



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES  
ACATLÁN

Métricas para la calidad del software en el diseño de un  
Sistema de Gestión de Incidencias Orientado a Objetos

TESINA

QUE PARA OBTENER EL TÍTULO DE:  
LICENCIADO EN MATEMÁTICAS APLICADAS Y COMPUTACIÓN

PRESENTA:

ANABEL JIMÉNEZ CANO

ASESOR:

LIC. ALEJANDRO ROBERTO RUBIO PEREZ

Mayo de 2013



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

*Con todo mi amor y cariño para:  
La familia Jiménez en especial para  
Sergio y Alan, como ejemplo de perseverancia.*

# AGRADECIMIENTOS

*A DIOS, por mostrarme que a pesar de las dificultades que tengamos en la vida siempre confiando en su voluntad lograremos cada uno de nuestros sueños con esfuerzo y amor.*

*A mis padres Francisco y Minerva, por ser un ejemplo de fortaleza, por sus consejos y apoyo en cada uno de mis planes, a mis hermanas Guadalupe y Mariela, por ser mis mejores amigas y compañeras en este camino.*

*A mis amigos, y compañeros de la carrera, gracias por sus consejos y reuniones, que hicieron la carrera mucho más sencilla, ah! Y mucho mas sencilla la TESINA, y por qué hoy en día muchos de ustedes forman parte importante de mi vida.*

*Un especial agradecimiento a mi equipo de trabajo QA Maxcom, por su apoyo y demostrarme que en el ámbito profesional podemos fomentar buenas amistades.*

*A mi asesor Rubio, por su paciencia conmigo y mi redacción, así como a todos mis sínodos por sus aportaciones a este trabajo.*

*A la Universidad Nacional Autónoma de México en especial a la FES Acatlán, por darme la oportunidad de fomentar mis estudios en esta casa, y enseñarme que lejos de una clase en una aula, la vida... la vida es mucho mejor cuando eres PUMA!.*

# ÍNDICE GENERAL

<b>I. Resumen</b>	8
<b>II. Introduccion</b>	9
<b>Capítulo 1. Introducción a la calidad del software</b>	10
1.1 ¿Qué es la Calidad?	10
1.2 Calidad de Software	10
1.2.1 Aseguramiento de la calidad del software	13
1.2.2 Gestión de la calidad del software	14
1.2.3 Control de la calidad del software	14
1.2.4 Certificación de la calidad del software	15
<b>Capítulo 2. Modelos, estándares y normas de la calidad de software</b>	16
2.1 Modelos de Calidad de Software a Nivel Proceso	16
2.2 Modelos de Calidad de Software a Nivel Producto	18
2.3 Estándares y normas de calidad de software	19
<b>Capítulo 3. Diseño del SGIIO (Sistema de Gestión de Incidencias Orientado a Objetos)</b>	22
3.1 Requerimientos del Sistema	23
3.2 Especificación de Requerimientos	25

3.3 Proceso de SGIIO (Sistema de Sistema de Gestión de Incidencias Orientado a Objetos).	29
3.4 Diseño del Sistema de Gestión de Incidencias Orientado a Objetos)..	33
<b>Capítulo 4. Implementación de Métricas</b>	42
4.1 Métricas Aplicadas al diseño del Sistema de Gestión de Incidencias Orientada a Objetos	46
<b>Conclusiones</b>	76
<b>Bibliografía</b>	78
<b>Anexo A</b>	80
<b>Anexo B</b>	82

# ÍNDICE DE FIGURAS

<i>Ilustración 1.</i> Factores y criterios de calidad McCall (1997).....	11
<i>Ilustración 2.</i> Componentes de Calidad Software .....	14
<i>Ilustración 3.</i> Diagrama Proceso.....	31
<i>Ilustración 4.</i> Diagrama de Estados SGIOO-Ticket Incidencia .....	33
<i>Ilustración 5.</i> Diagrama de Actividades SGIOO .....	34
<i>Ilustración 6.</i> Diagrama Secuencia Registrar .....	35
<i>Ilustración 7.</i> Diagrama Secuencia Categorizar .....	35
<i>Ilustración 8.</i> Diagrama Secuencia Priorizar .....	36
<i>Ilustración 9.</i> Diagrama de Secuencia Resolver.....	36
<i>Ilustración 10.</i> Diagrama de Secuencia Cierre.....	37
<i>Ilustración 11.</i> Caso de Uso Administrador .....	38
<i>Ilustración 12.</i> Caso de Uso Solicitante .....	39
<i>Ilustración 13.</i> Caso de Uso ServiceDesk.....	40
<i>Ilustración 14.</i> Caso de Uso Soporte Nivel 1 .....	40
<i>Ilustración 15.</i> Caso de Uso Soporte Nivel 2 .....	41
<i>Ilustración 16.</i> Diagrama de clases.....	41
<i>Ilustración 17.</i> Proceso Recopilación de Métricas.....	43

# ÍNDICE DE TABLAS

Tabla 1. Factores de Calidad de Producto.....	12
Tabla 2. Revisión del Producto.....	12
Tabla 3. Transacción del producto.....	13
Tabla 4. Modelos de Calidad de Software a Nivel Proceso.....	18
Tabla 5. Modelos de Calidad de Software a Nivel Producto.....	19
Tabla 6. Estándares y Normas de Calidad de Software.....	21
Tabla 7. Proceso del Sistema de Gestión de Incidencias Orientado a Objetos.....	32
Tabla 8. Objetivos por Roles de la Medición de Software.....	43
Tabla 9. Niveles de dificultad Puntos de Fusión EE.....	51
Tabla 10. Asignación de pesos Puntos de Fusión CE y SE.....	52
Tabla 11. Asignación de pesos Puntos de Fusión ALI y AIE.....	52
Tabla 12. Asignación de pesos Puntos de Fusión.....	53
Tabla 13. Estimadores Puntos de Fusión.....	55
Tabla 14. Factores de Ponderación de Actor.....	64
Tabla 15. Factores de Ponderación de Actor basada en transacciones.....	64
Tabla 16. Análisis de clases en Factores de Ponderación de Actor.....	64
Tabla 17. Factores de Ponderación de Administrador.....	64
Tabla 18. Factores de Ponderación de Administrador, basada en transacciones.....	64
Tabla 19. Análisis de clases en Factores de Ponderación de Administrador.....	65
Tabla 20. Factores de Ponderación de ServiceDesk.....	65
Tabla 21. Factores de Ponderación de ServidDesk basada en transacciones.....	65
Tabla 22. Análisis de clases en Factores de Ponderación de ServiceDesk.....	65
Tabla 23. Factores de Ponderación de Solicitante.....	65
Tabla 24. Factores de Ponderación de Solicitante, basada en transacciones.....	66
Tabla 25. Análisis de clases en Factores de Ponderación de Solicitante.....	66
Tabla 26. Factores de Ponderación de Soporte Nivel 1.....	66
Tabla 27. Factores de Ponderación de Soporte Nivel 1 basada en transacciones.....	66
Tabla 28. Análisis de clases en Factores de Ponderación de Soporte Nivel 1.....	66
Tabla 29. Factores de Ponderación de: Soporte Nivel 2.....	66
Tabla 30. Factores de Ponderación de Soporte Nivel 2 basada en transacciones.....	67
Tabla 31. Análisis de clases en Factores de Ponderación de Soporte Nivel 2.....	67
Tabla 32. Factores de ambiente de Propuesta de Karner.....	69
Tabla 33. Medidas del diagrama de estados.....	70
Tabla 34. Medidas de Cruz Lemus.....	71
Tabla 35. Medidas de Cruz Lemus aplicadas a SGIOO.....	72



# RESUMEN

En este trabajo se describe el uso de métricas de calidad de software, iniciando con términos generales en cuanto a calidad se refiere, hasta aspectos más específicos hablando de calidad de software y los términos que le atribuyen a éste, y los cuales son necesarios para entender hasta que nivel de calidad pretendemos llegar, presentando un ejemplo un diseño de un sistema.

Adicionalmente, se muestra el análisis de un diseño de sistema que tiene como objeto reducir los tiempos de respuesta y costo, en un servicio de atención, como lo es el servicedesk (mesa de ayuda), el cual es orientado a objetos, diseñando diagramas y especificando requerimientos.

Finalmente el trabajo termina mostrando cómo es que a través de los diagramas de diseño y definición de requerimientos del sistema es posible utilizar métricas de calidad, que nos ayuden a predecir cuales podrían ser nuestros posibles costos, tiempos y personal invertidos para implementar nuestro sistema y que a partir de estos resultados se tome una decisión.

# INTRODUCCIÓN

La hipótesis de la cual parte este trabajo de investigación es mostrar que basándose en el uso de métricas de calidad de software, las primeras fases del ciclo de vida de un sistema, - especificación y diseño -, puede aumentar la calidad del producto final reduciendo el tiempo y costos invertidos en los mismos.

Si hablamos de desarrollos de Software, muchas veces creemos que será un esfuerzo en el que además de verse involucrada una cierta cantidad de personal, también se verá invertida una cantidad de tiempo y dinero, y en algunos casos, aquellos proyectos que generaron más gastos para las empresas son proyectos que terminan fracasando, o su resultado es completamente distinto al que se tenía planeado desde un principio, y en la mayoría de las ocasiones, no se encuentra un motivo del porque estos resultados.

Es así como el objetivo de este trabajo es la enseñanza de buenas prácticas que ayuden a las empresas a mejorar la calidad de sus productos de software, con la búsqueda y el uso de métodos, modelos y métricas de calidad que si son conocidas; en la mayoría de las empresas de desarrollos de TI no son empleadas, debido a la falta de tiempos, costos y conocimientos.

En el primer capítulo se dará una pequeña introducción sobre lo que es la Calidad de Software así como sus variantes.

En el segundo capítulo, describiremos los Modelos y Estándares necesarios para la Calidad de Software con lo cual se pretende, tener una idea más general del porque es necesario el uso de estos, y con ello poder basarnos en el diseño de nuestro sistema.

En el tercer capítulo se describirá el diseño de nuestro sistema en base al uso de un modelo de calidad del mismo, también se describirán su análisis y requerimientos.

En el cuarto capítulo se explicará el uso de las métricas de calidad para el diseño de nuestro sistema, sin embargo el ejemplo pretende ser general, es decir que este uso pueda ser para: cualquier sistema que en algún momento pretendamos medir.

Para finalizar, haremos algunos comentarios, expondremos las conclusiones a las cuales hemos llegado al usar estas métricas en el ejemplo.

