconsistencias con el modelo de procesos.</li> </ul>
<b>Integridad</b>	<ul style="list-style-type: none"> <li>• Número de restricciones de integridad incluidas en el modelo que no corresponden a políticas de negocio.</li> <li>• Número de reglas del negocio que no se cumplen por el modelo de datos.</li> </ul>
<b>Flexibilidad</b>	<ul style="list-style-type: none"> <li>• Costes estimados de los cambios.</li> <li>• Importancia estratégica de los cambios.</li> <li>• Número de elementos del modelo que en el futuro estarán sometidos a cambios.</li> </ul>

<b>Corrección</b>	<ul style="list-style-type: none"> <li>• Número de violaciones a las formas normales.</li> <li>• Número de violaciones a las convenciones de modelos de datos.</li> <li>• Número de instancias de redundancias en el modelo.</li> </ul>
<b>Simplicidad</b>	<ul style="list-style-type: none"> <li>• Número de entidades.</li> <li>• Número de relaciones.</li> <li>• Número de constructores.</li> </ul>
<b>Integración</b>	<ul style="list-style-type: none"> <li>• Número de conflictos con los sistemas existentes.</li> <li>• Número de conflictos con el modelo de datos corporativo.</li> <li>• Valoración de los representantes de todas las áreas del negocio.</li> </ul>
<b>Implementabilidad</b>	<ul style="list-style-type: none"> <li>• Valoración de riesgo técnico.</li> <li>• Valoración de riesgo de planificación.</li> <li>• Estimación del coste del desarrollo.</li> <li>• Número de elementos físicos del modelo de datos.</li> </ul>
<b>Comprensibilidad</b>	<ul style="list-style-type: none"> <li>• Valoración de los usuarios sobre la comprensibilidad del modelo.</li> <li>• Capacidad de los usuarios de interpretar el modelo correctamente.</li> <li>• Valoración de los desarrolladores sobre la comprensibilidad del modelo.</li> </ul>

Tabla 6. Métricas de Moody

Moody propuso que investigadores y profesionales trabajen conjuntamente para demostrar la validez de estas métricas. Este no ha sido validado ni teórico ni prácticamente, no aporta herramientas, tiene medidas objetivas y estimaciones subjetivas, y solo tiene en cuenta algunos factores de calidad para modelos. (19)

#### 2.2.2.4. Métricas de Varas y Pradenas

<b>Nombre</b>	Métricas de Varas y Pradenas
<b>Autores</b>	Marcela Varas y Jaime Pradenas
<b>Año</b>	2000
<b>Clasificación</b>	Métricas para modelos conceptuales tradicionales.
<b>Fuente</b>	Hacia la Definición de Métricas de Calidad para Esquemas Conceptuales de BD
<b>Descripción</b>	Este modelo se basa en un conjunto de métricas para evaluar algunos factores de calidad de los modelos de datos.
<b>Complejidad</b>	Alta

$$Q = \{L, Cm, M, E, A, Ext, Con\} \text{ Con}$$

L: Legibilidad, Cm: Compleción, Crr: Corrección, M: Minimalidad,

E: Expresividad, A: Autoexplicación, Ext: Extensibilidad, Con: Consistencia

$$Q0 = p_1 * L + p_2 * Cm + p_3 * Crr + p_4 * M + p_5 * E + p_6 * A + p_7 * Ext + p_8 * Con$$

Con  $(p_1 + \dots + p_8) = 1$

$p_i$  = Asignación de valores ponderados para las variables (L...Con).

De este modo, se obtendrá un valor de Q0 entre 1 y 10, siendo el esquema con mayor valoración el de mejor calidad.

Aunque esta métrica es de fácil comprensión, la evaluación de cada característica es subjetiva, al igual que la asignación de valores a los ponderadores  $p_i$ . Otra característica de esta métrica es que la complejidad de la medición se traslada a la definición de criterios para la calificación de cada cualidad, lo que redundará en un mayor esfuerzo de aplicación y estandarización.

### **Métrica de Expresividad, Correctitud Semántica y Compleción de un esquema.**

La base para el uso correcto de esta métrica es la existencia de una especificación de requisitos detallada, que haya sido validada por el usuario.

Se utilizará como unidad base el "requisito", el cual expresa un requerimiento de información, o una restricción a los datos. Un requisito será cada sentencia que implique una construcción de una parte del esquema; ejemplos de requisito son: "Los colegios pueden ser públicos o privados", "Un colegio es público o privado pero no ambos" o "Un profesor puede hacer clases en uno o muchos colegios".

Una vez concluido un esquema conceptual se requerirá de un "lector de esquemas", el cual puede ser un desarrollador o idealmente una herramienta automatizada.

El encuestado debe "traducir" el esquema a lenguaje natural, luego de lo cual se contrasta con la especificación de requisitos.

De ahí se obtienen los siguientes valores.

Número de requisitos reales: **RT**

Número de requisitos acertados con los reales: **RA**

Número de requisitos inexistentes: **RI**

Número de requisitos no correctos: **RNC**

Con:

**RT**: número total de requisitos presentes en la especificación de requisitos.

**RA**: número de requisitos que el esquema satisface y que coinciden con algún requisito existente (de aquellos que se contabilizaron en RT).

**RI**: número de requisitos que el esquema satisface, que no aparecen en la especificación de requisitos y que no tienen conflicto con algún requisito de la especificación.

**RNC**: número de requisitos que, estando presentes en la especificación de requisitos, están representados de manera errónea en el esquema (no se contabilizan en RA ni en RI).

Además,  $RE = RA + RI + RNC$ ; indica los requisitos del esquema, es decir, número de requisitos que el esquema representa.

Sobre la base de lo anterior, se obtienen las siguientes métricas:

**Compleción** =  $RA/RT$

**Correctitud Semántica** =  $1 - RNC/RT$

**Expresividad** =  $1 - RI/RE$

Para cada métrica, los valores más cercanos a 1 indican mayor calidad.

### **Métrica de Autoexplicación de un esquema**

Un esquema se autoexplica si encierra en sí toda la información del problema abordado, por ello no necesita de otras notaciones para ser entendido.

Se distinguen dos tipos de información externa o no incluida en el esquema.

- Supuestos y descripciones no soportados por el esquema.
- Restricciones no modeladas y no soportadas por el esquema o por el lenguaje utilizado.

Sobre la base de examinar la información externa al esquema se obtiene:

Número de supuestos y descripciones: **NS**

Número de restricciones no modeladas: **NR**

Además, se utiliza de la especificación de requisitos:

Número total de requisitos: **RT**

Donde **NS** es el número de supuestos, que deben fraccionarse al nivel de cláusulas similares a un requisito.

**RT** es el número total de requisitos en la especificación.

**NR** es el número total de restricciones presentes en la especificación y no modeladas en el esquema, pero expresadas utilizando otros lenguajes.

La autoexplicación quedaría entonces determinada por: **Autoexplicación** =  $1 - (NS + NR)/RT$

Donde el mayor valor de calidad con respecto a la autoexplicación es 1.

### **Métrica de calidad sobre la base de la expresividad, correctitud semántica, completión y autoexplicación de un esquema.**

Se define la métrica de calidad **Q1** como:

**Q1 = Autoexplicación + Compleción + Correctitud Semántica + Expresividad**

Esta métrica entrega valores entre 0 y 4, asignándole a cada característica la misma importancia.

Considerando ahora una cierta priorización de criterios, donde completión y correctitud semántica son de mayor importancia que autoexplicación y expresividad, se define la siguiente métrica de calidad.

$Q = 0.5 * \text{Autoexplicación} + 0.5 * \text{Expresividad} + 1.8 * \text{Compleción} + 1.2 * \text{Correctitud semántica}$  (20)

### 2.2.2.5. Métricas de Piattini

<b>Nombre</b>	Métricas de Piattini.
<b>Autores</b>	Piattini.
<b>Año</b>	2000
<b>Clasificación</b>	Métricas para modelos conceptuales tradicionales.
<b>Fuente</b>	Calidad de software por Jorge Salamanca Escorial.
<b>Descripción</b>	Este modelo se basa en un conjunto de medidas para la evaluación del tamaño y otras para la complejidad.
<b>Complejidad</b>	Media

El conjunto de medidas propuestas es el siguiente:

<b>NE</b>	Número total de entidades del modelo.
<b>NA</b>	Número total de atributos en el modelo, tanto en las relaciones como en las entidades
<b>NDA</b>	Número total de atributos derivados en el modelo.
<b>NCA</b>	Número total de atributos compuestos en el modelo.
<b>NMVA</b>	Número total de atributos multivaluados en el modelo.
<b>NNR</b>	Número total de relaciones comunes en el modelo.
<b>NM: NR</b>	Número total de relaciones N: M en el modelo.
<b>NI: NR</b>	Número total de relaciones 1: N y 1:1 en el modelo.
<b>NBinaryR</b>	Número total de relaciones binarias en el modelo.
<b>NN-AryR</b>	Número total de relaciones n-arias.
<b>NIS_AR</b>	Número total de relaciones <b>Es_Un</b> (generalización/especialización) que existen en un modelo. En este caso se considera una relación por cada para padre-hijo, dentro de la relación <b>Es_Un</b> .
<b>NRefR</b>	Número total de relaciones reflexivas.
<b>NRR</b>	Número de relaciones redundantes.

Tabla 7. Métricas de Piattini

De estas, son métricas del tamaño las NE, NA, NDA y NMVA, y son métricas de complejidad el resto.

Estas métricas son objetivas y han sido validadas teóricamente, siguiendo el marco formal basado en la teoría de la medida, además de empíricamente mediante un caso de estudio y dos experimentos controlados.



Las métricas NNR, N1: NR, NBinaryR fueron caracterizadas por encima de la escala ordinal y NE, NA, NCA, NDA, NMVA, NN-AryR, NIS\_AR, NRefR, NRR en la escala de ratio. (19)

### 2.2.2.6. Métricas de Pradenas

<b>Nombre</b>	Métricas de Pradenas. (21)
<b>Autores</b>	Jaime Pradenas.
<b>Año</b>	2001.
<b>Clasificación</b>	Métricas para modelos conceptuales tradicionales
<b>Fuente</b>	Diseño Conceptual de Bases de Datos: Un enfoque basado en la medición de la calidad.
<b>Descripción</b>	Este modelo se basa en un conjunto de factores para evaluar la calidad de los modelos conceptuales utilizando fórmulas matemáticas.
<b>Complejidad</b>	Alta.

Tabla 8: Métricas de Pradenas.

### 2.2.2.7. Métricas de Abreu y Melo

<b>Nombre</b>	Métricas MOOD ( <i>Metrics for Object Oriented Design</i> )
<b>Autores</b>	Abreu y Melo
<b>Año</b>	2001
<b>Clasificación</b>	Métricas para modelos conceptuales orientados a objetos
<b>Fuente</b>	Diseño Conceptual de Bases de Datos: Un enfoque basado en la medición de la calidad.
<b>Descripción</b>	Son un conjunto de métricas para medir algunos de los principales mecanismos de los modelos orientados a objetos (encapsulamiento, polimorfismo, herencia y paso de mensajes), para poder evaluar la productividad del desarrollo y la calidad del producto. Se enfocan en las características de los diagramas de clases.
<b>Complejidad</b>	Media

Las definiciones de las diferentes métricas son:

**MHF:** El *Method Hiding Factor* (factor de ocultamiento de los métodos) se define como el cociente entre la suma de las invisibilidades de todos los métodos definidos en todas las clases y el número total de métodos definidos en el sistema. La invisibilidad de un método es el porcentaje total de clases desde las cuales el método es invisible.

**MHF:** El MHF es el ratio entre el número de métodos privados y el número total de métodos, y sirve para medir la encapsulación. Abreu y Melo demostraron empíricamente que cuando se incrementa MHF, la densidad de defectos y el esfuerzo necesario para corregirlos deberían disminuir. Para el cálculo de esta métrica no se consideran los métodos heredados.

**AHF:** El *Attribute Hiding Factor* (factor de ocultamiento de los atributos) se define como el cociente entre el número de invisibilidades de todos los atributos definidos en todas las clases y el número total de atributos definidos en el sistema. Se propone también como medida de encapsulación. Idealmente el valor de esta métrica debería ser siempre del 100%, siendo necesario para ello ocultar todos los atributos. Las pautas de diseño orientado a objetos (OO) sugieren que no hay que emplear atributos públicos, ya que se considera que esto viola los principios de encapsulación al exponer la implementación de las clases. Para mejorar el rendimiento, a veces se evita el uso de métodos que acceden o modifican atributos accediendo a ellos directamente.

**MIF:** El *Method Inheritance Factor* (factor de herencia de los métodos) es el cociente entre el número de métodos heredados en todas las clases del sistema y el número total de métodos (heredados y locales) en todas las clases. El MIF se utiliza como una medida de la herencia, y por tanto, sirve como medida de la reusabilidad. También se propone como ayuda para evaluar la cantidad de recursos necesarios a la hora de probar. El empleo de la herencia se ve como un compromiso entre la facilidad de reutilización que proporciona, y la facilidad de comprensión y mantenimiento del sistema.

**AIF:** El *Attribute Inheritance Factor* (factor de herencia de los atributos) está definido como el cociente entre el número de atributos heredados en todas las clases del sistema y el número total de atributos existentes (heredados y definidos localmente) en todas las clases.

**AIF:** Lo mismo que la anterior métrica expresa la capacidad de reutilización del sistema. Por el contrario se tiene demasiada reutilización de código a través de herencia hace que el sistema sea más difícil de entender y mantener.