# Capítulo 1

## Introducción a la calidad del software

*"Hagas lo que hagas no has terminado hasta que hayas verificado que lo que has hecho es lo que querías hacer" Ivar Jacobson Rational Software*

### 1.1 ¿Qué es la Calidad?

En este primer capítulo hablaremos del termino de Calidad, sin lugar a duda definir este término no es una tarea fácil ya que últimamente ha sido un término en boga, esto a consecuencia de las exigencias de un mundo globalizado, podemos encontrar miles de definiciones en libros, internet etc. y confundir el fin real por lo que hemos buscado: una definición concisa y general.

Tomemos en cuenta algunas definiciones de las fuentes más precisas que podemos encontrar:

Definición de RAE: *"Propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor"*<sup>1</sup>

La definición de ISO (international Standards Organization) en la norma 9000: *"Es la totalidad de las características de una entidad que le confieren la aptitud para satisfacer las necesidades establecidas e implícitas"*

"Calidad", es fácilmente entendida como *"aquello que satisface las necesidades de los consumidores, en relación de lo que éstos busca"*, pero es cierto que no podemos encontrar una definición general y que debemos de verla como una cualidad y propiedad inherente de las cosas, que permite que estas sean comparadas; ya que se trata de una apreciación subjetiva.

### 1.2 Calidad de Software

En un mundo controlado por la industria del software es más crítica la necesidad de contar con productos de calidad, la preocupación de buscar y optimizar productos ha aumentado considerablemente en la competencia comercial.

---

<sup>1</sup><http://lema.rae.es/drae/?val=calidad>

“La calidad del software no comprende el término de producto final”, ya que éste implica un proceso y ciclo de vida a un mayor en donde la calidad debería de estar marcada en cada proceso del ciclo de vida del mismo.

**Definición IEEE, Std 610-1900.** La calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario.

Hablar de calidad del software es hablar de los factores que la determinan, según el *Modelo de McCall* se clasifica en tres grandes grupos los cuales son:

- Operaciones del producto.
- Revisión del producto.
- Transacción del producto.

La descripción de la calidad de McCall comprende al producto final así como un aserie de factores y la relación jerárquica de estos, desde el punto de vista del usuario final, básicamente son factores externos, que resultan poco medibles, sin embargo también incluye criterios entre cada factores que pueden identificarse como medibles.

Ahora veamos un triángulo que describe de manera general, y la relación jerárquica que existe entre los factores de la calidad de McCall, así como sus criterios.

**Diagrama de Factores y criterios de Calidad McCall<sup>2</sup> (1997)**



**Ilustración 1.** Factores y criterios de calidad McCall (1997)

<sup>2</sup><http://www.cs.uns.edu.ar/~prf/teaching/SQ07/clase6.pdf>

## Capítulo 1. Introducción a la calidad del software

### Operaciones del Producto: Características Operativas<sup>3</sup>

Se enfocan en determinar, como funciona mi producto, es decir si realmente hace lo que yo quiero que haga y cumple las necesidades para las cuales fue desarrollado.

Criterio	Descripción
Corrección. (¿Hace lo que se pide?).	El grado en que una aplicación satisface sus especificaciones y consigue los objetivos encomendados por el cliente.
Fiabilidad. (¿Lo hace de forma fiable todo el tiempo?).	El grado que se puede esperar de una aplicación lleve a cabo las operaciones especificadas y con la precisión requerida
Eficiencia. (¿Qué recursos hardware y software necesito?).	La cantidad de recursos hardware y software que necesita una aplicación para realizar las operaciones con los tiempos de respuesta adecuados.
Integridad. (¿Puedo controlar su uso?).	El grado con que puede controlarse el acceso al software o a los datos a personal no autorizado.
Facilidad de uso. (Usabilidad) (¿Es fácil y cómodo de manejar?).	El esfuerzo requerido para aprender el manejo de una aplicación, trabajar con ella, introducir datos y conseguir resultados.

**Tabla 1.** Factores de Calidad de Producto

### Revisión del Producto: Capacidad para Soportar cambios

Se basan en aquellos factores que me ayuden a determinar cuales serian los costos y facilidades para hacer mi producto más robusto o en su defecto corregirlo.

Criterio	Descripción
Facilidad de mantenimiento. (¿Puedo localizar los fallos?)	El esfuerzo requerido para localizar y reparar errores
Flexibilidad. (¿Puedo añadir nuevas opciones?).	El esfuerzo requerido para modificar una aplicación en funcionamiento.
Facilidad de prueba. (¿Puedo probar todas las opciones?).	El esfuerzo requerido para probar una aplicación de forma que cumpla con lo especificado en los requisitos.

**Tabla 2.** Revisión del Producto

<sup>3</sup>[http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/gonzalez\\_d\\_h/capitulo3.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/gonzalez_d_h/capitulo3.pdf)

*Transacción del producto: Adaptabilidad a nuevos entornos*

Se basa en factores externos del producto y que me ayudan a determinar cuáles son algunas de las ventajas externas que desempeño tiene mi producto y hasta que punto puedo explotar su funcionalidad pero sobretodo

<b>Criterio</b>	<b>Descripción</b>
Portabilidad. <i>(¿Podré usarlo en otra máquina?).</i>	El esfuerzo requerido para transferir la aplicación a otro hardware o sistema operativo.
Reusabilidad <i>(¿Podré utilizar alguna parte del software en otra aplicación?).</i>	Grado en que partes de una aplicación pueden utilizarse en otras aplicaciones.
Interoperabilidad. <i>(¿Podrá comunicarse con otras aplicaciones o sistemas in formaticos?).</i>	El esfuerzo necesario para comunicar la aplicación con otras aplicaciones o sistemas informáticos.

**Tabla 3.** Transacción del producto

En conclusión podemos decir que la Calidad del Software es el desarrollo de software basado en estándares, procesos y modelos con la funcionalidad y rendimiento que satisfacen los requerimientos del cliente, en el que el producto de software incorpora un conjunto deseable, y definido por la industria, de características de manera que aumenta su rendimiento a lo largo de todo su tiempo de uso, simplifica su mantenimiento y facilita asignarle nuevos alcances.

### **1.2.1 Aseguramiento de la calidad del software**

Se enfoca en identificar todos y cada uno de los errores o defectos que puedan afectar el software, si dichos errores se pueden identificar en el principio del proceso las características del diseño se pueden especificar de modo que se eliminarán o controlarán los peligros potenciales, al corregir errores al principio de cada proceso se reducen costos, esfuerzo y tiempo.

La confianza de la empresa y del cliente se basa principalmente en la satisfacción, otro punto que trata es el costo que puede implementar un cambio a ciertas características del producto y con esto determinar la calidad de nuestro producto.

Sridharan<sup>4</sup> (2000) indica que mientras el software que se desarrolla cumpla y reúna los requerimientos además de que su desempeño sea el esperado, es preciso que se supervisen las actividades del desarrollo en distintas oportunidades durante cada fase del ciclo de vida del software. Este es el papel de Aseguramiento de la Calidad.

## 1.2.2 Gestión de la calidad del software

Es el conjunto de actividades de la dirección que define la calidad, los objetivos y las responsabilidades, se basa en la determinación y aplicación de las políticas de calidad de la empresa.

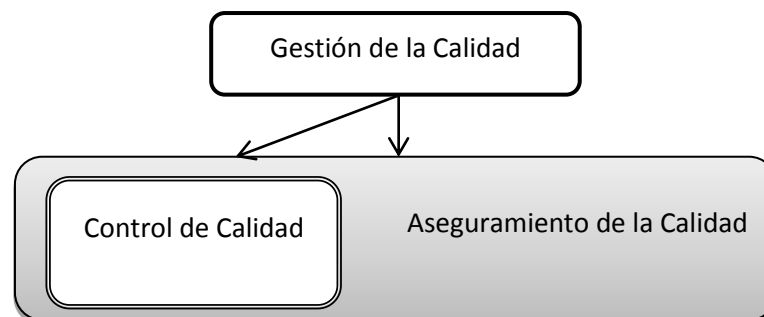
El propósito de la gestión de la calidad del software es entender las expectativas del cliente en términos de calidad, poner en práctica un plan proactivo para satisfacer estas expectativas.

## 1.2.3 Control de la calidad del software

En la ingeniería del Software el control de la calidad contempla la funcionalidad de asegurar que los procedimientos y normas son seguidos por el equipo de desarrollo de software.

Implica vigilar el proceso de desarrollo del software para asegurar que se siguen los procedimientos y los estándares de garantía de calidad, se comprueba que las entregas cumplan los estándares definidos, al final de cada proceso y revisar el producto este debe de cumplir producto con los requerimientos del cliente.

El siguiente diagrama muestra la relación entre estos tres conceptos.



*Ilustración 2.* Componentes de Calidad Software

<sup>4</sup> <http://www.qualitrain.com.mx/Aseguramiento-de-la-Calidad-de-Software.html>

## **1.2.4 Certificación de la calidad del software**

Cuando los productos, sistemas, máquinas o dispositivos funcionan bien y con seguridad, a menudo es porque cumplen con las normas.

Las normas que benefician a todo el mundo tienen la certificación ISO, con esto se incrementa la competitividad en el mercado como una compañía confiable y garantizada.

Las ventajas al crear y tener software cuya calidad y procesos están certificados por normas de reconocimiento internacional son indudables, tales como:

- Mejora de la productividad del proceso de producción del software por la reducción del “reproceso” que agrega costo y no agrega valor al producto.
- Reducción del “tiempo de comercializar” de un producto de software.
- Ampliación del target de clientes que elegirán a la organización por poseer una certificación que el cliente considera necesaria.

Los sistemas de tecnología desempeñan un papel importante en las funciones más generales de una empresa; si hablamos de una empresa dedicada al desarrollo de las tecnologías, la responsabilidad crece al ofrecer sus servicios a terceras personas, y una de las mejores maneras de demostrar, comprobar o avalar la calidad de nuestros productos es en base a las certificaciones.



# Capítulo 2

## Modelos, estándares y normas de la calidad de software.

*"Sin análisis de requisitos o sin diseño, programar es el arte de crear errores en un documento de texto vacío". Louis Srygley*

Modelo de calidad de software; es un conjunto de buenas prácticas, en el ciclo de vida del software enfocado en el proceso de gestión y desarrollo de los proyectos, ya que con la utilización de modelos, procedimientos y estándares se puede alcanzar y mejorar los factores de calidad.

Un modelo de Calidad nos dice que debemos de hacer, no como hacerlo, esto dependerá de las metodologías que usemos así como los objetivos de negocio que tengamos.

Un estándar es un modelo a seguir de cómo hacer algo, si hablamos de estándares de calidad, entendemos que es una serie de normas y protocolos internacionales que deben cumplir los productos para su distribución y consumo.