**PF:** El *Polymorphism Factor* (factor de polimorfismo) se define como el ratio entre el número actual de situaciones diferentes posibles de polimorfismo y el número máximo de posibles situaciones distintas de polimorfismos para cada clase. El factor PF es además, una medida indirecta de la asociación dinámica del sistema. El polimorfismo se debe a la herencia. Abreu indica que, en algunos casos, sobrecargando métodos se reduce la complejidad y, por tanto, se incrementa la facilidad de mantenimiento y la facilidad de comprensión del sistema. Diversos autores han demostrado que esta métrica no cumple todas las propiedades definidas para ser válida, ya que en un sistema sin herencia el valor de PF es indefinido, lo que exhibe una discontinuidad.

**CF:** *Coupling Factor* (factor de acoplamiento) se define como la proporción entre el máximo número posible de acoplamientos en el sistema y el número real de acoplamientos no imputables a la herencia. En otras palabras, indica la comunicación entre clases. El acoplamiento se ve como una medida del incremento de la complejidad, reduciendo la encapsulación y el posible rehusó; limita, por tanto, la facilidad de comprensión y de mantenimiento del sistema. CF puede ser una medida indirecta de los atributos con los cuales está relacionado: complejidad, falta de encapsulación, carencia de reutilización, facilidad de comprensión y poca facilidad de mantenimiento. Los autores han encontrado una correlación positiva: al incrementar el acoplamiento entre clases, se incrementa la densidad de defectos y la dificultad en el mantenimiento. (19)

### 2.2.2.8. Métricas de Chindamber y Kemerer

<b>Nombre</b>	Métricas de Chindamber y Kemerer.
<b>Autores</b>	Chindamber y Kemerer.
<b>Año</b>	-
<b>Clasificación</b>	Métricas para modelos conceptuales orientados a objetos
<b>Fuente</b>	Calidad de software por Jorge Salamanca Escorial.
<b>Descripción</b>	Es un conjunto de seis métricas para la complejidad del diseño OO, aunque no todas pueden aplicarse a nivel conceptual, y además han sido muy criticadas por su ambigüedad e imprecisión.
<b>Complejidad</b>	Media

#### Métricas a nivel de herencia

**DIT:** La métrica *Depth of Inheritance Tree* (profundidad en árbol de herencia) se define como la profundidad del árbol de una clase (en los casos de herencia múltiple es la máxima longitud desde el nodo hasta la raíz del árbol). Se basa en que cuanto más profunda está la clase en la jerarquía, mayor número de operaciones puede heredar. Se propuso como una medida de la complejidad de una clase, complejidad de diseño y reusabilidad potencial. El uso de la herencia se contempla como un compromiso, ya que:

- Altos niveles de herencia indican objetos complejos, los cuales pueden ser difíciles de probar y reutilizar.
- Bajos niveles en la herencia pueden señalar que el código está escrito en un estilo funcional, sin aprovechar el mecanismo de herencia proporcionado por la orientación a objetos.

**DIT:** En general la herencia se utiliza poco. Distintos autores han encontrado una correlación positiva entre DIT y el número de problemas emitidos por el usuario, poniendo en duda el uso efectivo de la herencia. Por su parte, otros autores sugieren un umbral de 6 niveles como indicador de un abuso en la herencia en distintos lenguajes de programación (C++, *Smalltalk*) Sistemas construidos a partir de *frameworks* suelen presentar unos niveles de herencia altos, ya que las clases se construyen a partir de una jerarquía existente. En lenguajes como Java o *Smalltalk*, las clases siempre heredan de la clase *Object*, lo que añade uno al valor de DIT. Los problemas que surgen con esta métrica se deben a las diferentes características de la herencia, ya que DIT no queda claramente definida y no se puede ver como una medida de reutilización. Es fácil imaginar clases con gran profundidad en la jerarquía reutilizando menos métodos que una clase poco profunda, pero que es muy ancha.

**NOC:** La métrica *Number Of Children* (número de hijos) se define como el número de subclases inmediatas subordinadas a una clase, es decir, la cantidad de subclases que pertenecen a una clase.

Esta medida indica cuántas subclases van a heredar las operaciones de la clase padre. También es un indicador del nivel de rehuso, la posibilidad de haber creado abstracciones erróneas, y es un indicador del nivel de pruebas requerido.

Aunque un mayor número de hijos representa una mayor reutilización de código, tiene sus inconvenientes:

- Mayor probabilidad de emplear incorrectamente la herencia creando abstracciones erróneas.
- Mayor dificultad para modificar una clase, pues esto afecta a todos los hijos que tienen dependencia con la clase base.
- Se requiere mayor número de recursos para probar. Es un potencial indicador de la influencia que una clase puede tener sobre el diseño del sistema. Si el diseño depende mucho de la reutilización a través de la herencia, quizás sea mejor dividir la funcionalidad en varias clases.

Estas dos métricas (DIT y NOC) presentan medidas objetivas para la complejidad de las clases y han sido validadas teóricamente por los autores al corroborar que satisfacen los axiomas de *Weyuker*. La validación empírica fue realizada por *Basil*, que encontraron que la posibilidad de encontrar un fallo es directamente proporcional a DIT e inversamente al NOC.

### **Métricas a nivel de acoplamiento**

**CBO:** *Coupling Between Objects* (acoplamiento entre objetos) de una clase se define como el número de clases a las cuales una clase está ligada. Se da dependencia entre dos clases cuando una de ellas usa métodos o variables de la otra clase. Las clases relacionadas por herencia no se tienen en cuenta. Los autores proponen que esta métrica sea un indicador del

esfuerzo necesario para el mantenimiento y las pruebas. También indican que cuanto más acoplamiento se da en una clase, más difícil será reutilizarla. Además las clases con excesivo acoplamiento dificultan la comprensión y hacen más difícil el mantenimiento, por lo que será necesario un mayor esfuerzo y unas pruebas rigurosas. Las clases tendrían que ser lo más independientes posible y, aunque siempre se precisa una dependencia entre clases, cuando ésta es grande, la reutilización puede ser más cara que la reescritura. Al reducir el acoplamiento se reduce la complejidad, se mejora la modularidad y se promueve la encapsulación.

### Métricas a nivel de clases

Las métricas que se consideran a nivel de clases identifican características dentro de las clases destacando diferentes aspectos de sus abstracciones y ayudando a descubrir clases que podrían necesitar ser rediseñadas.

Algunas de estas son:

**RFC:** *Response For a Class* (respuesta de una clase) es el cardinal del conjunto de todos los métodos que se pueden invocar como respuesta a un mensaje, a un objeto de la clase o como respuesta a algún método en la clase.

Esto incluye a todos los métodos accesibles dentro de la jerarquía de la clase.

En otras palabras, RFC cuenta las ocurrencias de llamadas a otras clases desde una clase particular.

$$RFC = |RS|$$

Donde RS es el conjunto respuesta para la clase, que se puede expresar como:

$RS = \{M\} \cup \{R_i\}$  donde  $\{R_i\}$  es el conjunto de métodos llamados por el método  $i$ , y  $\{M\}$  es conjunto de todos los métodos en la clase.

**RFC:** Para los autores, RFC es una medida de la complejidad de una clase a través del número de métodos y de su comunicación con otras, pues incluye los métodos llamados desde fuera de la clase. RFC es un indicador de los recursos necesarios para las pruebas y la depuración. Cuanto mayor es RFC, más complejidad tiene el sistema, ya que es posible invocar más métodos como respuesta a un mensaje. Se ha señalado que la definición de esta métrica es ambigua y fuerza al usuario a interpretarla.

**WMC:** *Weighted Methods per Class-WMC* (métodos ponderados por clase), mide la complejidad de una clase. Si todos los métodos se estiman igualmente complejos, entonces WMC es, simplemente, el número de métodos definidos en una clase.

$WMC = \sum c_i$  donde una clase  $C_i$  tiene los métodos  $M_1, \dots, M_n$  con su complejidad respectiva  $c_1, \dots, c_n$ .

**WMC:** Los autores sugieren que WMC es una medida de la complejidad de una clase. Clases con un gran número de métodos requieren más tiempo y esfuerzo para desarrollarlas y

mantenerlas, ya que influirán en las subclases heredando todos sus métodos. Además, estas clases tienden a ser específicas de la aplicación, con lo que se limita su posibilidad de reutilización. Lorenz y Kidd plantean un umbral de 40 o 20, dependiendo de si las clases son o no de interfaz de usuario.

En WMC se pueden apreciar los siguientes problemas:

- WMC mide presuntamente la complejidad, pero no ofrece ninguna definición asociada a la complejidad.
- WMC no se puede contemplar como un indicador del esfuerzo necesario para desarrollar una clase, ya que es fácil imaginar clases con pocos métodos complicados y clases con un gran número de métodos, pero muy simples.

Así pues parece que solo se debería considerar esta métrica simplemente como una medida del tamaño de una clase.

**LCOM: *Lack of Cohesion in Methods*** (falta de cohesión en los métodos) establece en qué medida los métodos hacen referencia a atributos.

LCOM es una medida de la cohesión de una clase midiendo el número de atributos comunes usados por diferentes métodos, indicando la calidad de la abstracción hecha en la clase.

Un valor alto de LCOM implica falta de cohesión, es decir, escasa similitud de los métodos. Esto quizás signifique que la clase está compuesta de elementos no relacionados, incrementándose la complejidad y la probabilidad de errores durante el desarrollo. Es deseable una alta cohesión de los métodos dentro de una clase, ya que ésta no se puede dividir fomentando la encapsulación.

Se han destacado dos problemas con esta métrica:

- Dos clases pueden tener LCOM=0, mientras una tiene más variables comunes que otra.
- No existen guías para la interpretación de esta métrica. (19)

### 2.2.2.9. Métricas de Lorenz y Kid

<b>Nombre</b>	Métricas de Lorenz y Kidd
<b>Autores</b>	Lorenz y Kidd
<b>Año</b>	-
<b>Clasificación</b>	Métricas para modelos conceptuales orientados a objetos
<b>Fuente</b>	Calidad de software por Jorge Salamanca Escorial.
<b>Descripción</b>	Este es un conjunto de métricas de diseño para medir las características estáticas de un producto software.
<b>Complejidad</b>	Alta

### **Métricas de tamaño**

**PIM:** La métrica Número de Métodos de Instancia Públicos es el número total de métodos públicos de instancias (los que están disponibles como servicios para otras clases). Se considera que mide la cantidad de responsabilidad que tiene una clase.

**NIM:** Se define el Número de Métodos de Instancia como la suma de todos los métodos (públicos, protegidos y privados) de una clase. Según los autores es una medida de la cantidad de colaboración utilizada.

**NIV:** El Número de Variables de Instancia se determina por el número total de variables (privadas y protegidas) a nivel de instancia que tiene una clase

**NCM:** El Número de Métodos de Clase es el número total de métodos a nivel de clase.

**NVV:** El Número de Variables de Clase es el total de variables a nivel de clase que tiene una clase.

### **Métricas de herencia**

**NMO:** El Número de Métodos Sobrecargados es el número total de métodos sobrecargados en una subclase. Se propuso para medir la calidad del uso de la herencia

**NMI:** El Número de Métodos Heredados se define como el número de métodos que hereda una clase. También mide la calidad del uso de la herencia.

**NMA:** El Número de Métodos Añadidos es el número total de métodos que se definen en una subclase. Igual que las anteriores mide la calidad de uso de la herencia.

**SIX:** El Índice de Especialización para una clase se define como el número de métodos sobrescritos multiplicado por el nivel de anidamiento en la jerarquía y dividido entre el número total de métodos.

$$SIX = \frac{No\_métodos\_redefinidos * Anidamiento\_en\_la\_Jerarquía}{No\_total\_de\_métodos}$$

Mide el grado en que una subclase redefine el comportamiento de una superclase.

Esta fórmula pondera más las redefiniciones que ocurren en niveles más profundos del árbol de herencia, ya que cuanto más especializada es una clase, menos probabilidad existe de que su comportamiento sea reemplazado.

### **Métricas de características internas de una clase**

**APPM:** El Promedio de Parámetros por Método se define como el cociente entre el número total de parámetros por método y el número total de métodos.

Estas métricas están enfocadas a las características internas del diseño O.O. con medidas objetivas y una herramienta, la *OOMetric*, que sólo puede aplicarse a código escrito en C++ y *Smalltalk*.

No se han validado teóricamente. Se han validado parcialmente de forma empírica. (19)

### 2.2.2.10. Métricas de Género

<b>Nombre</b>	Métricas de Género
<b>Autores</b>	Género
<b>Año</b>	2000
<b>Clasificación</b>	Métricas para modelos conceptuales orientados a objetos
<b>Fuente</b>	Calidad de software por Jorge Salamanca Escorial.
<b>Descripción</b>	Es un conjunto de métricas para la medida de la complejidad estructural de los modelos de clase debido al uso de relaciones UML
<b>Complejidad</b>	Media

### Métricas a nivel de modelo de clases (19)

<b>Nassoc</b>	<b>La métrica Número de Asociaciones</b> se define como el número total de asociaciones dentro de un modelo de clases.
<b>Nagg</b>	<b>La métrica Número de Agrupaciones</b> se define como el número de relaciones de agregación dentro de un modelo de clases.
<b>Ndep</b>	<b>El Número de Dependencias</b> es el número total de relaciones de dependencia en un modelo de clases.
<b>Ngen</b>	<b>El Número de Generalizaciones</b> se define como el número total de relaciones de generalización dentro de un modelo de clases.
<b>NgenH</b>	<b>La métrica Número de Jerarquías de Generalización</b> es el total de jerarquías de generalización en un modelo de clases.
<b>NaggH</b>	<b>La métrica Número de Jerarquías de Agregación</b> es el número total de jerarquías de agregación dentro de un modelo de clases.
<b>MaxDIT</b>	<b>La métrica Máximo DIT</b> se define como el máximo de los valores DIT obtenidos de cada clase del modelo de clases. El valor DIT ( <i>Depth Inheritance Tree</i> ) es la ruta más larga desde la clase a la clase raíz de la jerarquía de generalización.
<b>MaxHAgg</b>	<b>La métrica Máximo HAgg</b> es el máximo de los valores Hagg de cada clase del modelo de clases. El valor HAgg, dentro de la jerarquía de agregación, es la longitud de la ruta más larga desde la clase hasta las hojas.
<b>NassocC</b>	<b>El Número de Asociaciones por Clase</b> es el número total de asociaciones de una clase (con otras clases o con ella misma).
<b>Hagg</b>	<b>La Altura</b> de una clase es la longitud de la ruta más larga desde la clase a las hojas dentro de una jerarquía de agregación.
<b>NODP</b>	<b>El Número de Partes Directas</b> de una clase es el número de Partes Directas que



	contiene una clase que pertenece a una jerarquía de agregación.
<b>NP</b>	<b>El Número de Partes</b> es el número de clases “partes” (directas o indirectas) de una clase “todo”.
<b>NW</b>	<b>La métrica Número de Todos</b> se define como el número de clases “todos” (directas e indirectas) en una clase “parte”.
<b>Magg</b>	<b>La métrica Agregación Múltiple</b> es el número de clases “todo” directas que tiene una clase en una jerarquía de agregación.
<b>Ndepln</b>	<b>El Número de Dependencias In</b> se define como el número de clases que depende de una clase dada.
<b>NdepOut</b>	<b>El Número de Dependencias Out</b> es el número de clases de las que depende la clase dada.

Tabla 9. Métricas de Género

### Tablas resumen de la concordancia entre autores

Factores	Moody	Kesh	Varas P	Piattini	Pradenas
Compleción	X		X		
Flexibilidad	X				
Corrección	X		X		X
Simplicidad	X	X			
Integración	X				
Implementabilidad	X				
Validez		X			
Consistencia		X	X		X
Compleitud	X	X			X
Usabilidad		X			
Precisión		X			
Rendimiento		X			
Legibilidad			X		X
Minimalidad			X		X
Expresividad			X		X
Autoexplicación	X		X		X
Complejidad				X	
Tamaño			X	X	X

Tabla 10: Resumen de Modelos Conceptuales Tradicionales

Factores	Abreu y Melo	Chindamber Kemerer	y Lorenz y Kidd	Género
Encapsulamiento	X			
Herencia	X	X	X	
Polimorfismo	X			
Acoplamiento	X	X		
Clases		X	X	X
Tamaño			X	
Paso de mensaje	X			

Tabla 11: Tabla Resumen de Modelos Conceptuales Orientados a Objetos

### 2.3. Descripción del proceso de obtención de la propuesta

Para un mejor entendimiento del proceso a realizar para darle cumplimiento a las tareas propuestas se ofrece una descripción detallada de las principales fases en el desarrollo del trabajo.

**Estudio inicial:** En esta fase se va a realizar un estudio del estado actual del desarrollo de métricas para diseños de BD relacionales con el objetivo de definir cuáles son los factores de calidad considerados claves por los diferentes autores para un buen desarrollo de los mismos.

**Selección y ajuste:** En esta fase se realizará una selección de las métricas estudiadas para llevar a cabo las pruebas en fases posteriores. Se diseñará una plantilla en la cual se realizará el registro de los datos de la aplicación de la métrica en los diseños de BD de la muestra seleccionada.

**Prueba piloto:** Se aplicará la plantilla de prueba en los diseños de bases datos, tomando como muestra un grupo de proyectos de la facultad entre los que se encuentran: Servicios Comunitarios y Akademos por ser proyectos que desarrollan BD relativamente grandes y efectuados en diferentes gestores de BD.

**Evaluación de resultados:** Después de haber finalizado el período de prueba se realizará un análisis de los resultados obtenidos para verificar que los resultados fueran los correctos, además se emite una evaluación general del diseño, para esto se definirán varios intervalos en los cuales se le otorgará un valor cualitativo a los resultados obtenidos de la aplicación de las métricas expresados de manera cuantitativa.

**Refinamiento:** Después de haber realizado la prueba en los proyectos, y analizado el comportamiento de cada una de las métricas en busca de posibles inconsistencias se pasa a la fase de refinamiento, en la que se realizan algunos ajustes con el objetivo de conformar la propuesta final con las métricas que más impacto hayan tenido en el proceso de prueba.

**Definición de la propuesta:** En esta fase se realizará la definición de la propuesta, la cual será validada en dos sentidos: teórica y empíricamente, para ello se utilizará el método multicriterio que se basa en la evaluación cuantitativa de criterios previamente definidos por

parte de expertos en el tema, que permite determinar si se acepta o no la propuesta analizada. La validación empírica se garantiza en la aplicación de la prueba piloto.

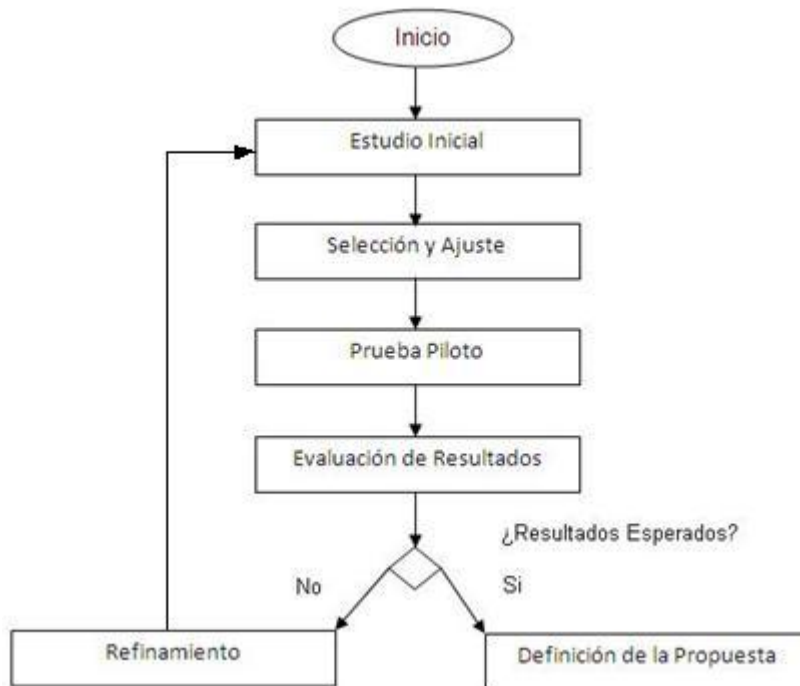


Fig 9. Proceso de desarrollo de la propuesta de solución

## 2.4. Beneficios de la propuesta

Validación de las teorías con datos reales, lo cual proporciona a los proyectos un mecanismo útil para el desarrollo eficiente de sus BD. Con el desarrollo de esta propuesta se puede contar con un proceso para identificar, ajustar y validar tanto empírica y teóricamente una métrica. Además, a partir de las métricas propuestas, para el desarrollo de la base de datos, es posible hacer una valoración objetiva de la calidad con que se desarrollaron las mismas.

## 2.5. Roles y responsabilidades en el proceso de evaluación

**Líder del equipo de evaluación y calidad del software:** Responsable de la evaluación, debe estar capacitado y autorizado por el asesor de calidad de la entidad y debe asegurar que:

- El proceso de evaluación sea realizado conforme a la guía propuesta y que los resultados de ésta sean representativos.
- Los datos de la evaluación sean registrados en la plantilla con el formato adecuado para la evaluación.

**Equipo evaluador:** Personas con experiencias en Ingeniería de Software, sólidos conocimientos de la guía, el número de integrantes dependerá de la complejidad del proyecto o la urgencia con que se realice la evaluación, estos realizan las siguientes actividades,

recolección de información, realización de la evaluación e información del estado actual del proyecto.

**Coordinador del proyecto:** Miembro del proyecto encargado de representarlo en la evaluación. Debe tener conocimiento del proceso de desarrollo de la base de datos.

## **2.6. Conclusiones**

El estudio realizado de los diferentes grupos de métricas existentes permitió identificar que la mayoría de ellos en su propuesta mide: completión, corrección, consistencia, completitud, legibilidad, minimalidad, expresividad y autoexplicación. Con todo la caracterización elaborada de métricas tanto para BD relacionales y Orientado a Objetos (OO), se realiza la propuesta de un proceso de prueba que se define para que utilizando un modelo de investigación acción se pueda ir perfeccionando la propuesta. Todo este proceso de caracterización permitió conformar una propuesta de solución así como la caracterización de cada fase que la compone.

# Capítulo 3

## Validación de la Propuesta

### 3.1. Introducción

Luego de haberse identificado los principales parámetros a los que estará enfocada la propuesta y de realizarse una cuidadosa investigación sobre las posibles métricas que puedan aplicarse en la universidad, se definen en este capítulo, como aporte a la mejora de la gestión de la calidad de los proyectos, un conjunto de métricas que han sido consideradas como las que mejor se ajustan a las necesidades actuales de la universidad, a las cuales se les aplicarán las mejoras necesarias.

### 3.2. Desarrollo del proceso de la aplicación de la propuesta

Para llegar a la definición de la propuesta final fue necesario ejecutar eficientemente las actividades planteadas en cada fase del proceso diseñado para la obtención de la propuesta descrito anteriormente.

#### 3.2.1. Fase inicial

Se realizó un estudio exhaustivo de las principales métricas encontradas. Se aplicó una encuesta a desarrolladores de BD de proyectos de la universidad con el objetivo de definir los principales factores que hay que tener en cuenta a la hora de evaluar una base de datos relacional.

#### 3.2.2. Selección y ajuste

A partir del estudio previo realizado y la opinión de los desarrolladores se efectuó una selección de las métricas, las cuales fueron seleccionadas a partir de las opiniones emitidas por los desarrolladores expresadas a través de la encuesta realizada en la fase anterior.

### 3.2.3. Aplicación de métricas en proyectos de la universidad

Para la validación de la propuesta se le aplicó las métricas a un grupo de diseños de BD tomando como muestra para este proceso los proyectos Servicios Comunitarios y Akademos. Para la realización de la prueba se le solicitó al proyecto los artefactos necesarios entre los que se encuentran: diseño de base de datos, especificación de requisitos y la descripción de casos de uso.

#### 3.2.3.1. Aplicación de métrica en el proyecto Servicios Comunitarios

Para la aplicación de estas métricas se tomó como muestra el diseño de la base de datos del Módulo: Reservación de Gas.

##### Conjunto #1:

Característica	Variable	Fórmula	Descripción	Valores	Resultado
Adecuación al problema	$O_1$		Criterio del usuario		5
Validez del Modelo	$O_2$		Criterio de un especialista		5
Consistencia	$O_3$	$O_3 = 5 - D1$	Donde NI: # de inconsistencias y n # de redacciones del modelo	NI = 0 N = 0	5
		$D1 = NI / 4n$			
Concisión	$O_4$	$O_4 = M((Cn2 - n1) / Cn - (n-1))$	Donde Cn2 son el # relaciones, n1 el # de entidades y n-1 # mínimo de relaciones	Cn2 = 21 N1 = 25 n-1 = 24	1.33
Compleitud	$O_5$	$O_5 = 5 - NF$	Donde NF: # de fallos en dependencia a la importancia de la consulta	NF = 0	5
Cohesión	$O_6$	$O_{6i} = M((ne - np) / (ne - 1))$ $O_6 = \sum O_{6i} / n$	Donde ne: # de atributos de la entidad np: # de atributos identificadores primarios	N= 108 $\sum O_{6i} = 23.63$	0.21
Validez de contenido	$O_7$	$O_7 = (1 - ni / \sum ne)$	Donde ni: # de entidades no Válidas ne: # de atributos de la entidad	Ni = 0	5

Tabla 12: Prueba Servicios Comunitarios Conjunto 1.

El rango de valores a obtener de las variables  $O_i$  ( $i=1.....7$ ) sería:  $(1 \leq O_i \leq 5)$ .

S<sub>1</sub>: Facilidad de Uso desde el punto de vista del usuario:

$$S_1 = (O_1 + O_3 + O_4 + O_5) / 4$$

$$S_1 = (5 + 5 + 1.33 + 5) / 4$$

$$S_1 = 4.08$$

S<sub>2</sub>: Usabilidad desde el punto de Vista del Diseñador:

$$S_2 = (O_2 + O_3 + O_5 + O_6 + O_7) / 5$$

$$S_2 = (5 + 5 + 5 + 0.21 + 5) / 5$$

$$S_2 = 4.04$$

S<sub>3</sub>: Facilidad de Mantenimiento:

$$S_3 = (O_2 + O_4 + O_6) / 3$$

$$S_3 = (5 + 1.33 + 0.21) / 3$$

$$S_3 = 2.18$$

S<sub>4</sub>: Precisión

$$S_4 = (O_3 + O_5) / 2$$

$$S_4 = (5 + 5) / 2$$

$$S_4 = 5$$

S<sub>5</sub>: Rendimiento:

$$S_5 = (O_4 + O_5) / 2$$

$$S_5 = (1.33 + 5) / 2$$

$$S_5 = 3.16$$

**Cálculo de la calidad del modelo:**

$$Q = \sum W_i S_i \text{ (con } i \text{ del 1 al 5)}$$

$$Q = 3 * 4.08 + 1 * 4.04 + 3 * 2.18 + 2 * 5 + 3 * 3.16$$

$$Q = 42.3$$

Donde W<sub>i</sub> son los pesos de los factores de comportamiento S<sub>i</sub> son los valores de dichos factores.