Implementar un estándar o norma dependerá de la misión y visión de la empresa, es decir si está es dedicada específicamente al desarrollo de software o tiene otras funciones aun más generales y específicas.<sup>5</sup>

### 2.1 Modelos de calidad de software a nivel proceso

En estos modelos encontramos dos tipos de evaluaciones:

- Por niveles de Madurez: Se obtiene una "puntuación", cuya orientación es el alcance es la organización.
- Por niveles de Capacidad (Manera Continua): Se obtiene una "puntuación" para un proceso concreto.

---

<sup>5</sup>[http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/garcia\\_r\\_ci/capitulo2.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/garcia_r_ci/capitulo2.pdf)

## 2.1 Modelos de calidad de software a nivel proceso

Entre algunos de los modelos a nivel proceso podemos mencionar a los más conocidos tales como:<sup>6</sup>

Modelo	Descripción
<b>CMMI</b>	<ul style="list-style-type: none"> <li>• Clasifica a las empresas en diferentes niveles de madurez, alcanzar cada nivel dependerá del nivel anterior.</li> <li>• Los niveles en los que se divide CMMI son:               <ul style="list-style-type: none"> <li>- Inicial (nivel 1).</li> <li>- Gestión del proyecto (nivel 2).</li> <li>- Estandarización del Proceso (nivel 3).</li> <li>- Control estadístico (nivel 4).</li> <li>- Optimización Permanente (nivel 5: estado final).</li> <li>- CMMI reduce el costo de la No Calidad aumento la satisfacción del cliente, así como la productividad y rentabilidad.</li> </ul> </li> </ul>
<b>guiTickIT</b>	<ul style="list-style-type: none"> <li>• Abarca la evaluación y certificación del esquema de gestión de calidad de software.</li> <li>• Estimula a los desarrolladores de software a pensar en lo realmente significa calidad en el contexto del desarrollo de software y como puede lograrse.</li> </ul>
<b>Bootstrap</b>	<ul style="list-style-type: none"> <li>• Reduce costos y mejora la calidad previniendo problemas, desarrollando un método para la evaluación de procesos de desarrollo software, se basa en el modelo CMM.</li> <li>• Establece un diagnóstico del proceso de desarrollo de software en la organización, métodos y capacidad, creando un plan de acción que defina los pasos, detalles y marcos para aumentar la entrega de productos y servicios de calidad.</li> </ul>
<b>Personal Software Process (PSP)</b>	<ul style="list-style-type: none"> <li>• Está diseñado para ser usado por un Ingeniero de Software individual, el cual debe ser entrenado por una persona calificada en PSP.</li> <li>• Su objetivo es guiar el planteamiento y desarrollo de los módulos de software, optimizar la interacción entre equipo y posteriormente trabajar con TSP.</li> </ul>
<b>Team Software Process (TSP)</b>	<ul style="list-style-type: none"> <li>• Es una metodología para dirigir el trabajo de mejora y desarrollo de software además de establecer un entorno donde el trabajo efectivo de equipo sea normal y natural.</li> </ul>
<b>Practical Software Measurement (PSM)</b>	<ul style="list-style-type: none"> <li>• Sirvió de documento base para el desarrollo de la norma ISO / IEC 15939, proporciona información detallada de "cómo hacer" incluidas las medidas de orientación de la muestra, las lecciones aprendidas, estudios de casos, y orientaciones para la aplicación</li> <li>• PSM ofrece un conjunto de medidas de la muestra, utilizando el modelo de información de la medida terminología.</li> </ul>
<b>QualipSo Open SourceMaturity Model (OMM)</b>	<ul style="list-style-type: none"> <li>• Es una nueva opción para desarrollar software con modelos de madurez de calidad, una opción los amantes del código abierto FLOSS (Free/Libre Open Source Software).</li> </ul>

<sup>6</sup><http://laboratorios.fi.uba.ar/lis/scalone-tesis-maestria-ingenieria-en-calidad.pdf>

<b>A Competency-Based Maturity Model</b>	<ul style="list-style-type: none"> <li>• Se basa en una perspectiva multi-dimensional, proporcionando una visión mas completa de la madurez de una organización y la probable complejidad de los cambios necesarios para la adopción de ITIL.</li> </ul>
--	--

**Tabla 4.** Modelos de Calidad de Software a Nivel Proceso

## 2.2 Modelos de Calidad de Software a nivel producto

Los modelos de Calidad a Nivel producto especifican la calidad mediante la definición y descomposición de un conjunto de atributos o características, los cuales pueden ser medidos mediante métricas

Entre algunos de los modelos a nivel producto podemos mencionar a los más conocidos tales como:

Modelo	Descripción
<b>Modelo Gilb</b>	<ul style="list-style-type: none"> <li>• Plantea la creación de requisitos de calidad escritas conjuntamente por el usuario y analista, determinando una lista de características y subcategorías que definen la calidad, las cuales se pueden medir</li> </ul>
<b>Modelo GQM (Goal-Question-Metric)</b>	<ul style="list-style-type: none"> <li>• Plantea el enfoque de la medición para evaluar la calidad del software basado en la identificación de objetivos a lograr, ayudando a la toma de decisiones con la interpretación de las métricas.</li> </ul>
<b>Modelo McCall</b>	<ul style="list-style-type: none"> <li>• Contiene la visión del usuario, de la dirección y del desarrollador .Tiene 3 ejes o puntos de vista, 11 factores, 23 criterios y 41 métricas.</li> </ul>
<b>Modelo BOEHM</b>	<ul style="list-style-type: none"> <li>• Contiene la visión del usuario de la dirección y del desarrollador, tiene 7 factores y 12 criterios.</li> </ul>
<b>Modelo de Dromey</b>	<ul style="list-style-type: none"> <li>• Plantea la calidad en subcaracterísticas que pueden ser medidas y evaluadas, aumentando el entendimiento de la relación entre atributos</li> </ul>
<b>Modelo C-QM</b>	<ul style="list-style-type: none"> <li>• Provee un modelo de calidad comprensivo que se aplica para evaluar aspectos de la calidad del software, su estructura es de 3 etapas: factor, criterio y métricas.</li> </ul>
<b>ISO 9126</b>	<ul style="list-style-type: none"> <li>• Pensado para los desarrolladores adquirentes, personal de aseguramiento de calidad y evaluadores independientes.</li> <li>• Puede servir para validar la completitud de una definición de requisitos, identificar requisitos de calidad de software, objetivos de diseño y prueba, etc.</li> <li>• Propone modelo de calidad, métricas externas, internas y en</li> </ul>

	USO.
<b>ISO 14598</b>	<ul style="list-style-type: none"> <li>• Contiene evaluación del producto de software (soporte lógico )</li> </ul>
<b>Metodología SQAE (Software Quality Assessment Exercise)</b>	<ul style="list-style-type: none"> <li>• Plantea factores tangibles y medibles cuantificando los riesgos asociados al ciclo de vida del software.</li> </ul>
<b>WebQEM (Web Quality Evaluation Method)</b>	<ul style="list-style-type: none"> <li>• Comprende el grado de cumplimiento de un conjunto de características y subcaracterísticas con respecto a los requerimientos de calidad establecidos.</li> </ul>

**Tabla 5.** Modelos de Calidad de Software a Nivel Producto

## 2.3 Estándares y normas de calidad de software

El término ISO son las siglas de la Organización Internacional para la Estandarización con sede en Génova Suiza.

A continuación describiremos uno de los estándares y normas generales más utilizados, en las empresas de desarrollo de software, ya que aunque una empresa implemente un modelo de software este no servirá de nada si no cuenta con el respaldo de un estándar que ayudara a las mismas a obtener una certificación de la Calidad de Software que servirá como carta de presentación frente a la competencia.

<b>Norma /Estándar</b>	<b>Descripción</b>
<b>ISO 9003</b>	<ul style="list-style-type: none"> <li>• Es una norma de buenas prácticas para definir con más detalle los conceptos de software sobre los procesos de la organización.</li> </ul>
<b>ISO 9000:2000</b>	<ul style="list-style-type: none"> <li>• Gestión y Aseguramiento de Calidad (conceptos y directrices generales) <ul style="list-style-type: none"> <li>○ Sistemas de Gestión de la Calidad Sistema de Gestión: Sistema para establecer la política y los objetivos, y para la consecución de dichos objetivos.</li> <li>○ Gestión de la Calidad. Conjunto completo de procesos utilizados en un sistema de gestión de la calidad.</li> </ul> </li> </ul>
<b>ISO 9001:2000</b>	<ul style="list-style-type: none"> <li>• Obtención de resultados, incluyendo la satisfacción del cliente y otros stakeholders.</li> <li>• Enfocada a: orientación al cliente y modelo de procesos.</li> </ul>

Capítulo 2. Modelos, estándares y normas de calidad de software

<b>ISO 12207</b>	<ul style="list-style-type: none"> <li>• Es el estándar para los procesos de ciclo de vida del software de la organización. Es la base para ISO 15504-SPICE.</li> <li>• Information Technology/Software Life Cycle Processes</li> </ul>
<b>ISO/IEC 15504</b>	<ul style="list-style-type: none"> <li>• Conjunto de 7 normas para establecer y mejorar la capacidad y madurez de los procesos de las organizaciones, proporcionando los principios requeridos para realizar una evaluación de la calidad de los procesos.</li> </ul>
<b>ISO /IEC 15939</b>	<ul style="list-style-type: none"> <li>• Cubre las actividades, modelo de información, terminología de medición, información requerida, aplicación de los resultados de medición, análisis y determinar si los resultados del análisis son válidos.</li> </ul>
<b>ISO/IEC 15288:2008</b>	<ul style="list-style-type: none"> <li>• Establece un marco común para describir el ciclo de vida de los sistemas creados por los seres humanos</li> <li>• También proporciona los procesos que apoyan la definición, el control y la mejora del ciclo de vida de los procesos utilizados en una organización o un proyecto.</li> </ul>
<b>ISO/IEC 9126</b>	<ul style="list-style-type: none"> <li>• Software engineering product quality.</li> <li>• Se divide en 4 partes: define las características de calidad del producto de software (parte 1), las métricas internas y externas (partes 2 y 3), y la calidad en uso que explica como la calidad del producto está sujeta a las condiciones particulares de uso (parte 4)</li> </ul>
<b>ISO/IEC 14598</b>	<ul style="list-style-type: none"> <li>• Contiene los requisitos generales para la especificación y evaluación de la calidad del software y aclara los conceptos generales.</li> <li>• Proporciona una estructura para la evaluación de la calidad de todos los tipos de productos de software y fija los requisitos para los métodos de medición y evaluación del producto de software</li> </ul>
<b>IEEE 1061:1998</b>	<ul style="list-style-type: none"> <li>• Una metodología para el establecimiento de requisitos de calidad y la identificación, implementación, análisis y validación del proceso y métricas de software de calidad del producto se define. La metodología abarca el software-todo el ciclo de vida.</li> </ul>
<b>ISO/IEC 29119</b>	<ul style="list-style-type: none"> <li>• “Software Testing”, es una nueva norma que tiene como objetivo cubrir el ciclo de vida completo de las pruebas de software; análisis, diseño, implementación, y mantenimiento.</li> </ul>
<b>BSI (British Standards Institution)</b>	<ul style="list-style-type: none"> <li>• BS 7925-1, Software Testing: Part 1-Vocabulary</li> <li>• BS 7925-2, Software Testing: Part 2-Software Component Testing.</li> </ul>
<b>IEEE: IEEE Std. 829</b>	<ul style="list-style-type: none"> <li>• Software Test Documentation. Hace referencia a la documentación necesaria para las pruebas de Software, la cual es: <ul style="list-style-type: none"> <li>• Plan de Pruebas</li> <li>• Diseño y Especificación de Pruebas</li> </ul> </li> </ul>
<b>IEEE Std 1008, Software Unit Testing,</b>	<ul style="list-style-type: none"> <li>• Hace referencia a las pruebas de unidad.</li> <li>• Especifica un modo de como realizar las pruebas de unidad del software y puede ser utilizado como base de buenas practicas.</li> </ul>
<b>Software Verification and Validation</b>	<ul style="list-style-type: none"> <li>• Se lleva acabo con una actividad que debe de cubrir de conformidad a los requerimientos y necesidades de los usuarios</li> </ul>

### 2.3 Estándares y normas de calidad de software

<b>IEEE Std 1028-1997 Software Reviews</b>	<ul style="list-style-type: none"><li>• Hace referencia a las revisiones del software, no especifica de los resultados de las revisiones.</li></ul>
<b>ISO/IEC 15289, System and Software</b>	<ul style="list-style-type: none"><li>• LifeCycleProcessInformationProducts</li><li>• Se basa en el ciclo de vida de los procedimientos de información de los productos (documentación), identifica información de los ítems de la documentación de 12207.</li></ul>
<b>ISO/IEC TR 19759,</b>	<ul style="list-style-type: none"><li>• Guide to the Software Engineering Body of Knowledge</li><li>• Caracteriza los conocimientos del cuerpo de conocimiento de IS, además aporta un acceso por temas al cuerpo del conocimiento, promueve consistente de la IS con respecto a otras disciplinas y aporta una base para el desarrollo de un círculo materia para poder certificar y conceder licencia a profesionales.</li></ul>

**Tabla 6.** Estándares y Normas de Calidad de Software

# Capítulo 3

## Diseño del SGIIO (Sistema de Gestión de Incidencias Orientado a Objetos)

*"El comienzo de la sabiduría de un programador está en reconocer la diferencia entre obtener un programa que funcione y obtener uno que funcione correctamente"*  
M.A. Jackson

Como ejemplo del uso de buenas prácticas, el diseño de nuestro sistema de gestión de incidencias orientado objetos, se basa en conceptos generales y puntuales de ITIL.

ITIL (Information Technology Infrastructure Library), fue creado al reconocer que las empresas dependen cada vez más de la informática para alcanzar sus objetivos, basado en un proceso de código de buenas prácticas ayuda a la Gestión de Servicios Informáticos reduciendo costos y tiempos.<sup>7</sup>

Contiene diferentes módulos de Gestión de Servicios de un área de TI sin embargo, es la Gestión de Incidencias, lo que nos interesa y dónde basamos el diseño y análisis del sistema de gestión de incidencias orientado a objetos.

Entre sus metas principales se encuentran:

- Un enfoque sistemático del servicio TI centrado en los procesos y procedimientos.
- El establecimiento de estrategias para la gestión operativa de la infraestructura TI.

Con la buena práctica de un diseño esperamos obtener:

- Descripción de los elementos del sistema a construir.
- Evaluación de la calidad del desarrollo en conjunto de revisiones técnicas.
- Implementar todos los requisitos explícitos contenidos en el modelo de análisis.
- Implementar todos los requisitos implícitos que desea el cliente.
- Ser una guía que puedan leer y entender los que construyan código y los que prueban y mantienen el código.
- Debe proporcionar una completa idea de lo que es el software, enfocando los dominios de datos funcionales.

---

<sup>7</sup>[http://itilv3.osiatis.es/disenio\\_servicios\\_TI.php](http://itilv3.osiatis.es/disenio_servicios_TI.php)

## 3.1 Requerimientos del Sistema

En algunas empresas los costos generados por mantenimiento de problemas o errores, es abrumador, por lo que la búsqueda de mejores prácticas y métodos que nos ayuden a reducir estos costos son más que bienvenidos, es así como el objetivo del sistema que vamos a analizar en el presente trabajo es:

Objetivo. Administrar el registro de una incidencia basado en el uso de un proceso de buenas prácticas, y así dar seguimiento a su proceso de alta, solución y cierre, ayudando con esto a las áreas involucradas, a encontrar una mejor solución y con ello reducir tiempos y costos.

Para cumplir con el objetivo ITIL3 propone:

Gestión de Incidencias. Tiene como objetivo resolver cualquier incidente que cause una interrupción en el servicio de la manera más rápida y eficaz posible.

Objetivos generales:

- Registrar y clasificar las incidencias encontradas por los usuarios.
- Asignar al personal adecuado para la solución de la incidencia.
- Detectar cualquier alteración en los servicios TI.

No se preocupa de encontrar y analizar las causas subyacentes a un determinado incidente sino exclusivamente a restaurar el servicio, facilita el seguimiento de la incidencia al crearla y hasta su cierre.

Los principales beneficios de usar una correcta Gestión de incidencias son:

- Mejorar la productividad de los usuarios.
- Cumplimiento de los niveles acordados en el SLA.
- Mayor control de los procesos y monitorización del servicio.
- Optimización de los recursos disponibles.
- Un CMDB (conjunto mínimo básico de datos) más preciso pues se registran los incidentes en relación con los elementos de configuración.
- Y principalmente, mejorar la satisfacción general de clientes y usuarios.

Y evitarnos los siguientes problemas:

- Reducción de los niveles de servicio.
- Evitar demasiada gente o gente del nivel inadecuado trabajando concurrentemente en la solución de un incidente.



- Pérdida de información sobre las causas y efectos de los incidentes para futuras reestructuraciones y evoluciones.
- Evitar clientes y usuarios insatisfechos por la mala y/o lenta gestión de incidentes.

Sin embargo, algunas de las principales dificultades para poder conseguir una buena Gestión de Incidentes son:

- No se siguen los procedimientos previstos y se resuelven las incidencias sin registrarlas o se escalan innecesariamente y/o omitiendo los protocolos preestablecidos.
- No existe un margen operativo que permita gestionar los picos de incidencias por lo que no se registran adecuadamente e impiden la correcta operación de los protocolos de clasificación y escalado.
- No están bien definidos los niveles de calidad de servicio ni los productos soportados.

Ahora teniendo en mente los objetivos generales de nuestro sistema y los objetivos específicos de ITL, describiremos las funcionalidades del sistema de gestión de incidencias orientada a objetos, puntualizando cada uno de los factores involucrados en su efectividad.

#### **Descripción de Requerimientos del Usuario**

A continuación mostramos aquellos requerimientos que el usuario considera, deben de ser realizados por el sistema de gestión de incidencias orientado a objetos, de manera general, es decir sin excepción alguna el sistema debe de proveer a los usuarios de las siguientes funcionalidades:

- El sistema debe proveer un medio para registrar y administrar el proceso de cierre de las incidencias.
- Roles de los usuarios permitidos; administradores, solicitantes, soporte y servicedesk.
- Retroalimentación constante en cuanto al proceso de solución y cierre de la incidencia por medio de correos electrónicos, y de la aplicación.

#### **Descripción de Requerimientos Funcionales**

Así mismo, y debido a las funcionalidades generales del sistema descrito con anterioridad debe de proveer funciones específicas, que a continuación describimos:

- i. Por medio de un Front End se registrará una incidencia, y se le dará seguimiento.

- ii. Estará direccionada hacia dos BBDD, una que servirá para el registro de las incidencias y otra que proveerá las soluciones de éstas, la cual estará basada en experiencias antiguas de incidencias.
- iii. Se identificará cada incidencia por medio de un ID, el cual se generará después de dar de alta una Incidencia y será enviado por mail a los roles con privilegios.
- iv. El ID servirá para dar seguimientos y cierre a la incidencia.
- v. La BD con información en base a experiencias será nombrada BD del Conocimiento y será quién proveerá la solución correspondiente para cada Nivel de Soporte.
- vi. Las Incidencias serán resueltas por medio de niveles, los cuales dependerán de la gravedad de la incidencia y de la solución encontrada, es decir escalonadas.

Del mismo modo como parte de la funcionalidad del sistema describiremos los niveles de solución:

- a. Nivel de Soporte 1. En este nivel se encontraran las incidencias que tienen un nivel de criticidad Bajo y/o en su defecto se encuentre una Solución en la BD del Conocimiento.
- b. Nivel de Soporte 2. En caso de No encontrar una solución a la incidencia de Nivel 1, se dará seguimiento en un Nivel 2 y se registrara la nueva solución.
- c. Se generara un Reporte de Cálculo de Solución de Incidencias Mensual.

## 3.2 Especificación de Requerimientos

### Requerimientos Funcionales

En Ingeniería de Software los requerimientos son necesidades documentadas sobre el contenido, forma o funcionalidad de un producto o servicio; son declaraciones que definen un atributo, característica y/o cualidades que necesita cumplir un sistema, para que tenga valor y utilidad para el usuario.

Al hablar de requerimientos funcionales no referimos a lo que debe hacer nuestro sistema.

La importancia de los requerimientos es tal, que de ellos depende gran parte del éxito de un desarrollo, se divide en 5 diferentes etapas que comienza con la identificación de requerimientos, el análisis de requerimientos y negociación, especificación de requerimientos, modelados del sistema y la gestión de requerimientos.

La identificación y análisis de requerimientos, comprende a los objetivos generales de ITIL así como sus beneficios:

*Objetivos generales:*

- Registrar y clasificar las incidencias encontradas por los usuarios.
- Generar un Ticket de Incidencia, que lleve el control de las incidencias registradas, dicho Ticket será un número consecutivo.
- Asignar al personal adecuado para la solución de la incidencia.
- Detectar cualquier alteración en los servicios TI.
- Contar con una BD que proporcione aquellas soluciones previas a incidencias, y con esto se reduzcan tiempos y costos, dicha BD será nombrada, BD del Conocimiento.
- Dar seguimiento a las incidencias por medio de un correo y por una página.