**Conjunto #2:**

Característica	Fórmula	Descripción	Valores	Resultado
<b>Legibilidad</b>	$\frac{1}{\frac{NFI}{e^{0.5 * NE}}}$	Donde NFI: # de fuentes de legibilidad NE: # de estructuras	NE = 25 NFI = 0	0
<b>Compleitud</b>	$(RE/RT)^2$	Donde RE: # de requisitos de especificación expresados en el esquema RT: # de requisitos totales	RE = 11 RT = 12	0.83

<b>Correctitud Semántica</b>	$1/ErrSem + 1$	Donde ErrSem: # de errores semánticos	ErrSem = 0	1
<b>Correctitud Sintáctica</b>	$1/ErrSin + 1$	Donde ErrSin: # de errores sintácticos	ErrSin = 0	1
<b>Minimalidad</b>	$1-(FR/A)^2$	Donde FR: # de fuentes de redundancia A: # de atributos del esquema	A = 108 FR = 0	1
<b>Expresividad</b>	$\frac{RA}{2 * RT} + \frac{1}{2 * e^{\frac{RI}{RT}}}$	Donde RA: # de requisitos acertados RI: # de requisitos presentes en el esquema RT: # de requisitos totales de la especificación.	RA= 11 RI = 0 RT= 12	0.95
<b>Autoexplicación</b>	$\frac{1}{IA / e^{0.5 * RT}}$	Donde IA: información anexa al esquema	IA = 0	0
<b>Consistencia</b>	$1/ NI + 1$	Donde NI: # de inconsistencias del esquema	NI = 1	0.5
<b>Economía</b>	$\left[ \frac{RAC}{NE * RTE} * 20 \right]^2$	Donde RAC: # de requisitos de estructura acertados RTE: # total de requisitos de estructura de la especificación NE: # de estructuras	NE = 25 RTE = 12 RAC = 11	0.53

Tabla 13 Prueba Servicios Comunitarios Conjunto 2.

**Cálculo de la calidad del modelo:**

$Q = W_1 * \text{Legibilidad} + W_2 * \text{Compleitud} + W_3 * \text{Correctitud Semántica} + W_4 * \text{Correctitud Sintáctica} + W_5 * \text{Minimalidad} + W_6 * \text{Expresividad} + W_7 * \text{Autoexplicación} + W_8 * \text{Consistencia} + W_9 * \text{Economía}$

Donde  $W_i$   $i= 1$  a  $9$  son los pesos asociados a cada criterio de calidad.

$Q = 14.12.$



### 3.2.3.2. Aplicación de métrica en el proyecto Akademos

Conjunto #1:

Característica	Variable	Fórmula	Descripción	Valores	Resultado
Adecuación al problema	O <sub>1</sub>				5
Validez del Modelo	O <sub>2</sub>				4
Consistencia	O <sub>3</sub>	O <sub>3</sub> = 5- D1	Donde NI: # de inconsistencias y n # de redacciones del modelo	NI = 0 N = 0	5
		D1= NI/ 4n			
Concisión	O <sub>4</sub>	O <sub>4</sub> = ((Cn2 - n1)/Cn2 - (n-1))	Donde Cn2 son el # relaciones, n1 el # de entidades y n-1 # mínimo de relaciones	Cn2 = 81 N1 = 53 n-1 = 52	1.037
Compleitud	O <sub>5</sub>	O <sub>5</sub> = 5 - NF	Donde NF: # de fallos en dependencia a la importancia de la consulta	NF = 0	5
Cohesión	O <sub>6</sub>	O <sub>6i</sub> = M((ne-np) / (ne - 1)) O <sub>6</sub> = ∑O <sub>6i</sub> /n	Donde ne: # de atributos de la entidad Np: # de atributos identificadores primarios	∑O <sub>6i</sub> = 43.9 N = 53	0.82
Validez de contenido	O <sub>7</sub>	O <sub>7</sub> = (1- ni/∑ne)	Donde ni: # de entidades no Válidas ne: # de atributos de la entidad		5

Tabla 14. Prueba Akademos Conjunto 1.

El rango de valores a obtener de las variables O<sub>i</sub> (i =1.....7) sería: (1<=O<sub>i</sub><=5).

S<sub>1</sub>: Facilidad de Uso desde el punto de vista del usuario

$$S_1 = (5 + 5 + 1.037 + 5) / 4$$

$$S_1 = 4$$

$$S_1 = (O_1 + O_3 + O_4 + O_5) / 4$$

S<sub>2</sub>: Usabilidad desde el punto de Vista del Diseñador

$$S_2 = (4 + 5 + 5 + 0.82 + 5) / 5$$

$$S_2 = 3.96$$

$$S_2 = (O_2 + O_3 + O_5 + O_6 + O_7) / 5$$

S<sub>3</sub>: Facilidad de Mantenimiento

S<sub>4</sub>: Precisión

$$S_3 = (O_2 + O_4 + O_6) / 3$$

$$S_4 = (O_3 + O_5) / 2$$

$$S_3 = (4 + 1.037 + 0.82) / 3$$

$$S_4 = (5 + 5) / 2$$

$$S_3 = 1.95$$

$$S_4 = 5$$

S<sub>5</sub>: Rendimiento

$$S_5 = (O_4 + O_5) / 2$$

$$S_5 = (1.037 + 5) / 2$$

$$S_5 = 3.01$$

**Cálculo de la Calidad del Modelo:**

$$Q = \sum W_i S_i \text{ (con } i \text{ del 1 al 5)} \quad (1 \leq W_i \leq 3)$$

$$Q = 40.94$$

Donde  $W_i$  son los pesos de los factores de comportamiento  $S_i$  son los valores de dichos factores en este caso.

**Conjunto #2:**

Característica	Fórmula	Descripción	Valores	Resultado
<b>Legibilidad</b>	$\frac{1}{\frac{NFI}{e^{0.5*NE}}}$	Donde NFI: # de fuentes de ilegibilidad NE: # de estructuras	NFI = 0	0
<b>Compleitud</b>	$(RE/RT)^2$	Donde RE: # de requisitos de especificación expresados en el esquema RT: # de requisitos totales	RE=51 RT=51	1
<b>Correctitud Semántica</b>	$1/ErrSem + 1$	Donde ErrSem: # de errores semánticos	ErrSem=0	1
<b>Correctitud Sintáctica</b>	$1/ErrSin + 1$	Donde ErrSin: # de errores sintácticos	ErrSin=0	1
<b>Minimalidad</b>	$1-(FR/A)^2$	Donde FR: # de fuentes de redundancia A: # de atributos del	FR=0 A=239	1

		esquema		
<b>Expresividad</b>	$\frac{RA}{2 * RT} + \frac{1}{2 * e^{\frac{RI}{RT}}}$	Donde RA: # de requisitos acertados RI: # de requisitos presentes en el esquema	RA=51 RI=0 RT=51	1
<b>Autoexplicación</b>	$\frac{1}{\frac{IA}{e^{0.5 * RT}}}$	Donde IA: Información anexa al esquema	IA=0 RT=51	0
<b>Consistencia</b>	1/ NI + 1	Donde NI: # de inconsistencias del esquema	NI= 0	1
<b>Economía</b>	$\left[ \frac{RAC}{NE * RTE} * 20 \right]^2$	Donde RAC: # de requisitos de estructura acertados RTE: # total de requisitos de estructura de la especificación NE: # de estructuras	RAC=51 NE= 52 RTE= 51	0.14

Tabla 15. Prueba Akademos Conjunto 2

**Cálculo de la calidad del modelo:**

$$Q = W_1 * \text{Legibilidad} + W_2 * \text{Compleitud} + W_3 * \text{Correctitud Semántica} + W_4 * \text{Correctitud Sintáctica} + W_5 * \text{Minimalidad} + W_6 * \text{Expresividad} + W_7 * \text{Autoexplicación} + W_8 * \text{Consistencia} + W_9 * \text{Economía}$$

Donde  $W_i$   $i= 1$  a  $9$  son los pesos asociados a cada criterio de calidad.

$$Q = 3*0 + 2*1 + 3*1 + 2*1 + 3*1 + 2*1 + 2*0 + 3*1 + 2*0.14$$

$$Q = 0 + 2 + 3 + 2 + 3 + 2 + 0 + 3 + 0.28$$

$$Q = 15.28$$

**3.2.4. Evaluación de los resultados**

A lo largo del proceso de prueba desarrollado se han identificado una serie de deficiencias que no permite al evaluador llegar a una valoración objetiva de los resultados. Entre ellas está el caso del cálculo de los factores economía y consistencia que mostraban resultados contradictorios a los esperados. Algunos factores eran subjetivos pues eran dependientes de la opinión del evaluador por lo que los resultados no son precisos, ejemplo de estos factores: adecuación al problema y validez del modelo.

### **3.2.5. Principales deficiencias encontradas**

En la primera iteración del proceso de prueba se detectaron algunas dificultades en las métricas, entre ellas están:

- Proceso engorroso: se hace difícil de forma manual la obtención de los datos, pues existen proyectos con BD extensas y una gran cantidad de requisitos.
- Métricas complejas de calcular: dentro de las métricas analizadas y probadas hay algunas muy fáciles de calcular pero otras son bastante complejas por lo que manualmente es un poco trabajoso, sería ideal que se automatizara este cálculo.
- Algunas no brindan una información precisa funcionan como un indicador.
- Los pesos asociados a la importancia de cada factor presente en la expresión de QDB influyen en ocasiones de forma determinante en la evaluación final.

### **3.2.6. Refinamiento**

Se definió cuales métricas eran las apropiadas para conformar la propuesta final. Además se identificaron las posibles mejoras y ajustes a realizar para obtener un conjunto de métricas más precisas, coherentes y eficientes. Se definieron varios intervalos en los cuales se le otorga un valor cualitativo a los resultados obtenidos de la aplicación de las métricas expresados de forma cuantitativa. Con el objetivo de proporcionar al evaluador una forma fácil y rápida de realizar el cálculo de las métricas se diseñó una hoja de cálculo en Excel que realiza automáticamente el cálculo de los indicadores de manera individual, además del cálculo final con su evaluación cualitativa.

Con el objetivo de solucionar la deficiencia encontrada en la influencia de los pesos en la expresión de la calidad total, se definió la obtención de la evaluación mediante el cálculo de la calidad ideal para ese proyecto teniéndose en cuenta los pesos asignados a cada indicador por el diseñador de BD según sus perspectivas y asumiendo que cada indicador obtiene el máximo valor. Con el resultado obtenido se establece una relación con el valor real de la calidad total y el resultado se expresa finalmente en escala de 24 puntos como máxima puntuación posible.

### **3.2.7. Definición de propuesta final**

Para obtener la propuesta final se tuvieron en cuenta varios aspectos: el análisis realizado al comportamiento de cada una de las métricas en el proceso de prueba y la opinión emitida por los especialistas a través de encuestas. Se realizó una recopilación de las más eficientes, es decir, las que obtuvieron mejores resultados en la prueba y de esta forma quedó conformada la propuesta final.

### 3.2.7.1. Propuesta final

Las características que se van a tener en cuenta están incluidas en la tabla siguiente que las describe:

Característica	Fórmula	Descripción
<b>Legibilidad</b>	$Legibilidad = \frac{1}{\frac{NFI}{e^{0.5*NE}}}$	NFI: # de fuentes de ilegibilidad. NE: número de estructuras.
<b>Compleitud</b>	$Compleitud = \left[ \frac{RE}{RT} \right]^2$	RE: # de requisitos de la especificación expresados en el esquema. RT: # de requisitos totales de la especificación.
<b>Correctitud Sintáctica</b>	$Correct\_Sin = \frac{1}{ErrSin + 1}$	ErrSin: # de errores sintácticos
<b>Correctitud Semántica</b>	$Correct\_Sem = 1 - \frac{RNC}{RT}$	RT: # total de requisitos presentes en la especificación. RNC: # de requisitos en la especificación, representados de manera errónea en el esquema.
<b>Minimalidad</b>	$Minimalidad = \left[ 1 - \frac{FR}{A} \right]^2$	FR: # de fuentes de redundancia. A: # de atributos del esquema.
<b>Expresividad</b>	$Expresividad = 1 - \frac{RI}{RE}$ $RE = RA + RI + RNC$	RA: # de requisitos acertados. RI: # de requisitos presentes en el esquema, pero que no se encuentran en la especificación. RNC: # de requisitos de la especificación, representados de manera errónea en el esquema.
<b>Autoexplicación</b>	$Autoexp = 1 - \frac{NS + NR}{RT}$	NS: Información anexa al esquema. RT: # de requisitos totales. NR: Número de restricciones no modeladas

<b>Consistencia</b>	$Consistencia = \frac{1}{NI + 1}$	NI: # de inconsistencias del esquema
---------------------	-----------------------------------	--------------------------------------

Tabla 16. Propuesta Final

La **calidad del modelo** (QDB<sup>1</sup>) se calcularía a partir de los valores de las características anteriores de la siguiente manera:

$$QDB = w_1 * Legibilidad + w_2 * Completitud + w_3 * CorrectSin + w_4 * CorrectSem + w_5 * Minimalidad + w_6 * Expresividad + w_7 * Autoexp + w_8 * Consistencia.$$

W<sub>i</sub> i=1...8 son los pesos asociados a cada criterio de calidad según su importancia.

Donde (0 <= QDB <= 24) y (1 <= W<sub>i</sub> <= 3)

Teniendo en cuenta los pesos asignados por los diseñadores a la importancia de cada indicador, se define el máximo valor posible que sería el ideal para ese proyecto según sus expectativas. Posteriormente se lleva el resultado obtenido QDB<sub>p</sub> a una escala de 24, que no es más que el mayor valor definido en los intervalos estándares de evaluación final.

Rango	Deficiente	Adecuado	Eficiente
0 <= QDB <= 8	X		
9 <= QDB <= 16		X	
17 <= QDB <= 24			X

Tabla 17 Rango de evaluación de resultados

### 3.2.8. Validación empírica de la propuesta

#### 3.2.8.1. Aplicación de propuesta final en Akademos

Característica	Fórmula	Descripción	Valores	Resultado
<b>Legibilidad</b>	$Legibilidad = \frac{1}{\frac{NFI}{e^{0.5 * NE}}}$	NFI: # de fuentes de ilegibilidad. NE: número de estructuras.	NFI = 0 NE = 52	1
<b>Completitud</b>	$Completitud = \left[ \frac{RE}{RT} \right]^2$	RE: # de requisitos de la especificación expresados en el	RE = 51 RT = 51	1

<sup>1</sup> Quality Database

		esquema. RT: # de requisitos totales de la especificación.		
<b>Correctitud Sintáctica</b>	$Correct\_Sin = \frac{1}{ErrSin + 1}$	ErrSin: # de errores sintácticos	ErrSin = 0	1
<b>Correctitud Semántica</b>	$Correct\_Sem = 1 - \frac{RNC}{RT}$	RT: # total de requisitos presentes en la especificación. RNC: # de requisitos en la especificación, representados de manera errónea en el esquema.	RT= 12 RNC = 0	1
<b>Minimalidad</b>	$Minimalidad = \left[ 1 - \frac{FR}{A} \right]^2$	FR: # de fuentes de redundancia. A: # de atributos del esquema.	FR = 0 A = 239	1
<b>Expresividad</b>	$Expresividad = 1 - \frac{RI}{RE}$ $RE = RA + RI + RNC$	RA: # de requisitos acertados. RI: # de requisitos presentes en el esquema, pero que no se encuentran en la especificación. RNC: # de requisitos de la especificación, representados de manera errónea en el esquema.	RA= 51 RI = 0 RNC = 0 RE = 51	1
<b>Autoexplicación</b>	$Autoexp = 1 - \frac{NS + NR}{RT}$	NS: Información anexa al esquema. RT: # de requisitos totales. NR: Número de restricciones no modeladas	NS = 0 RT = 51 NR = 0	1
<b>Consistencia</b>	$Consistencia = \frac{1}{NI + 1}$	NI: # de inconsistencias del esquema	NI = 1	0.5

Tabla 18 Propuesta final en Akademos

**Calidad del modelo:**

$$QDB = w_1 * Legibilidad + w_2 * Completitud + w_3 * CorrectSin + w_4 * CorrectSem + w_5 * Minimalidad + w_6 * Expresividad + w_7 * Autoexp + w_8 * Consistencia.$$

Donde  $(0 \leq QDB \leq 24)$  y  $(1 \leq W_i \leq 3)$

$W_i$   $i=1...8$  son los pesos asociados a cada criterio de calidad según su importancia.

Los pesos asignados por los diseñadores según su el grado de importancia que le atribuyen en dependencia de las características del proyecto a los indicadores son:

Indicadores	Pesos
Legibilidad	1
Completitud	3
Correctitud Sintáctica	2
Correctitud Semántica	2
Minimalidad	3
Expresividad	1
Autoexplicación	2
Consistencia	3

$$QDB = w_1 * Legibilidad + w_2 * Completitud + w_3 * CorrectSin + w_4 * CorrectSem + w_5 * Minimalidad + w_6 * Expresividad + w_7 * Autoexp + w_8 * Consistencia.$$

$$QDB = 15.5, \text{ (Valor primario de } QDB_p)$$

Teniendo en cuenta los pesos asignados a la importancia de cada indicador, podemos definir que el máximo valor posible a obtener en el proyecto Akademos es 17, y el valor que arrojó el cálculo de la calidad total del diseño fue de 15.5. Llevando este resultado a una escala de 24 tenemos que la calidad total del diseño del proyecto Akademos es de 21.88 y tiene una evaluación de Eficiente.

Rango	Deficiente	Adecuado	Eficiente
$0 \leq QDB \leq 8$			
$9 \leq QDB \leq 16$			
$17 \leq QDB \leq 24$			<b>X</b>

Tabla 19. Evaluación de Akademos



### 3.2.8.2. Aplicación de propuesta final en Servicios Comunitarios

Característica	Fórmula	Descripción	Valores	Resultado
<b>Legibilidad</b>	$Legibilidad = \frac{1}{\frac{NFI}{e^{0.5*NE}}}$	NFI: # de fuentes de ilegibilidad. NE: número de estructuras.	NFI = 0 NE = 25	1
<b>Compleitud</b>	$Compleitud = \left[ \frac{RE}{RT} \right]^2$	RE: # de requisitos de la especificación expresados en el esquema. RT: # de requisitos totales de la especificación.	RE= 11 RT= 12	0.83
<b>Correctitud Sintáctica</b>	$Correct\_Sin = \frac{1}{ErrSin + 1}$	ErrSin: # de errores sintácticos	ErrSin = 0	1
<b>Correctitud Semántica</b>	$Correct\_Sem = 1 - \frac{RNC}{RT}$	RT: # total de requisitos presentes en la especificación. RNC: # de requisitos en la especificación, representados de manera errónea en el esquema.	RT= 12 RNC = 1	0.92
<b>Minimalidad</b>	$Minimalidad = \left[ 1 - \frac{FR}{A} \right]^2$	FR: # de fuentes de redundancia. A: # de atributos del esquema.	FR = 0 A = 108	1
<b>Expresividad</b>	$Expresividad = 1 - \frac{RI}{RE}$ $RE = RA + RI + RNC$	RA: # de requisitos acertados. RI: # de requisitos presentes en el esquema, pero que no se encuentran en la especificación. RNC: # de	RA= 11 RI = 1 RNC = 1 RE = 13	0.92

		requisitos de la especificación, representados de manera errónea en el esquema.		
<b>Autoexplicación</b>	$Autoexp = 1 - \frac{NS + NR}{RT}$	NS: Información anexa al esquema. RT: # de requisitos totales. NR: Número de restricciones no modeladas	NS = 0 RT = 12 NR = 0	1
<b>Consistencia</b>	$Consistencia = \frac{1}{NI + 1}$	NI: # de inconsistencias del esquema	NI = 1	0.5

Tabla 20. Propuesta final en Servicios Comunitarios

**Calidad del modelo:**

$$QDB = w_1 * Legibilidad + w_2 * Completitud + w_3 * CorrectSin + w_4 * CorrectSem + w_5 * Minimalidad + w_6 * Expresividad + w_7 * Autoexp + w_8 * Consistencia.$$

Donde (0 ≤ QDB ≤ 24) y (1 ≤ W<sub>i</sub> ≤ 3)

W<sub>i</sub> i=1...8 son los pesos asociados a cada criterio de calidad según su importancia.

Los pesos asignados por los diseñadores según su el grado de importancia que le atribuyen en dependencia de las características del proyecto a los indicadores son:

Indicadores	Pesos
Legibilidad	1
Completitud	3
Correctitud Sintáctica	2
Correctitud Semántica	2
Minimalidad	3
Expresividad	1
Autoexplicación	2
Consistencia	3

$$QDB = w_1 * Legibilidad + w_2 * Completitud + w_3 * CorrectSin + w_4 * CorrectSem + w_5 * Minimalidad + w_6 * Expresividad + w_7 * Autoexp + w_8 * Consistencia.$$

$$QDB = 14.78, \text{ (Valor primario de } QDB_p)$$

Teniendo en cuenta los pesos asignados a la importancia de cada indicador, podemos definir que el máximo valor posible a obtener en el proyecto Servicios Comunitarios es 17, y el valor que arrojó el cálculo de la calidad total del diseño fue de 14.78. Llevando este resultado a una escala de 24 tenemos que la calidad total del diseño del proyecto Akademos es de 20.86 y tiene una evaluación de Eficiente.

Rango	Deficiente	Adecuado	Eficiente
$0 \leq QDB \leq 8$			
$9 \leq QDB \leq 16$			
$17 \leq QDB \leq 24$			X

Tabla 21. Evaluación de Servicios Comunitarios

El valor de la calidad del modelo es 14.75 lo cual indica que el diseño se adecúa al problema que lo originó, pasando por un proceso de mejora pudiera llegar a ser eficiente.

### 3.2.9. Evaluación de los resultados

Después de haber realizado la prueba en ambos casos los resultados fueron adecuados lo que significa que se pudiera mejorar el diseño pero que cumple con las especificaciones.

### 3.3. Validación teórica

Para la validación teórica de la propuesta se utilizó el método multicriterio, éste se basa en la evaluación cuantitativa de criterios previamente definidos por parte de expertos en el tema, que permite determinar si se acepta o no la propuesta analizada. Para esto se seleccionó un conjunto de especialistas en las principales áreas que abarca la investigación: calidad de software y BD. Los entrevistados fueron especialistas de la dirección de calidad de la universidad, profesores con gran nivel de experiencia como diseñadores de BD y diversos especialistas con alto grado científico en el tema.

### 3.4. Guía para la evaluación técnica de la propuesta

A continuación se describen los pasos que se efectuaron para llevar a cabo la evaluación utilizando el método multicriterio (26):

**Paso #1:** Se elaboran los criterios que fueron utilizados en la evaluación y se agrupan de acuerdo a las características de la propuesta.

**Grupo No 1: Criterios de mérito científico**

1. Valor científico de la propuesta.
2. Calidad de la investigación.
3. Contribución científica.
4. Responsabilidad científica y profesionalidad de los investigadores.

**Grupo No 2: Criterios de implantación**

5. Necesidad de empleo de la propuesta.
6. Posibilidades de aplicación.

**Grupo No 3: Criterios de flexibilidad**

7. Adaptabilidad a entidades dedicadas a evaluar la calidad de productos de software.
8. Capacidad del proceso de evaluación para la admisión de cambios que impliquen mejoras.

**Grupo No 4. Criterios de impacto**

9. Impacto en el área para la cual está destinada la propuesta.
11. Organización en el proceso de desarrollo.

**Paso #2:** Se le asigna un peso relativo a cada grupo de criterios de acuerdo al porcentaje que representa cada grupo del total y los intereses a evaluar.

Grupo No. 1.....40

Grupo No. 2.....20

Grupo No.3.....20

Grupo No.4.....20

**Paso #3:** Se realiza una selección de 9 expertos en la cual se tiene en cuenta su especialidad, grado científico y currículo.

**Paso #4:** Se hace entrega de la propuesta que se desea validar a todos los expertos para que se documenten sobre el tema de la investigación y luego expresen sus criterios. En la encuesta de evaluación de la propuesta (Ver Anexo 6), los expertos conceden pesos a cada uno de los criterios establecidos, teniendo en cuenta que la suma de los valores dados para un grupo no exceda del peso relativo asignado a este. Además la encuesta permite realizar una evaluación cualitativa de la clasificación final del proyecto en excelente, bueno, aceptable, cuestionable y

malo. También se ofrece la posibilidad de dar su opinión haciendo una valoración final del proyecto, emitiendo todas aquellas consideraciones que estimen convenientes.

**Paso #5:** Después de recibir los valores del peso relativo de cada criterio se construye la tabla de los pesos otorgados.

Grupo	C/E	E1	E2	E3	E4	E5	E6	E7	E8	E9	Ep
1	C1	9	10	10	10	8	9	10	7	8	9
	C2	10	10	9	7	7	9	10	9	9	8.89
	C3	8	8	9	7	6	10	9	7	8	8
	C4	10	7	10	10	9	9	10	9	10	9.33
2	C5	10	10	8	10	10	10	10	10	9	9.67
	C6	10	6	9	7	9	10	10	9	9	8.78
3	C7	10	10	10	10	10	10	9	8	8	9.44
	C8	10	10	10	10	6	9	9	8	9	9
4	C9	10	10	10	10	10	9	10	9	8	9.55
	C10	10	10	10	10	10	9	10	9	9	9.67
Total		97	91	95	91	85	94	97	85	87	91.33

Tabla 22 Pesos otorgados por los expertos

**Paso #6:** Se verifica la consistencia en el trabajo de los expertos, para lo que se utiliza el coeficiente de concordancia de Kendall y el estadígrafo Chi cuadrado (X<sup>2</sup>). Siguiendo el procedimiento que se muestra a continuación.

- Sea C el número de criterios que van a evaluarse y E el número de expertos que realizan la evaluación.
- Para cada criterio se determina la  $\Sigma E$  que representa la sumatoria del peso dado por cada experto, Ep que es la puntuación promedio de los pesos correspondientes a cada criterio.
- Se determina la desviación de la media, que posteriormente se eleva al cuadrado para obtener la dispersión S por la expresión:

$$S = \sum (\Sigma E - \sum \sum E/C)^2$$

	$\Sigma E$	$\Sigma E/C$	$\Sigma E - \Sigma E/C$	$(\Sigma E - \Sigma E/C)^2$
C1	81	8.1	-1.2	1.44
C2	80	8.0	-2.2	4.84
C3	72	7.2	-10.2	104.04
C4	84	8.4	1.8	3.24
C5	87	8.7	4.8	23.04
C6	79	7.9	-3.2	10.24
C7	85	8.5	2.8	7.84
C8	81	8.1	-1.2	1.44
C9	86	8.6	3.8	14.44
C10	87	8.7	4.8	23.04
$\Sigma \Sigma E/C$	-	82.2		
$S = \Sigma(\Sigma E - \Sigma \Sigma E/C)^2$	-	-	-	193.6

Tabla 23. Cálculo de la dispersión

- Conociendo la dispersión se puede calcular el coeficiente de concordancia de Kendall W:

$$W = \frac{S}{\frac{E^2(C^3 - C)}{12}}$$

$$W = 0.028$$

- El coeficiente de concordancia de Kendall permite calcular el Chi cuadrado real:

$$X^2 = E * (C - 1) * W$$

$$X^2 = 2.26$$

El  $X^2_{real}$  es 2.26, para seleccionar el  $X^2$  de la tabla de Distribución Chi Cuadrado, se toma  $1-\alpha=0.99$  dónde  $\alpha$  es el error permisible, entonces  $\alpha=0.01$ .