Ahora definiremos cada uno de los roles de los usuarios finales de el sistema, con el objetivo de entender las responsabilidades y actividades que desempeñarán.

**Responsabilidades de los Roles involucrados en el sistema.**

**Rol Solicitante.** Aquel usuario que se encuentre dado de alta en la BD, con privilegios únicamente de levantar una incidencia, darle seguimiento y cerrarla, se clasificara de acuerdo a su área de trabajo.

**Rol Administrador.** Aquel usuario dado de alta en la BD, con privilegios de gestionar, a los solicitantes, servicedesk y soporte, registrados, se encarga principalmente de crear los grupos de los cuales formaran parte cada unos de los roles, esto de acuerdo a sus funciones principales, además de evaluar el desempeño de cada una de las áreas en tiempos de respuesta, así mismo es el encargado de los costos de las soluciones de las incidencias.

**Rol ServiceDesk.** Aquel usuario dado de alta en la BD con privilegios de asignar las incidencias (tickets) que se registran en la BD, de acuerdo al grupo de soporte que considere el más adecuado para resolver la incidencia, es el encargado de gestionar la BD del conocimiento, y de dar seguimiento a la incidencia hasta su cierre.

**Rol SoporteNivel 1.** Aquel usuario con privilegios de recibir las incidencias (tickets) asignados por ServiceDesk, y de indagar en la BD del conocimiento para la solución previa de la incidencia y reasignará a otro nivel en caso de contar con la solución.

**Rol Soporte Nivel 2.** Aquel usuario con los privilegios similares a Soporte Nivel 2, solo que en este caso este rol recibe incidencias que no se encuentran con solución previa, y debe de resolver la incidencia y guardar la solución en la BD del conocimiento

**Actividades de los Roles involucrados en el sistema.**

**I. Requerimientos del Rol Solicitante**

- a) El Solicitante se encuentra registrado con el rol y privilegios de “Solicitante” en la aplicación.
- b) El Solicitante ingresa a la aplicación por medio de un User y Password, agregados en la BD
- c) El Solicitante registrará Incidencias en el momento que son identificadas, por medio de un formulario específico que deberá llenar.
- d) El Solicitante recibirá un mail con el ID generado por el sistema para su Incidencia, y poder dar seguimiento de la misma.
- e) El Solicitante recibirá un mail del Estatus de su Incidencia cada vez que esta cambie de estatus.
- f) El ID servirá como llave primaria de la aplicación para ver los estatus:
- g) El Solicitante recibe un mail de Solución de incidencia, para su validación.
- h) El Solicitante cerrara la incidencia después de su validación.

**II. Requerimientos del Rol Administrador**

- a) El Administrador se encuentra registrado con el rol y privilegios de “Administrador” en la aplicación.
- b) El Administrador ingresa a la aplicación por medio de un Usset y Password, agregados en la BD.
- c) El Administrador es el encargado de realizar modificaciones en:
  - (i) Soporte
  - (ii) ServiceDesk
  - (iii) Administrar el control de Solicitantes
  - (iv) Administrar el control de Incidencias.
- d) El Administrador revisa periódicamente su Inbox, el cual contiene las incidencias registradas por el sistema hasta ese momento.
- e) El Administrador gestiona el Registro de los Solicitantes, Modifica, Crea y Elimina Grupos de Solicitantes.
- f) El Administrador gestiona el ServiceDesk Modificar, Crear y Eliminar los Grupos de ServiceDesk asignando un Grupo de Solicitante.
- g) El Administrador gestiona el Grupo de Soporte que dará solución a la incidencia de acuerdo al Nivel de Soporte.
- h) El Administrador asigna incidencias a ServiceDesk de acuerdo a su gestión.
- i) El Administrador crea los grupo de categorización de incidencias.
- j) El Administrador genera el escalamiento de las incidencias.
  - (i) Nivel Alto de Afectación. Urgente
  - (ii) Nivel Medio de Afectación. Medio

(iii) Nivel Bajo de Afectación. Bajo

- k) El Administrador es el encargado de extraer el cálculo de incidencias resueltas por mes.
- l) El Administrador asigna la incidencia al equipo de ServiceDesk responsable de solucionar la aplicación.
- m) El Administrador puede observar los detalles de la incidencia enviados por ServiceDesk.
- n) El Administrador envía la solución de la incidencia enviado por ServiceDesk al Solicitante.
- o) El Administrador recibe el mail del Visto Bueno del Solicitante, después de la garantía y envía su Visto Bueno a ServiceDesk, para que cierre la incidencia.
- p) El Administrador es el encargado de generar los KPI's y enviarlos a las áreas correspondientes de ServiceDesk y Usuaría.

**III. Requerimientos del Rol ServiceDesk**

- a) El ServiceDesk se encuentra registrado con el rol y privilegios de "ServiceDesk" en la aplicación.
- b) El ServiceDesk ingresa a la aplicación por medio de un User y Password.
- c) El ServiceDesk revisa su Inbox con las incidencias asignadas por el Administrador, agrega comentarios de ser necesarios y es el encargado de rechazar una solicitud errónea y esperar a que se reabra.
- d) ServiceDesk recibe las notificaciones del estatus de la incidencia.
- e) ServiceDesk Categoriza la incidencia de acuerdo al área de levantamiento.
- f) ServiceDesk Prioriza la incidencia de acuerdo al escalado.
- g) ServiceDesk hace la asignación del primer Nivel de Soporte de Solución.
- h) ServiceDesk trata de encontrar una solución de segundo nivel que no genere costos.
- i) ServiceDesk recibe la solución de Soporte y la envía al Administrador.
- j) ServiceDesk cierra la incidencia una vez que recibe el Visto Bueno. del Administrador.

**IV. Requerimientos del Rol Soporte.**

- a) El Soporte se encuentra registrado con el rol y privilegios de "Soporte" en la aplicación.
  - El Soporte ofrecerá dos Niveles de Solución, en base a las incidencias registradas: Nivel 1 Soporte, Busca una solución a la incidencia en la BD del

Conocimiento y/o resuelve incidencia de prioridad baja. En caso de no tener la solución escala la incidencia a Nivel de Soporte 2.

- Nivel 2 Soporte Resuelve la incidencia reasignada por el Nivel de Soporte 1 al recibir la notificación de ServiceDesk para los costos aprobados, y guarda la solución en la BD del Conocimiento.
- b) El Soporte es el encargado de enviar la solución a ServiceDesk, la cual será cerrará únicamente por el Solicitante, y en automático desaparece del Inbox del Soporte.

## Requerimientos No Funcionales

Los requerimientos NO Funcionales hacen referencia a las características del sistema que aplican de manera general como un todo, son adicionales a los Requerimientos Funcionales del Sistema y hacen hincapié a los Factores de Calidad, es decir, Mantenibilidad, funcionalidad, usabilidad, entre los principales requerimientos no funcionales de nuestro sistema encontramos:

- Desempeño.
- El tiempo de respuesta de la aplicación debe ser automática, así como el envío de correos, notificando a los usuarios el estatus de sus requerimientos e informando a todas las áreas involucradas.

Ahora describiremos el proceso de la funcionalidad del sistema de gestión de incidencias orientado a objetos, con el que posteriormente se describirá el diseño del sistema.

## 3.3 Proceso de SGIOO (Sistema de Gestión de Incidencias Orientado a Objetos).

### Proceso y diagrama del Sistema de Gestión de Incidencias Orientado a Objetos

**Registrar.** En esta etapa se registra la incidencia y se genera el número de ticket correspondiente para su seguimiento.

**Categorizar.** En esta etapa se categoriza la incidencia de acuerdo al área de levantamiento.

**Priorizar.** En esta etapa se prioriza el nivel de afectación de la incidencia.

**Resolver.** De acuerdo a la prioridad se asigna el nivel de soporte correspondiente para resolver la incidencia.

**Soporte Nivel 1.** Busca una solución a la incidencia en la BD del Conocimiento y/o resuelve incidencia de prioridad baja. En caso de no tener la solución escala la incidencia a Nivel de Soporte 2.

**Soporte Nivel 2.** Resuelve la incidencia reasignada por el Nivel de Soporte 1, y guarda la solución en la BD del Conocimiento.