Debe cumplirse que:  $X^2_{real} < X^2_{(\alpha, c-1)}$ .

El cálculo arrojó como resultado:  $2.26 < 21.66$ , por lo que se llega a la conclusión de que existe concordancia entre los expertos y se puede pasar a la construcción de la tabla de clasificación de cada criterio para saber el índice de aceptación que tuvo la propuesta. Si no existe concordancia se hace necesario repetir el trabajo de los expertos.

**Paso #7:** Posteriormente se identifica el peso relativo de cada criterio P y se calcula el Índice de Aceptación (IA) de la propuesta.

Para esto se utiliza el procedimiento siguiente:

Conociendo el número de expertos que realizan la evaluación E y la sumatoria de las puntuaciones de cada criterio C se puede calcular el peso de cada criterio P.

Conociendo el peso de cada criterio P y la calificación dada por los evaluadores c en una escala de 1 a 5 que se recogieron en la encuesta de evaluación, se puede calcular el valor de  $P \times c$ .

Con el valor anterior se calcula el Índice de Aceptación del proyecto (IA).

$$IA = P * c / 5$$

Criterio	Calificación (c)					P	P x c
	1	2	3	4	5		
CE1					X	0.098	0.49
CE2				X		0.097	0.388
CE3				X		0.087	0.348
CE4					X	0.1021	0.510
CE5					X	0.1058	0.529
CE6				X		0.096	0.384
CE7					X	0.1033	0.516
CE8					X	0.098	0.49
CE9					X	0.1045	0.522
CE10					X	0.1058	0.529

Tabla 24. Cálculo de Índice de Aceptación

Se calcula el índice de aceptación:

$$IA = P * c / 5$$

$$IA = 0.9412$$

**Paso #8:** Por último se determina la probabilidad de éxito de la propuesta, ubicando el IA calculado anteriormente en rangos que están predefinidos en dependencia de donde se ubique será la probabilidad de éxito que tenga la propuesta.

<b>0.7 &lt; IA</b>	Existe alta probabilidad de éxito
<b>0.5 &lt; IA &lt; 0.7</b>	Existe probabilidad media de éxito
<b>0.3 &lt; IA &lt; 0.5</b>	Probabilidad de éxito baja
<b>IA &lt; 0.3</b>	Fracaso seguro

Tabla 25 Interpretación del índice de aceptación

El IA fue de 0.9412 por lo que se puede afirmar que existe una alta probabilidad de éxito.

### 3.5. Principales valoraciones y sugerencias de los expertos

Como parte de la validación técnica de la propuesta los expertos emitieron una valoración final, así como sus principales sugerencias para mejorar la calidad de la propuesta. A modo de resumen los criterios fueron:

- Se debe extender las métricas a otro tipo de BD.
- Es necesario ampliar el conjunto de métricas y sus áreas de evaluación.
- Integrarlas con métricas de seguridad aplicadas a estos entornos.
- Es muy importante que el desarrollo del modelo de evaluación de métricas sobre base datos , tema que es donde se encuentran a veces los principales problemas en el código debido a inconsistencias y deficiente normalización de los datos , se pueda trabajar en implementar este modelo en los proyectos de desarrollo y sobre todo en el grupo de BD.
- Se considera que el trabajo tiene una muy buena calidad y puede ser llevada a cabo su implantación en los diferentes proyectos de la universidad lo que permitirá lograr una mayor calidad en los diseños de sus BD los cuales son el corazón de los proyectos productivos.



### Valoración Final de los Expertos

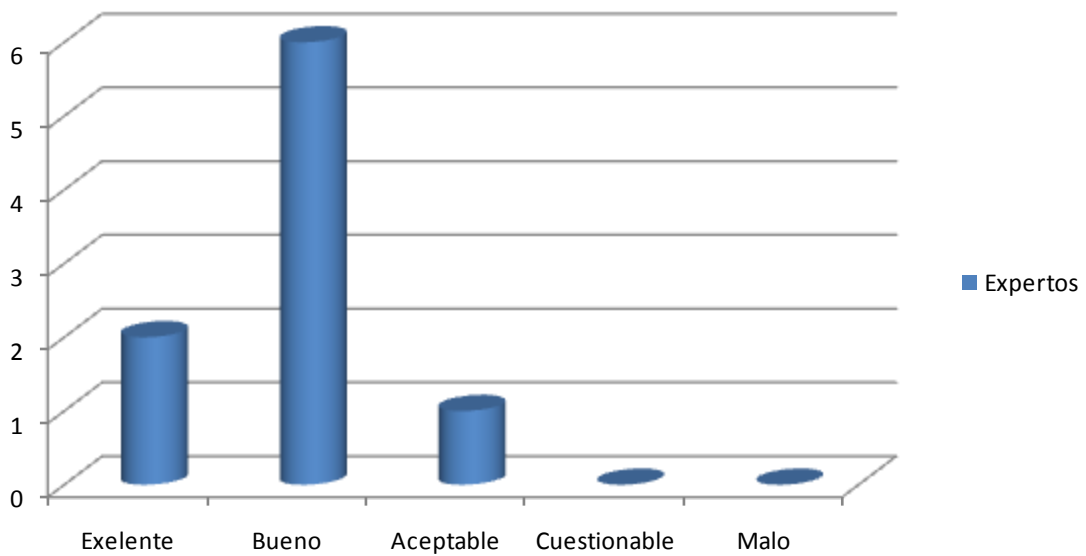


Fig 10. Valoración de Expertos

### 3.6. Conclusiones

Para la validación de la propuesta se realizaron principalmente dos actividades: la validación empírica y validación teórica. En la validación empírica se aplicaron las métricas en proyectos de la facultad obteniendo resultados satisfactorios. En este proceso se llevo a cabo un refinamiento de las métricas donde se obtuvo la propuesta final. Para la validación técnica se utilizó el método multicriterio cuyos resultados fueron favorables, obteniéndose una probabilidad de éxito alta demostrando que lo planteado hasta el momento se adapta valiosamente a las condiciones existentes y aporta novedosos elementos que resultan imprescindible para la práctica de la aplicación de algo tan complejo y en ocasiones confuso como las métricas.

## **Conclusiones Generales**

A partir del trabajo realizado se obtuvo la propuesta final de las métricas, la cual recoge un conjunto de métricas que permiten evaluar las BD relacionales. Para ello se realizó una exhaustiva investigación sobre calidad, BD y métricas dirigidas a la evaluación de las mismas en la cual se ha arribado a las siguientes conclusiones:

- Se identificó un conjunto de métricas para la evaluación de las BD relacionales.
- Se diseñó un proceso para identificar, ajustar y validar empíricamente una métrica.
- Se aplicó el proceso de validación y se propuso un conjunto de métricas.
- El conjunto obtenido se validó por vía empírica y por expertos.
- Se obtuvo una herramienta que facilita el cálculo de QDB.

Se puede afirmar que se cumplieron todos los objetivos trazados a lo largo de la investigación.

## Recomendaciones

Este trabajo de diploma propuso un conjunto de métricas para la evaluación integral de BD relacionales donde se recomienda:

- Proponer la utilización de las mismas en los proyectos productivos para refinar aun más la propuesta.
- Desarrollar herramientas que automaticen el cálculo de las métricas y el almacenamiento de sus resultados, además de permitir generar estadísticas sobre el comportamiento de las BD a lo largo del ciclo de vida del proyecto.
- Profundizar el estudio de las métricas asociadas a los factores implementabilidad y portabilidad.
- Realizar talleres, seminarios, conferencias donde se refleje claramente la importancia de la aplicación de métricas en el proceso de evaluación del software.

## Referencias Bibliográficas

1. **Ing. Ludisley la Torre Hernández.** Monografias.com. [En línea] 2006.  
<http://www.monografias.com/trabajos55/proceso-de-desarrollo-software/proceso-de-desarrollo-software2.shtml>.
2. **Alarcos, Grupo.** *Métricas para la evaluación de la complejidad de BDRelacionales.* 2000.
3. **Andrés, María Mercedes Marqués.** Ciclo de vida de los sistemas de información . [En línea] 2001.  
<http://www3.uji.es/~mmarques/f47/apun/node66.html>.
4. **Anónimo.** catardockear.webnode.com. [En línea] 2008.  
<http://catardockear.webnode.com/news/definiciones-basicas/>.
5. **Caballero, Dr. Ismael.** *Calidad y Medición de Sistemas de Información.* 2008.
6. **Desconocido.** *Conceptos básicos de Métricas.* 2008.
7. —. slideshare.net. [En línea] 2008. <http://www.slideshare.net/rmonago/t1-gestion-de-datos-presentation>.
8. —. Wikipedia. [En línea] 2008. [http://es.wikipedia.org/wiki/Sistema\\_inform%C3%A1tico#cite\\_note-0](http://es.wikipedia.org/wiki/Sistema_inform%C3%A1tico#cite_note-0).
9. **Graells, Dr. Pérez Márquez.** Monografias.com. *Monografias.com.* [En línea] 2003.  
<http://www.monografias.com/trabajos35/manual-base-datos/manual-base-datos.shtml>.
10. **Jose Rico Herreros, Rogelio Valero Ros, Roberto Lopez Arrufat, Jose Edo Butiña.** *CLASIFICACIÓN DE LOS SISTEMAS SGBD.* 2008.
11. **Ludisley, Mariela y.** *PROPUESTA DE MÉTRICAS PARA PERFECCIONAR LA GESTIÓN DE LA CALIDAD EN LOS PROCESOS DE DESARROLLO DE SOFTWARE.* . 2007.
12. **Marcela Genero, Luis Jiménez, and Mario Piattini1.** *Measuring the Quality of Entity Relationship Diagrams.* 2000.
13. **Marcela Varas, Jaime Pradenas.** *Hacia la Definición de Métricas de Calidad para Esquemas Conceptuales de Bases de Datos.* 2001.
14. **MARIO PIATTINI, MARCELA GENERO and CORAL CALERO.** *DATA MODEL METRICS.* 2002.
15. **Martinez, Janhil Aurora Trejo.** [En línea] 2007.  
<http://www.monografias.com/trabajos11/basda/basda.shtml>.

16. **Pacheco, Henry Jesús Mendoza.** Monografias.com. [En línea] 2007.  
<http://www.monografias.com/trabajos55/analisis-sistemas-informacion/analisis-sistemas-informacion3.shtml>.
17. **PIATTINI VELTHUIS, M.** [En línea] 2007.  
<http://www.uylibros.com/verlibro.asp?xprod=0&id=19056&idcat=24&idsubcat=41&idarea=1226>.
18. **Raga, Charlis.** Monografias.com. [En línea] 2008.  
<http://www.monografias.com/trabajos7/bada/bada.shtml>.
19. **Salamanca, Jorge.** *CALIDAD DEL SOFTWARE.*
20. **Valdés, Damián Pérez.** [En línea] 2007.  
<http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/>.
21. **Varas, Marcela.** *Diseño Conceptual de Bases de Datos: Un enfoque basado en la medicion de la calidad.* 2002.
22. **Yeneirys Hernández Castellanos, Wendysh Pérez García.** *Diseño de BD para la Intranet 2.* Habana : s.n., 2007.
23. **Yudisleidys Peña Lemus, Yunierys Hernández Díaz.** *SIMETSE – Sistema de METricas para evaluar el Software Educativo.* 2007.
24. **MARQUÉS, M. M.** Metodología de diseño de bases de datos. [En línea] 2001.  
<http://www3.uji.es/~mmarques/f47/apun/node81.html>.
25. **Hinojosa, Lizzet Cabrera Montoya Virgen Yuliet Acosta.** *MÉTRICAS ESTANDARIZADAS INTERNACIONALMENTE, PROPUESTAS PARA EVALUAR LA CALIDAD DE LOS PRODUCTOS DE SOFTWARE .* 2008.
26. **Daismary Oliva Agüero, Olga Esther Cruz Pichardo.** *Metricas para la evaluacion del proceso de desarrollo del software educativo.* 2008.

## Bibliografía Consultada

1. **C. Batini, S. Ceri, S.B. Navathe.** *"Diseño Conceptual de Bases de Datos. Un enfoque de entidades-interrelaciones."*. 1994.
2. **E, Piattini F and García.** *"Calidad en el desarrollo y mantenimiento del Software"*. s.l. : Editorial RA-MA, Madrid, 2002
3. **Hernandez, and Minguet. M.J.** *"La Calidad del Software y su medida"*. s.l. : Editorial CERASA , 2003.
4. **Kesh, Someswar.** *Evaluating the quality of entity relationship models.* 2002.
5. **LÓPEZ, C.** Mejoramiento continuo principio de Gestión de la Calidad. [En línea] 2007.  
<http://www.gestiopolis.com/canales5/ger/gksa/136.htm>.
6. **Marcela Genero, Luis Jiménez, and Mario Piattini**<sup>1</sup>. *Measuring the Quality of Entity Relationship Diagrams.* 2000.
7. **MARIO PIATTINI, MARCELA GENERO and CORAL CALERO.** *DATA MODEL METRICS.* 2002.
8. **Miguel, A. y Piattini.** *"Fundamentos y modelos de bases de Datos"*. 1999.
9. **Mario Piattini, Coral Calero, Houari Sahraoui, Hakim Lounis.** *Object-relational database metrics.* 2002.
10. **Norick, Ro y.** "Diseño de BDRelacionales". [En línea] 2003. <http://usuarios.lycos.es/cursosgbd>.
11. **Orallo, José Hernández.** "La disciplina de los sistemas de bases de datos. Historia, Situación actual y perspectiva". [En línea] 2002.  
<http://www.dsic.upv.es/~jorallo/docent/BDA/DisciplinaBD.pdf>.
12. **PRESSMAN, R. S.** *Ingeniería del Software. Un Enfoque Práctico.* 1998.
13. **Sitio del Ministerio de Relaciones Exteriores de Cuba.** La informatización en Cuba. [En línea] [http://www.cubaminrex.cu/Sociedad\\_Informacion/Cuba\\_SI/Informatizacion.htm](http://www.cubaminrex.cu/Sociedad_Informacion/Cuba_SI/Informatizacion.htm). 2005.

# Glosario de Términos y Siglas

**Autoexplicación:** Un esquema se autoexplica si todos los requerimientos pueden ser modelados a través de los conceptos del modelo sin la ayuda de otros formalismos (ej.: lógica, lenguaje natural). Se puede tener acceso al mundo modelado solamente a través del esquema.

**Base de datos:** Conjunto no redundante de información almacenada en memoria organizada independientemente de su utilización y su implementación en máquinas accesibles en tiempo real y compatibles con usuarios concurrentes con necesidad de información diferente y no predicable en tiempo.

**Calidad:** La calidad es un conjunto de características de una entidad que le confiere la aptitud para satisfacer las necesidades establecidas y las implícitas.

**Calidad de software:** Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados, y con las características implícitas que se espera de todo software desarrollado profesionalmente.

**Certificación:** Garantía que asegura la certeza o autenticidad de algo.

**Compleción:** Se trata del esquema que incluye la representación total del mundo o la realidad diseñada, esto a través de los requerimientos impuestos y restricciones, es decir, que exista una correspondencia uno a uno de los requerimientos con el esquema y también viceversa, además no deben existir requerimientos no formulados (inexistentes).

**Consistencia:** Un esquema de BD es consistente, si existe una instanciación permitida (no vacía) para cada elemento del esquema.

**Control de calidad:** El control está dirigido al cumplimiento de requisitos, no es más que un conjunto de actividades y técnicas operativas utilizadas para verificar los requerimientos relativos a la calidad del producto o servicio.

**Corrección:** Un esquema cumple con esta característica cuando usa correctamente los conceptos y definiciones del lenguaje (por ejemplo el modelo entidad interrelación) ya sea sintáctica como semánticamente.

**Estándar de calidad:** Compromiso respecto al mantenimiento de un nivel de calidad en el contenido de sus prestaciones por parte de una unidad administrativa.

**Evaluación:** Es un proceso de información, interpretación y valoración para la toma de decisiones y para la mejora.

**Evaluación cualitativa:** Cada fenómeno es considerado como algo único que debe ser analizado en su ambiente natural y con la utilización de procedimientos e instrumentos que permitan captarlos en su integridad.

**Evaluación cuantitativa:** Procedimientos que requieren medir y cuantificar los fenómenos para describir causas y efectos y explicar relaciones entre variables independientes (tratamiento) y variables dependientes (resultados).

**Experto:** Persona que aporta conocimientos o experiencia específica con respecto a una organización, proceso, actividad o materia que se vaya a auditar.

**Expresividad:** La expresividad se enfoca a la naturalidad con que se expresan los requerimientos. Se diferencia de la legibilidad en que aquí el esquema es de fácil comprensión, pero desde el punto de vista de lo que representa.

**Extensibilidad:** Se refiere a la capacidad de un esquema para poder tolerar cambios en los requerimientos. Se consideran extensibles aquellos esquemas que pueden descomponerse en módulos o partes. Un esquema "modular", en el sentido recién expuesto, es más escalable, pudiendo incorporarse nuevos elementos alterando levemente los existentes.

**Garantía de calidad:** Conjunto de acciones planificadas y sistemáticas necesarias para proporcionar la confianza adecuada de que un producto o servicio satisface los requerimientos dados sobre calidad, o sea tiene como finalidad inspirar confianza en que se cumplirá el requisito pertinente.

**Gestión de la calidad:** La gestión de la calidad es un conjunto de actividades y medios necesarios para definir e implantar un sistema de la calidad, responsabilizarse de su control, aseguramiento y mejora continua.

**Indicador:** Magnitud utilizada para medir o comparar los resultados efectivamente obtenidos en la ejecución de un trabajo. Es, por tanto, un resultado cuantitativo que se mide en porcentaje, tasas y razones para permitir comparaciones.

**Legibilidad:** Se refiere a la estética del esquema. Un esquema posee esta cualidad cuando puede leerse con facilidad, reconociendo tipos de entidad, tipos de interrelación y atributos, deben existir mínimos cruces de líneas siendo estas horizontales y verticales, que deje en sus debidos niveles las jerarquías, agregando además que si existen estructuras simétricas estas deben aparecer como tales. Se recalca que se trata de una fácil lectura no incluyendo la sencillez en la "comprensión" del esquema.



**Métrica:** asignación de valor a un atributo de una entidad propia del software, ya sea un producto o un proceso.

**Métricas del software:** es una medida cuantitativa del grado en el que un sistema, un componente o un proceso poseen un determinado atributo.

**Minimalidad:** La característica de minimalidad se refiere a la existencia de información redundante o duplicada en el esquema, es decir, este no es mínimo si existe redundancia. Dicho de otra forma, en un esquema mínimo no se puede borrar del esquema un elemento sin perder alguna información.

**Norma o estándar:** documento aprobado por consenso por un organismo reconocido, que proporciona reglas, pautas y/o características para uso común, con el objeto de obtener un óptimo nivel de resultados en un contexto dado.

**Peso:** nivel de importancia asignado a cada una de las actividades que se pretenden medir.

**Procedimiento:** Acción de proceder. Método de ejecutar algunas cosas.

**Proceso:** Conjunto de actividades, realizadas en forma secuencial, que realiza una organización, para crear, producir y entregar productos, de tal manera que satisfagan las necesidades de sus clientes.

**Producto:** Resultado concreto, observable y medible que surge como consecuencia del proceso, proyecto o experiencia desarrollada.

**Sistema de Calidad:** Conjunto de las estructuras, responsabilidades, actividades, recursos y procedimientos de la organización de una empresa, que ésta establece para garantizar que lo que ofrece cumple con las especificaciones establecidas previamente por la empresa y el cliente, asegurando una calidad continua a lo largo del tiempo.

**Validación:** Confirmación mediante el suministro de evidencia objetiva de que se han cumplido los requisitos para una utilización o aplicación específica prevista.

**CMM:** Modelo de Madurez de la Capacidad del Desarrollo del Software.

**CMMI:** Modelo de capacidad de madurez integral

**ICSW:** Industria Cubana del Software.

**IEEE:** Institute of Electrical and Electronics Engineers.

**ISO:** Organización Internacional para la Estandarización.

**PSP:** Personal Software Process.

**SGBDR:** Sistemas Gestores de BDRelacionales.

**SGBD:** Sistemas Gestores de Bases de Datos.

**TIC:** Tecnologías de la Informática y las Comunicaciones.

**TSP:** Team Software Process.

# Anexos

## Anexo #1: Encuesta Aplicada a Jefe de Proyecto

Facultad: \_\_\_\_\_ Proyecto: \_\_\_\_\_

1. Tienen definido algún Plan de Pruebas:

Sí\_\_ No\_\_

Explique brevemente \_\_\_\_\_

2. ¿Se usa algún estándar para el plan de pruebas?

Si\_\_ No\_\_ No sé\_\_

De utilizarse diga:

2.1. ¿Cuáles? \_\_\_\_\_

2.2. ¿El plan de pruebas es revisado con anterioridad por los responsables de su ejecución y luego por el equipo de prueba?

Si\_\_ No\_\_ No sé\_\_

2.3 Se lleva a cabo alguna estrategia de prueba que especifique ¿como los objetivos serán alcanzados?

Si\_\_ No\_\_ No sé\_\_

2.4. En que Niveles de Pruebas están desarrollando las mismas:

\_\_\_ Unidad (proceso de verificación en la menor unidad del diseño del software )

\_\_\_ Integración (para construir la estructura del programa)

\_\_\_ Sistema (integración del software en el sistema)

\_\_\_ Aceptación (Verificar que el software esté listo)

2.5. La planeación y ejecución de pruebas se realiza en paralelo con el proceso de desarrollo de software:

Sí\_\_ No\_\_

En caso de no, explicar cómo se realizan.

\_\_\_\_\_

3. ¿Se utiliza algún proceso o estándar para el diseño de las bases de datos?

Si\_\_\_ No\_\_\_

4. ¿Conoce alguna métrica para medir la calidad de un diseño de Base de Datos?

Si\_\_\_ No\_\_\_

¿Cual?:\_\_\_\_\_

5. ¿Se aplica alguna métrica para la medición de la calidad del producto obtenido o en el proceso de obtención del mismo?

Si\_\_\_ No\_\_\_

¿Cual?:\_\_\_\_\_

En caso que si responda:

5.1. ¿En qué medida considera usted que estas métricas garantizan calidad?

Total\_\_\_ Parcial\_\_\_ Ninguna\_\_\_

En caso que responda no:

5.2. ¿Cree que se deba utilizar alguna métrica en su proyecto?

Si\_\_\_ No\_\_\_

5.2.1 En caso que responda Si:

Cuáles de estas opciones cree que se ajuste a las características del proyecto:

\_\_\_ Métricas técnicas: Se centran en las características de software pro ejemplo: la complejidad lógica, el grado de modularidad. Mide la estructura del sistema, el cómo está hecho.

\_\_\_ Métricas de calidad: proporcionan una indicación de cómo se ajusta el software a los requisitos implícitos y explícitos del cliente. Es decir cómo voy a medir para que mi sistema se adapte a los requisitos que me pide el cliente.

\_\_\_ Métricas de productividad: Se centran en el rendimiento del proceso de la ingeniería del software. Es decir que tan productivo va a ser el software que voy a diseñar.

\_\_\_ Métricas orientadas a la persona: Proporcionan medidas e información sobre la forma que la gente desarrolla el software de computadoras y sobre todo el punto de vista humano de la efectividad de las herramientas y métodos. Son las medidas que voy a hacer de mi personal que va hará el sistema.

\_\_\_ Métricas orientadas al tamaño: Es para saber en qué tiempo voy a terminar el software y cuantas personas voy a necesitar. Son medidas directas al software y el proceso por el cual se desarrolla, si una organización de software mantiene registros sencillos, se puede crear una tabla de datos orientados al tamaño.

\_\_\_ Métricas orientadas a la función: Son medidas indirectas del software y del proceso por el cual se desarrolla. En lugar de calcularlas las LDC, las métricas orientadas a la función se centran en la funcionalidad o utilidad del programa.

Diga el por qué de su elección y que factores considera importante en el diseño de métricas para la obtención de una mayor calidad de las BD de su proyecto.

---

---

6. ¿Se le da seguimiento y se reportan los defectos encontrados?:

Sí\_\_ No\_\_ No sé\_\_

7. ¿El equipo de trabajo define sus objetivos con anterioridad, se define una estrategia de desarrollo, y se produce un plan general de desarrollo?:

Sí\_\_ No\_\_ No sé\_\_

Explique brevemente

---

---

8. ¿El equipo de trabajo define el proceso, se reparten correctamente los roles y se lleva a cabo un análisis de riesgos?:

Sí\_\_ No\_\_ No sé\_\_

Explique brevemente. \_\_\_\_\_

---

9. ¿Son documentados los resultados obtenidos cuando el esfuerzo de prueba es terminado?:

Sí\_\_ No\_\_ No sé\_\_

10. ¿Utilizan alguna herramienta automatizada para gestionar el seguimiento, control y registro de las pruebas?:

Sí\_\_ No\_\_ No sé\_\_

Mencionar las herramientas

---

11. ¿Qué herramienta de modelación se utiliza en el proyecto para modelar las BD? ¿Qué SGBD?

---

12. ¿Qué arquitectura se utiliza en el proyecto para el desarrollo de las Bases de Datos?

---

## Anexo #2: Encuesta Aplicada al Diseñador de Bases de Datos

Facultad: \_\_\_\_\_ Proyecto: \_\_\_\_\_

1. Tienen definido algún Plan de Pruebas para el diseño de las BD:

Sí\_\_ No\_\_

Explique brevemente

\_\_\_\_\_

2. ¿Se usa algún estándar para el plan de pruebas?

Si\_\_ No\_\_ No sé\_\_

De utilizarse diga:

2.1 ¿Cuáles?\_\_\_\_\_

2.2 ¿El plan de pruebas es revisado con anterioridad por los responsables de su ejecución y luego por el equipo de prueba?

Si\_\_ No\_\_ No sé\_\_

2.4 Se lleva a cabo alguna estrategia de prueba que especifique ¿como los objetivos serán alcanzados?

Si\_\_ No\_\_ No sé\_\_

2.5 En que Niveles de Pruebas están desarrollando las mismas:

\_\_\_ Unidad (proceso de verificación en la menor unidad del diseño del software)

\_\_\_ Integración (para construir la estructura del programa)

\_\_\_ Sistema (integración del software en el sistema)

\_\_\_ Aceptación (Verificar que el software esté listo)

2.6 La planeación y ejecución de pruebas se realiza en paralelo con el proceso de desarrollo de software:

Sí\_\_ No\_\_

En caso de no, explicar cómo se realizan.

\_\_\_\_\_

3. ¿Se utiliza algún proceso o estándar para el diseño de las BD?

Si\_\_\_ No\_\_\_

4. ¿Conoce alguna métrica para medir la calidad de un diseño de Base de Datos?

Si\_\_\_ No\_\_\_

¿Cual?: \_\_\_\_\_

5. ¿Se aplica alguna métrica para la medición de la calidad del producto obtenido o en el proceso de obtención del mismo?

Si\_\_\_ No\_\_\_

¿Cual?: \_\_\_\_\_

En caso que si responda:

5.1. ¿En qué medida considera usted que estas métricas garantizan calidad?