**Cerrar.** Validación de la solución y cierre

3.3 Proceso de SGIOO (Sistema de Gestión de Incidencias Orientado a Objetos).

DIAGRAMA DE PROCESO DE SISTEMA DE GESTIÓN DE INCIDENCIAS ORIENTADO A OBJETOS

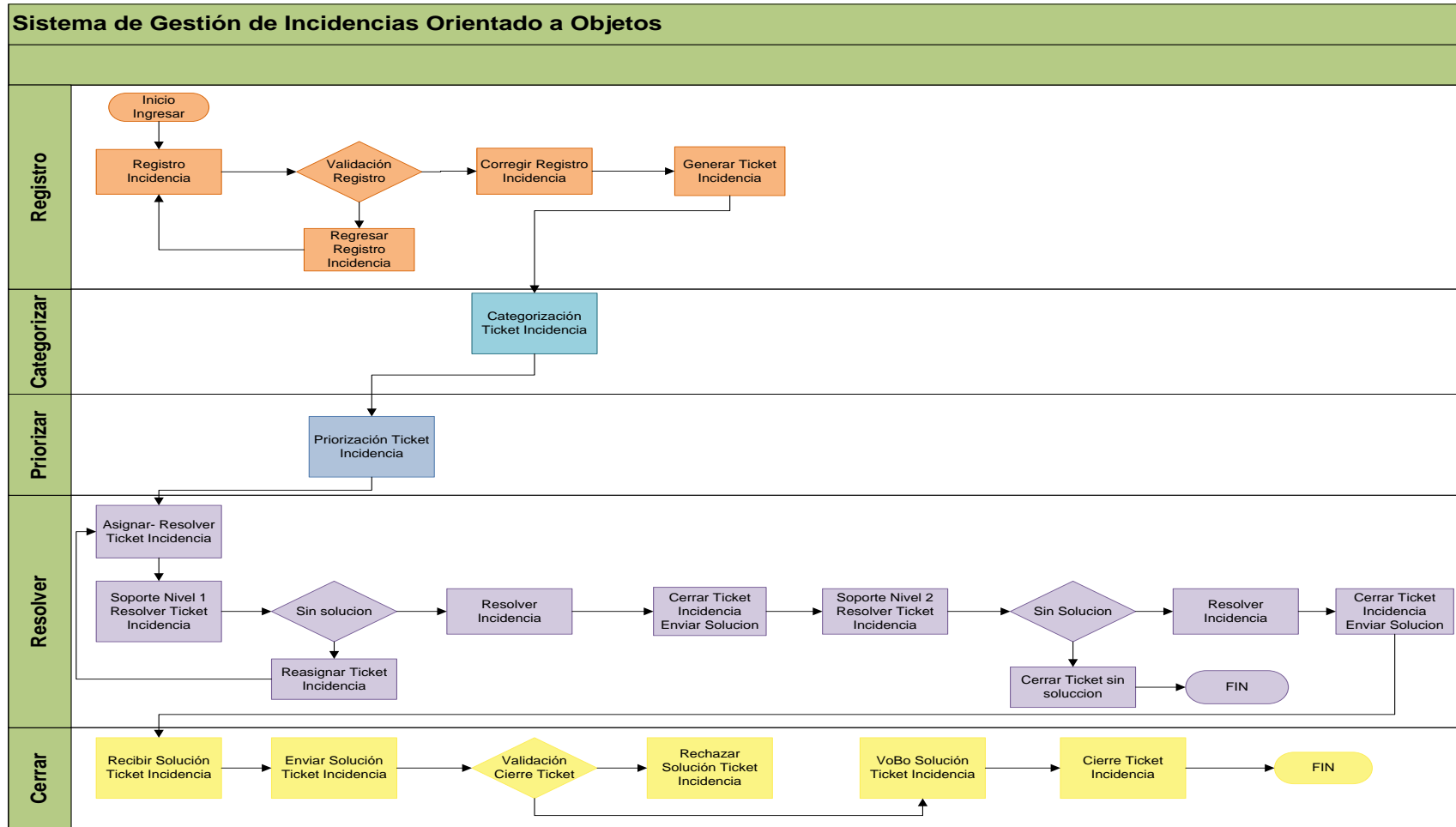


Ilustración 3. Diagrama Proceso



**TABLA. PROCESO DEL DIAGRAMA DE GESTIÓN DE INCIDENCIAS ORIENTADO A OBJETOS elaborada por Anabel Jimenez Cano**

Paso	Nombre Actividad	Responsable	Estado	Descripcion	Tiempo medido (en días)	Costos	Entradas	Salidas
1	Registro Incidencia	Solicitante	Inactivo	El solicitante registra la incidenciasolicitante en aplicación.	1	0	Datos de Incidencia	Registro
2	Validacion Registro Incidencia	Aplicación	Activo	La aplicación valida los datos ingresados por el solicitante	1	1	Datos de incidencia	Ticket Registro Incidencia
3	Validacion Registro Incidencia	Service Desk	Activo	Service Desk valida los datos enviados por el solicitante (Acepta o Rechaza)	1	2	Ticket Incidencia	Ticket Incidencia Validado
4	Validacion Registro Incidencia Rechazado	Solicitante	Activo	El solicitante corrige los datos erroneos en el registro de sus incidencia	1	3	Ticket Incidencia Rechazado	Ticket incidencia Corregido
5	Categorizacion Ticket Incidencia	Service Desk	Activo	Service Desk realiza la categorizacion de la incidencia de acuerdo al area solicitante	1	0	Ticket Incidencia Validado	Ticket Incidencia Categorizado
6	Priorizacion Ticket Incidencia	Service Desk	Activo	Service Desk prioriza el Ticket de Incidencia Categorizado de acuerdo a una tabla de escalado.	1	0	Ticket Incidencia Categorizado	Ticket Incidencia Priorizado
7	Asignar-Resolver Ticket Incidencia	Service Desk	Activo	Service Desk asigna el Ticket Incidencia al grupo de soporte nivel 1 para resolver la incidencia	1	4	Ticket Incidencia Priorizado	Ticket Incidencia Asignado
8.1	Resolver Ticket Incidencia	Soporte Nivel 1	Activo	Soporte Nivel 1 valida la solucion en la BDK de encontrarla cierra el Ticket , caso contrario resuelve el ticket de ser posible con sus recursos.	4	4	Ticket Incidencia Asignado	Ticket Incidencia Resuelto
8.2	Resolver Ticket Incidencia	Soporte Nivel 1	Activo	Soporte Nivel 2 no encuentra solucion en la BDK y no tiene recurso para resolver, se reenvia el ticket a Service Desk para reasignar.	3	5	Ticket Incidencia Asignado	Ticket Incidencia Re-asignado
9	Re-Asignar Ticket Incidencia	Service Desk	Activo	Service Desk re-asigna el Ticket de Incidencia sin solucion en soporte nivel 1 , a soporte nivel 2 despues de un analisis de recursos	3	6	Ticket Incidencia Re-Asgnado	Ticket Incidencia Asignado
9.1	Resolver Ticket Incidencia	Soporte Nivel 2	Activo	Soporte Nivel 2 resuelve el Ticket de incidencia con los recursos aprobados por Service Desk, y registra la solucion en BDK	4	6	Ticket Incidencia Asignado	Ticket Incidencia Resuelto
10	Cierre Ticket Incidencia	Service Desk	Activo	Service Desk recibe el Ticket de incidencia resuelto por Soporte Nivel 1 o Soporte Nivel 2, registra la fecha envia la solucion a Administrador	3	2	Ticket Incidencia Resuelto	Ticket Incidencia Resuelto
10.1	Validacion Cierre Ticket Incidencia	Administrador	Activo	Administrador recibe la solucion enviada por Service Desk y la reenvia a Solicitante, despues de validarla	2	2	Ticket Incidencia Resuelto	Ticket Incidencia Resuelto
10.2	Validacion Cierre Ticket Incidencia	Solicitante	Activo	Solicitante recibe el Ticket de Incidencia Resuelto Valida y Reabre el Ticket al no cubrir la solucion su peticion	2	4	Ticket Incidencia Resuelto	Ticket Incidencia Resuelto Re-Abierto
10.3	Validacion Cierre Ticket Incidencia	Solicitante	Activo	Solicitante recibe el Ticket de Incidencia Resuelto Valida y Cierra el Ticket	2	0	Ticket Incidencia Resuelto	Ticket Incidencia Cerrado
11	Cierre Ticket Incidencia	Service Desk	Activo	Service Desk recibe la notificacion de que el Solicitante cerro el Ticket de Incidencia y Cierra el Ticket de incidencia	2	0	Ticket Incidencia Cerrado	Ticket Incidencia Cerrado

**Tabla 7.** Proceso del Sistema de Gestión de Incidencias Orientado a Objetos

### 3.4 Diseño del Sistema de Gestión de Incidencias Orientado a Objetos.<sup>8</sup>

El presente desarrollo de diagramas fue montado sobre una herramienta Open Source con el objetivo de implementar nuevas técnicas de diseño que no sólo se basen en herramientas con licencias, abriendo con esto el panorama de posibilidades de una mejor etapa de diseño, dicha herramienta responde al nombre de Argo UML.<sup>9</sup>

Describiremos los diagramas creados para el diseño de nuestro sistema, para tener una visión mas clara del objetivo de su funcionalidad.

#### Diagrama de Estados

El Diagrama de Estados, muestra una máquina de estado que consiste de estados, transiciones, eventos y actividades.

En la ilustración 4 se muestra el diagrama de estados de nuestro Sistema de Gestión de Incidencias Orientado a Objetos, el cual cuenta con cinco estados de proceso.

Para mostrar un flujo completo del proceso de nuestro sistema es necesario ejemplificar la apertura y cierre de un ticket de incidencia, mostrando cada uno de los procesos por los que debe de pasar y cómo se relacionan entre sí.

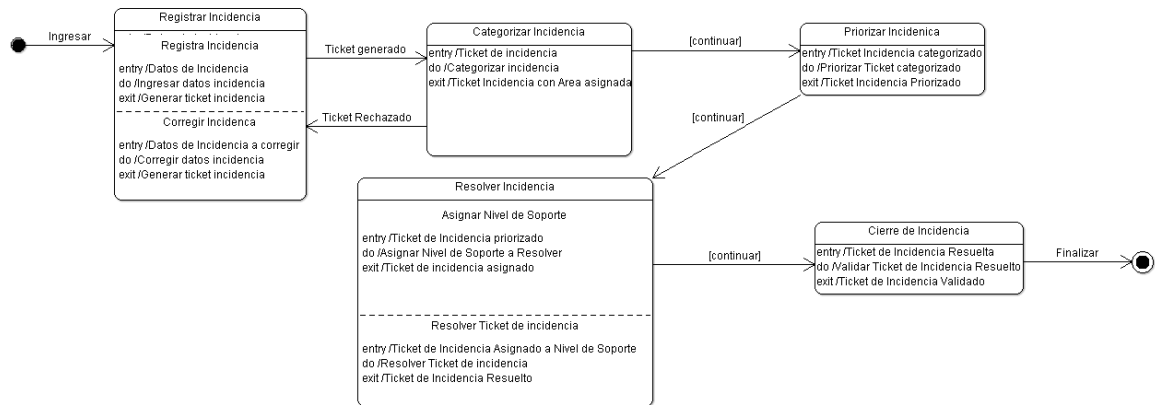


Ilustración 4. Diagrama de Estados SGIOO-Ticket Incidencia

<sup>8</sup>Pressman, R. S. (2002). Ingeniería del Software un enfoque práctico. Madrid España: McGraw Hill 4ta Edición.

<sup>9</sup> <http://argouml.tigris.org/>

### Diagrama de Actividades

El diagrama de actividades es una variante de diagrama de estado que muestra el flujo de una actividad a otra actividad dentro de un sistema.

A continuación siguiendo el flujo de nuestro Ticket de Incidencia, en la ilustración 5 mostramos las actividades por las cuales será necesario que pase, para poder realizar un flujo completo de su funcionalidad, con las actividades de los procesos.