Total\_\_\_ Parcial\_\_\_ Ninguna\_\_\_

En caso que responda no:

5.2. ¿Cree que se deba utilizar alguna métrica en su proyecto?

Si\_\_\_ No\_\_\_

5.2.1 En caso que responda Si:

Cuáles de estas opciones cree que se ajuste a las características del proyecto:

\_\_\_ Métricas técnicas: Se centran en las características de software pro ejemplo: la complejidad lógica, el grado de modularidad. Mide la estructura del sistema, el cómo está hecho.

\_\_\_ Métricas de calidad: proporcionan una indicación de cómo se ajusta el software a los requisitos implícitos y explícitos del cliente. Es decir cómo voy a medir para que mi sistema se adapte a los requisitos que me pide el cliente.

\_\_\_ Métricas de productividad. Se centran en el rendimiento del proceso de la ingeniería del software. Es decir que tan productivo va a ser el software que voy a diseñar.

\_\_\_ Métricas orientadas a la persona. Proporcionan medidas e información sobre la forma que la gente desarrolla el software de computadoras y sobre todo el punto de vista humano de la efectividad de las herramientas y métodos. Son las medidas que voy a hacer de mi personal que va hará el sistema.

\_\_\_ Métricas orientadas al tamaño. Es para saber en qué tiempo voy a terminar el software y cuantas personas voy a necesitar. Son medidas directas al software y el proceso por el cual se desarrolla, si una organización de software mantiene registros sencillos, se puede crear una tabla de datos orientados al tamaño.

\_\_\_ Métricas orientadas a la función: Son medidas indirectas del software y del proceso por el cual se desarrolla. En lugar de calcularlas las LDC, las métricas orientadas a la función se centran en la funcionalidad o utilidad del programa.

Diga el por qué de su elección y que factores considera importante en el diseño de métricas para la obtención de una mayor calidad de las BD de su proyecto.

\_\_\_\_\_

---

6. ¿Qué factores cree que influyen en la calidad de una base de datos?

Integridad

Consistencia

Rendimiento

Compleción

Concisión

Comprensibilidad

Flexibilidad

Completitud

Legibilidad

Integración

Cohesión

Minimalidad

Implementación

Usabilidad

Complejidad

Validez

Mantenimiento

Tamaño

Otros \_\_\_\_\_

7. ¿Utilizan alguna herramienta automatizada para gestionar el seguimiento, control y registro de las pruebas?:

Sí  No  No sé

Mencionar las herramientas

---

8. ¿Qué herramienta de modelación se utiliza en el proyecto para modelar las BD? ¿Qué SGBD?

---

9. ¿Qué arquitectura utilizan para el desarrollo de las BD?

---



### Anexo #3: Plantilla de Aplicación de Métricas

Nombre de Proyecto:

Siglas:

#### 1- Conjunto #1

Característica	Variable	Fórmula	Descripción	Valores	Resultado
Adecuación al problema	O1		Criterio del usuario		
Validez del Modelo	O2		Criterio de un especialista		
Consistencia	O3	O3= 5- D1	Donde NI: # de Inconsistencias y n # de redacciones del modelo		
		D1= NI/ 4n			
Concisión	O4	O4= $M((Cn^2 - n^1)/Cn - (n-1))$			
Compleitud	O5	O5= 5 - NF	Donde NF: # de fallos en dependencia a la importancia de la consulta		
Cohesión	O6	O6i = $M((ne-np) / (ne -1))$ O6= $\sum O6i / n$	Donde ne: # de atributos de la entidad Np: # de atributos identificadores primarios		
Validez de contenido	O7	O7= $(1 - ni/\sum ne)$	Donde ni: # de entidades no Válidas ne: # de atributos de la Entidad		

El rango de valores a obtener de las variables Oi (i =1.....7) sería: (1<=Oi<=5).

#### S1: Facilidad de Uso desde el punto de vista del usuario

$$S1= (O1 + O3+ O4 + O5) / 4$$

#### S2: Usabilidad desde el punto de Vista del Diseñador

$$S2= (O2+ O3+ O5+ O6 + O7) / 5$$

### S3: Facilidad de Mantenimiento

$$S3 = (O2 + O4 + O6) / 3$$

### S4: Precisión

$$S4 = (O3 + O5) / 2$$

### S5: Rendimiento

$$S5 = (O4 + O5) / 2$$

### Cálculo de la Calidad del Modelo:

$$Q = \sum W_i S_i \text{ (con } i \text{ del 1 al 5)}$$

Donde  $W_i$  son los pesos de los factores de comportamiento  $S_i$  son los valores de dichos factores.

## 2- Conjunto #2

Característica	Fórmula	Descripción	Valores	Resultado
<b>Legibilidad</b>	$\frac{1}{\frac{NFI}{e^{0.5*NE}}}$	Donde NFI: # de fuentes de Ilegibilidad NE: # de estructuras		
<b>Compleitud</b>	$(RE/RT)^2$	Donde RE: # de requisitos de especificación expresados en el esquema RT: # de requisitos totales		
<b>Correctitud Semántica</b>	$1/ErrSem + 1$	Donde ErrSem: # de errores semánticos		
<b>Correctitud Sintáctica</b>	$1/ErrSin + 1$	Donde ErrSin: # de errores sintácticos		
<b>Minimalidad</b>	$1-(FR/A)^2$	Donde FR: # de fuentes de redundancia A: # de atributos del esquema		
<b>Expresividad</b>	$\frac{RA}{2*RT} + \frac{1}{2*e^{\frac{RI}{RT}}}$	Donde RA: # de requisitos acertados RI: # de requisitos presentes en el esquema		

<b>Autoexplicación</b>	$\frac{1}{IA / e^{0.5*RT}}$	Donde IA: Información anexa al esquema		
<b>Consistencia</b>	$1/ NI + 1$	Donde NI: # de inconsistencias del esquema		
<b>Economía</b>	$\left[ \frac{RAC}{NE * RTE} * 20 \right]^2$	Donde RAC: # de requisitos de estructura acertados RTE: # total de requisitos de estructura de la especificación NE: # de estructuras		

### **Cálculo de la Calidad del Modelo:**

$Q = W1 * \text{Legibilidad} + W2 * \text{Compleitud} + W3 * \text{Correctitud Semántica} + W4 * \text{Correctitud Sintáctica} + W5 * \text{Minimalidad} + W6 * \text{Expresividad} + W7 * \text{Autoexplicación} + W8 * \text{Consistencia} + W9 * \text{Economía}$

Donde  $W_i$   $i= 1$  a  $9$  son los pesos asociados a cada criterio de calidad.

## Anexo #4: Plantilla de aplicación de las Métricas QDB.

Nombre de Proyecto:

Siglas:

Característica	Fórmula	Descripción	Valores	Resultado
Legibilidad	$Legibilidad = \frac{1}{\frac{NFI}{e^{0.5*NE}}}$	<p>NFI: # de fuentes de ilegibilidad.</p> <p>NE: número de estructuras.</p>		
Compleitud	$Compleitud = \left[ \frac{RE}{RT} \right]^2$	<p>RE: # de requisitos de la especificación expresados en el esquema.</p> <p>RT: # de requisitos totales de la especificación.</p>		
Correctitud Sintáctica	$Correct\_Sin = \frac{1}{ErrSin + 1}$	<p>ErrSin: # de errores sintácticos</p>		
Correctitud Semántica	$Correct\_Sem = 1 - \frac{RNC}{RT}$	<p>RT: # total de requisitos presentes en la especificación.</p> <p>RNC: # de requisitos en la especificación, representados de manera errónea en el esquema.</p>		
Minimalidad	$Minimalidad = \left[ 1 - \frac{FR}{A} \right]^2$	<p>FR: # de fuentes de redundancia.</p> <p>A: # de atributos del esquema.</p>		
Expresividad	$Expresividad = 1 - \frac{RI}{RE}$ $RE = RA + RI + RNC$	<p>RA: # de requisitos acertados.</p> <p>RI: # de requisitos presentes en el esquema, pero que no se encuentran en la especificación.</p> <p>RNC: # de requisitos de la especificación, representados de manera errónea en el esquema.</p>		

Autoexplicación	$Autoexp = 1 - \frac{NS + NR}{RT}$	NS: Información anexa al esquema. RT: # de requisitos totales. NR: Número de restricciones no modeladas		
Consistencia	$Consistencia = \frac{1}{NI + 1}$	NI: # de inconsistencias del esquema		

### Calidad del Modelo:

La **calidad del modelo** (QDB<sup>2</sup>) se calcularía a partir de los valores de las características anteriores de la siguiente manera:

$$QDB = w_1 * Legibilidad + w_2 * Completitud + w_3 * CorrectSin + w_4 * CorrectSem + w_5 * Minimalidad + w_6 * Expresividad + w_7 * Autoexp + w_8 * Consistencia.$$

$W_i$   $i=1...8$  son los pesos asociados a cada criterio de calidad según su importancia.

Donde  $(0 \leq QDB \leq 24)$  y  $(1 \leq W_i \leq 3)$

Teniendo en cuenta los pesos asignados por los diseñadores a la importancia de cada indicador, se define el máximo valor posible que sería el ideal para ese proyecto según sus expectativas. Posteriormente se lleva el resultado obtenido QDB<sub>p</sub> a una escala de 24, que no es más que el mayor valor definido en los intervalos estándares de evaluación final.

Rango	Deficiente	Adecuado	Eficiente
$0 \leq Q \leq 8$	X		
$9 \leq Q \leq 18$		X	
$19 \leq Q \leq 27$			X

<sup>2</sup> Quality Database

## **Anexo #5: Guía de evaluación técnica de la propuesta mediante el criterio de expertos.**

### **Guía para informar el peso de los criterios.**

Fecha de recepción...00/00/00.....

Fecha de entrega....00/00/00.....

Nombre y Apellidos del evaluador.....

Le otorgará un peso a cada criterio de acuerdo a su opinión y el peso total de cada grupo debe sumar:

Grupo No.1..... 40

Grupo No.2..... 20

Grupo No.3.....20

Grupo No.4.....20

Para que el peso total asignado sea 100. Cada Factor que se mida tiene un valor de 10 puntos.

#### **Grupo No 1: Criterios de mérito científico.**

1. Valor científico de la propuesta.

Peso.....

2. Calidad de la investigación

Peso.....

3. Contribución científica.

Peso.....

4. Responsabilidad científica y profesionalidad de los investigadores.

Peso.....

#### **Grupo No 2: Criterios de implantación.**

5. Necesidad de empleo de la propuesta.

Peso.....

6. Posibilidades de aplicación.

Peso.....

#### **Grupo No 3: Criterios de flexibilidad.**

7. Adaptabilidad a entidades dedicadas a evaluar la calidad de los productos de software.

Peso.....

8. Capacidad del proceso de evaluación para la admisión de cambios que impliquen mejoras.

Peso.....

**Grupo No 4. Criterios de impacto.**

9. Impacto en el área para la cual está destinada la guía.

Peso.....

10. Organización en el proceso de desarrollo.

Peso.....

**Categoría final del proyecto**

\_\_\_ Excelente: Alta novedad científica, con aplicabilidad y resultados relevantes.

\_\_\_ Bueno: Novedad científica, resultados destacados.

\_\_\_ Aceptable: Suficientemente bueno con reservas.

\_\_\_ Cuestionable: No tiene relevancia científica y los resultados son malos.

\_\_\_ Malo: No aplicable.

Valoración final Sugerencias del experto para mejorar la calidad del proyecto.

Elementos críticos que deben mejorarse.

Anexo #6: Tabla de Distribución Chi Cuadrado ( $\chi^2$ ).

v/p	0,001	0,0025	0,005	0,01	0,025	0,05	0,1	0,15	0,2
1	10,8274	9,1404	7,8794	6,6349	5,0239	3,8415	2,7055	2,0722	1,6424
2	13,8150	11,9827	10,5965	9,2104	7,3778	5,9915	4,6052	3,7942	3,2189
3	16,2660	14,3202	12,8381	11,3449	9,3484	7,8147	6,2514	5,3170	4,6416
4	18,4662	16,4238	14,8602	13,2767	11,1433	9,4877	7,7794	6,7449	5,9886
5	20,5147	18,3854	16,7496	15,0863	12,8325	11,0705	9,2363	8,1152	7,2893
6	22,4575	20,2491	18,5475	16,8119	14,4494	12,5916	10,6446	9,4461	8,5581
7	24,3213	22,0402	20,2777	18,4753	16,0128	14,0671	12,0170	10,7479	9,8032
8	26,1239	23,7742	21,9549	20,0902	17,5345	15,5073	13,3616	12,0271	11,0301
9	27,8767	25,4625	23,5893	21,6660	19,0228	16,9190	14,6837	13,2880	12,2421
10	29,5879	27,1119	25,1881	23,2093	20,4832	18,3070	15,9872	14,5339	13,4420
11	31,2635	28,7291	26,7569	24,7250	21,9200	19,6752	17,2750	15,7671	14,6314
12	32,9092	30,3182	28,2997	26,2170	23,3367	21,0261	18,5493	16,9893	15,8120
13	34,5274	31,8830	29,8193	27,6882	24,7356	22,3620	19,8119	18,2020	16,9848
14	36,1239	33,4262	31,3194	29,1412	26,1189	23,6848	21,0641	19,4062	18,1508
15	37,6978	34,9494	32,8015	30,5780	27,4884	24,9958	22,3071	20,6030	19,3107
16	39,2518	36,4555	34,2671	31,9999	28,8453	26,2962	23,5418	21,7931	20,4651
17	40,7911	37,9462	35,7184	33,4087	30,1910	27,5871	24,7690	22,9770	21,6146
18	42,3119	39,4220	37,1564	34,8052	31,5264	28,8693	25,9894	24,1555	22,7595