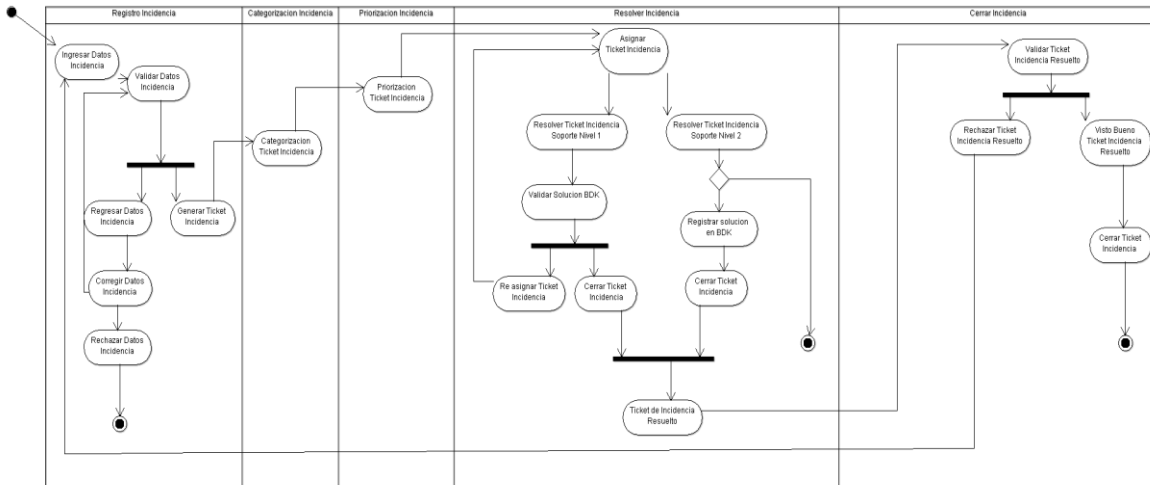


Ilustración 5. Diagrama de Actividades SGIOO

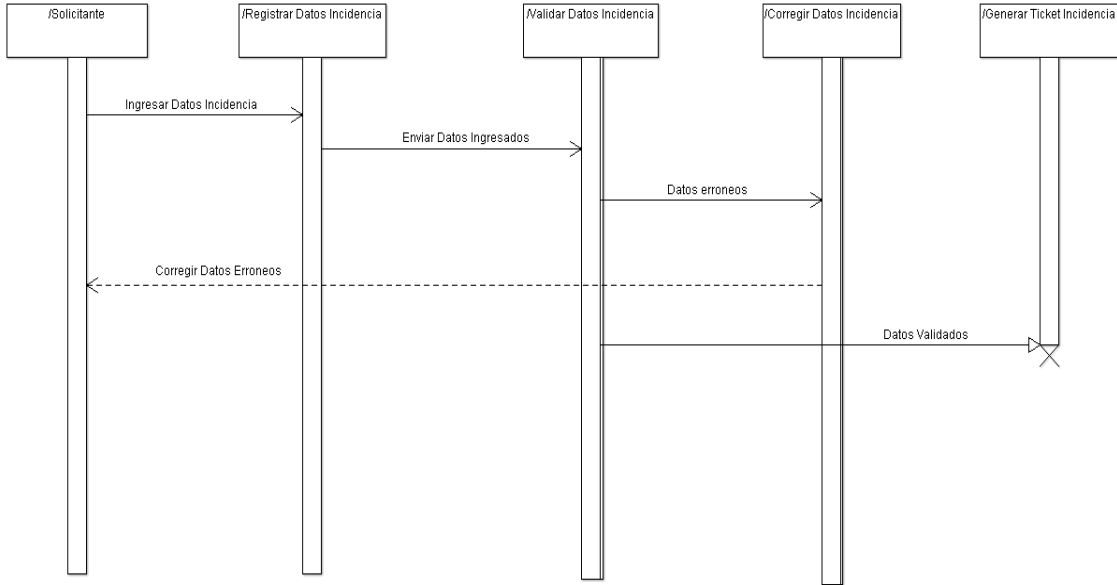
### Diagrama de Secuencia

El diagrama de secuencia muestra configuración de los nodos de procesamiento en tiempo de corrida y los componentes que viven en ellos.

Las ilustraciones de 6 a la 10 describen la secuencia de los procesos que se encontrarán en SGIOO, de acuerdo con el Diagrama de Proceso de SGIOO, en relación con el Ticket de Incidencia.

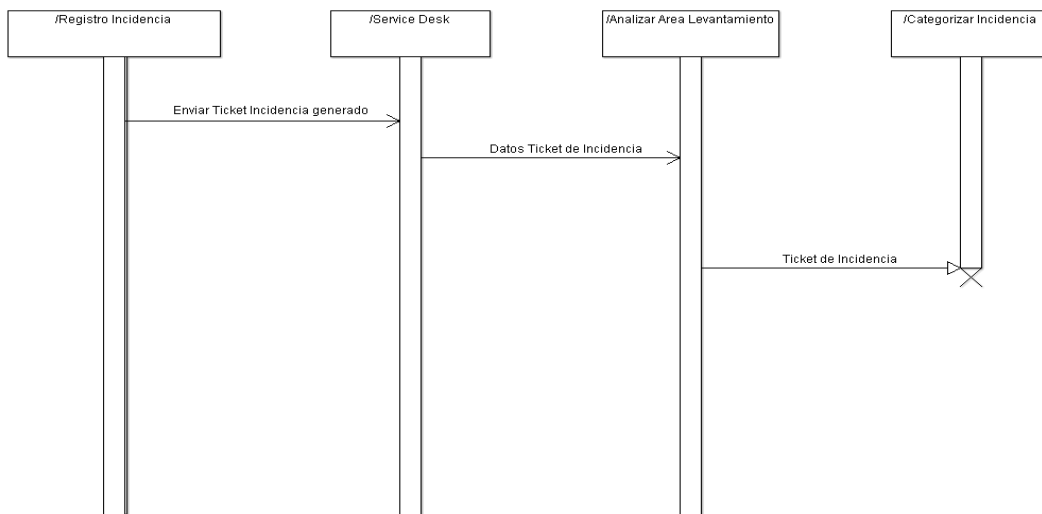
**Diagrama de Secuencia de Registrar.** En la ilustración 6 se describe la apertura de un ticket de incidencia, es decir desde el momento en que el solicitante ingresa los datos que son necesarios para que pueda generarse un ticket de incidencia, así como las validaciones que son necesarias para generar el ticket.

### 3.4 Diseño del Sistema de Gestión de Incidencias Orientado a Objetos



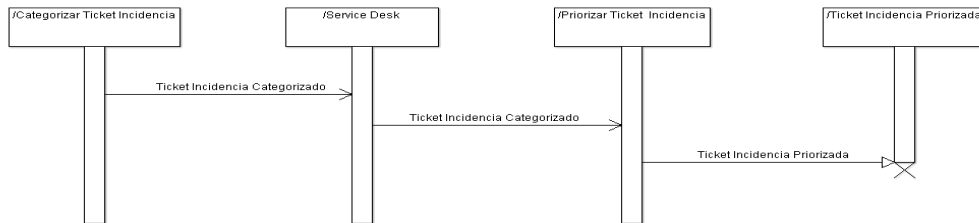
**Ilustración 6.** Diagrama Secuencia Registrar

**Diagrama de Secuencia Categorizar.** La ilustración 7 muestra que una vez que se ha generado el Ticket de Incidencia, es necesario de acuerdo a nuestro flujo de proceso categorizar la incidencia, es decir saber cuál es el área que será la encargada en resolver la incidencia, esto de acuerdo a los datos ingresados por nuestro solicitante, para eso será necesaria la participación del Rol ServiceDesk, el cual asignara la incidencia de acuerdo a esta información.



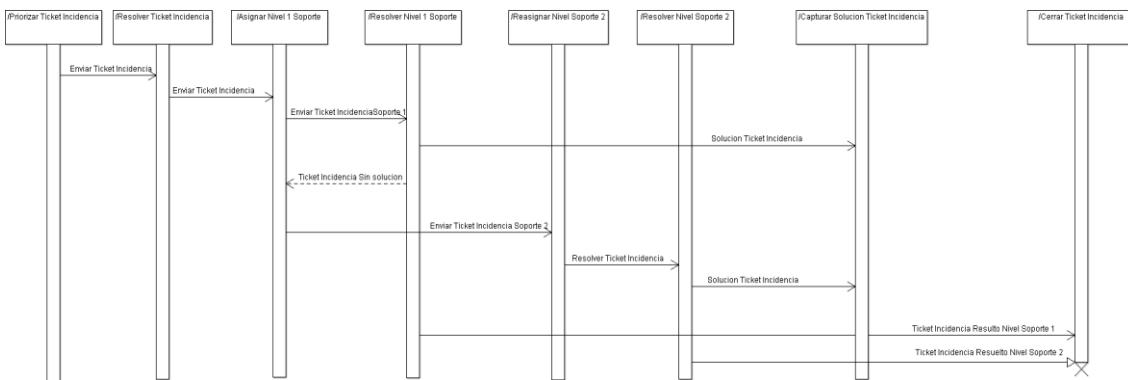
**Ilustración 7.** Diagrama Secuencia Categorizar

**Diagrama de Secuencia Priorizar.** La ilustración 8 muestra que al concluir la categorización, el mismo ServiceDesk, se encargara de priorizar la incidencia de acuerdo a los mismos datos que se ingresaron por el Solicitante, es decir se le dará un grado de solución para resolver o no lo antes posible la incidencia.



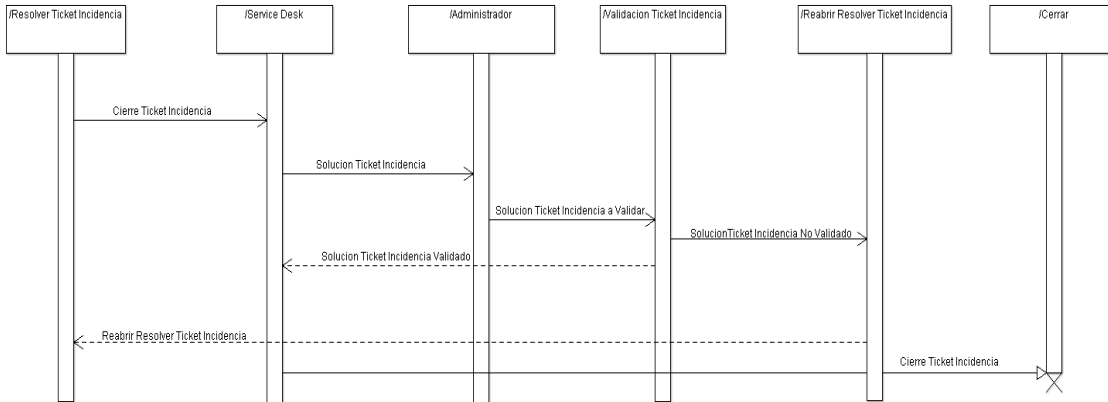
**Ilustración 8.** Diagrama Secuencia Priorizar

**Diagrama de Secuencia Resolver.** En este diagrama participan los niveles de soporte de acuerdo al rol de Soporte, la ilustración 9 muestra que una vez que ServiceDesk ha categorizado y priorizado el Ticket de incidencia, será necesario resolver el Ticket, de acuerdo al análisis que realiza el área de Soporte, es decir, si puede o no ser resuelto por el Nivel 1, y en caso de no poder ser resuelto en este nivel, reasignar el Ticket a Nivel 2, el cual resolverá el Ticket, sin embargo en caso de no contar con recursos económicos se cerrará el Ticket, dicho flujo es de acuerdo a los procesos generales del Sistema, ambos niveles tienen la responsabilidad de guardar la incidencia en la BD del Conocimiento.



**Ilustración 9.** Diagrama de Secuencia Resolver

**Diagrama de Secuencia Cierre.** La ilustración 10 describe como participan los roles de Administrador y ServiceDesk, los cuales son encargados de revisar y enviar la solución al área Solicitante que será el encargado enviar su Visto Bueno para cerrar por completo el Ticket de Incidencia, una vez que ha sido validada y registrada la solución.



**Ilustración 10.** Diagrama de Secuencia Cierre

### Diagrama de Casos de Uso

El diagrama de Casos de Uso muestra un conjunto de casos de uso, actores y sus relaciones<sup>10</sup>

En este diagrama indicaremos las funciones exteriores del sistema que se esperan realicen los Actores involucrados en cada uno de los procesos que se ejecutaran en el Sistema, de este modo tendremos una visión más clara de lo que se espera realice el Sistema.

**Caso de Uso. Administrador.** Administrador debe de controlar los datos del Solicitante. Se debe de tener un control sobre los datos del registro, modificación y eliminación de Solicitantes.

Administrador debe de controlar los datos del Soporte. Se debe de tener un control sobre los datos del registro, modificación y eliminación de Soporte.

<sup>10</sup> Referencia en Anexos

Administrador debe de controlar los datos del ServiceDesk. Se debe de tener un control sobre los datos del registro, modificación y eliminación de ServiceDesk.

Administrador debe de controlar los datos del Solicitudes de Incidencias Registradas. Calcular Índices de acuerdo a la Solución y Cierres de Tickets de Incidencia.

Para más detalle consultar Anexo B página 82.

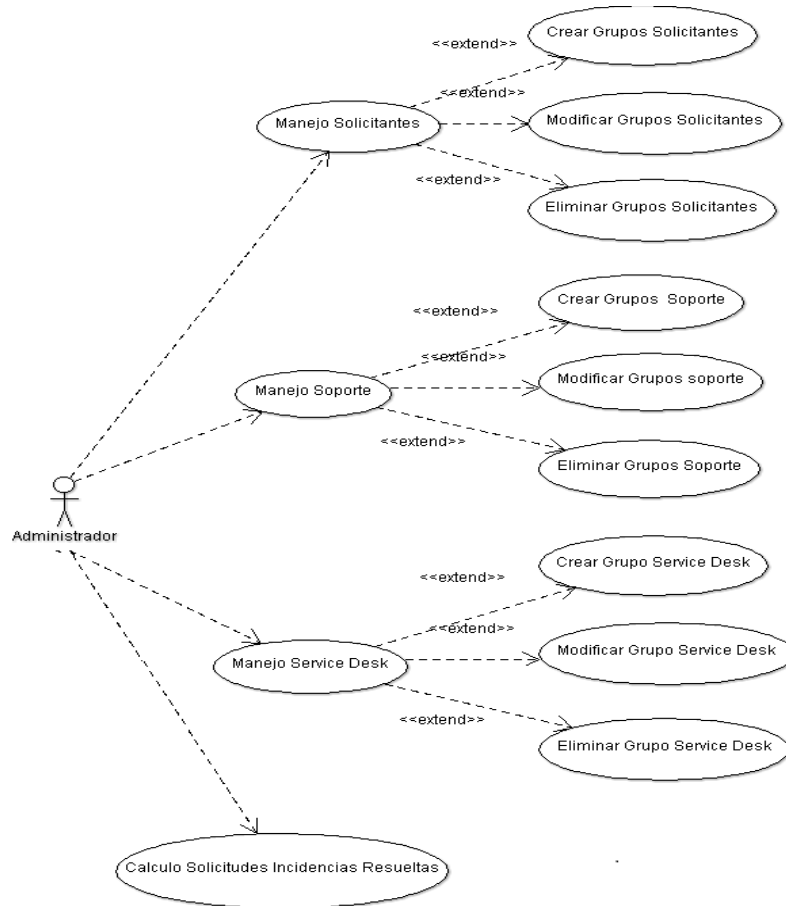


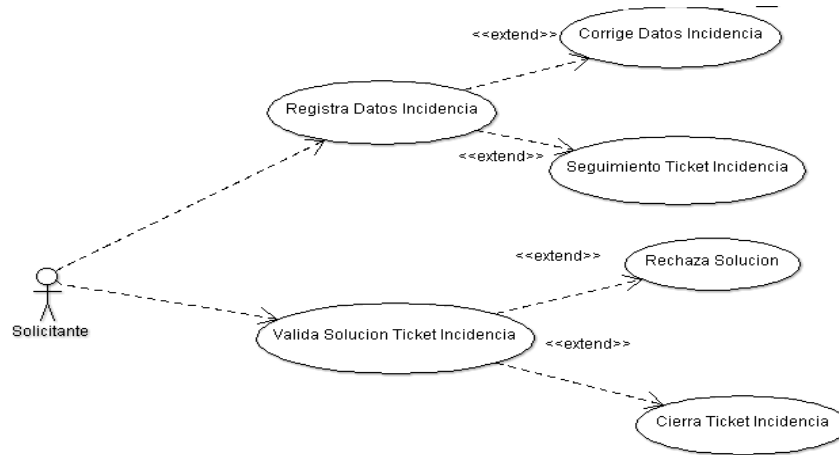
Ilustración 11. Caso de Uso Administrador

**Caso de Uso. Solicitante.** El solicitante ingresa a la aplicación para realizar un alta de incidencia. Generar Ticket de Incidencia

El solicitante ingresa a la aplicación para Cerrar el Ticket de Incidencia. Cerrar Ticket de Incidencia.

Para más detalle consultar Anexo B página 88.

### 3.4 Diseño del Sistema de Gestión de Incidencias Orientado a Objetos



**Ilustración 12.** Caso de Uso Solicitante

**Caso de Uso. ServiceDesk.** ServiceDesk, valida los datos introducidos por el Solicitante, los cuales deben de ser correctos, y corresponder a las áreas asignadas. Validar Datos de Incidencia, ingresados por el Solicitante.

ServiceDesk, Categoriza la incidencia, de acuerdo al Grupo de Soporte correspondiente al Área Solicitante. Categorizar el Ticket de Incidencia, generado por el Sistema.

ServiceDesk, Categoriza la incidencia, de acuerdo al Grupo de Soporte correspondiente al Área Solicitante. Priorizar el Ticket de Incidencia, Categorizado por ServiceDesk.

ServiceDesk, asigna el Grupo de Soporte de Nivel 1 o Nivel 2, de acuerdo a las especificaciones de la incidencia. Asignar-Resolver Ticket de Incidencia, categorizado y priorizado.

ServiceDesk, cierra la Solución del Ticket de Incidencia validada por el Solicitante. Cierre del Ticket de Incidencia, asignado a un Nivel de Soporte por ServiceDesk.

Para más detalle consultar Anexo B página 91.



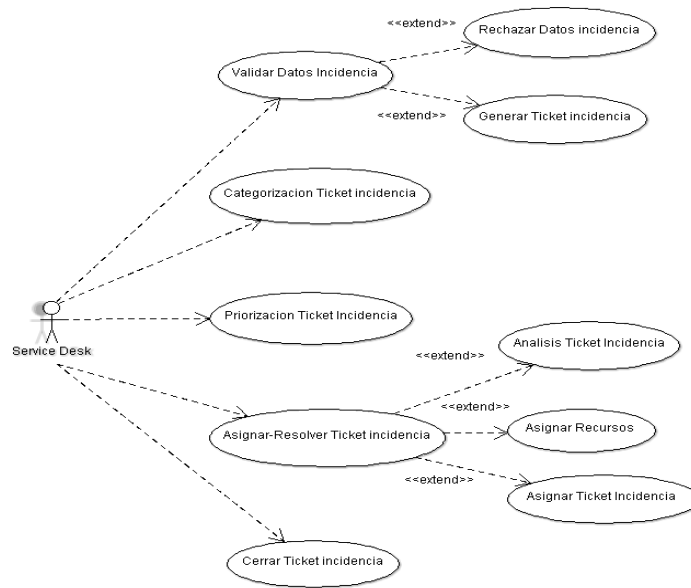


Ilustración 13. Caso de Uso ServiceDesk

**Caso de Uso. Soporte Nivel 1.** Soporte Nivel 1 ingresa a su Inbox, para ver sus Tickets de Incidencias asignados por ServiceDesk y resolver el Ticket. Soporte Nivel 1, analiza la incidencia, y revisa en la BDK si existe una incidencia similar resuelta.

Para más detalle consultar Anexo B página 99.

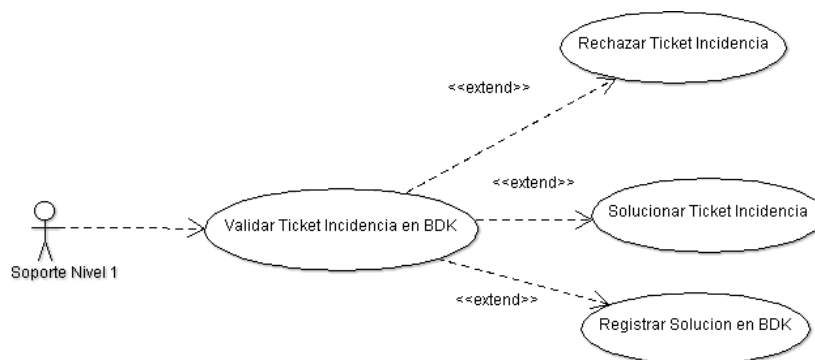
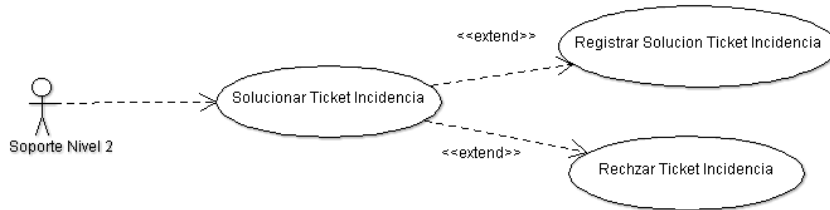


Ilustración 14. Caso de Uso Soporte Nivel 1

**Caso de Uso. Soporte Nivel 2.** Soporte Nivel 2 ingresa a su Inbox, para ver sus Tickets de Incidencias asignados por ServiceDesk y resolver el Ticket, registra la solución en la BD del Conocimiento.

Para más detalle consultar Anexo B página 100.

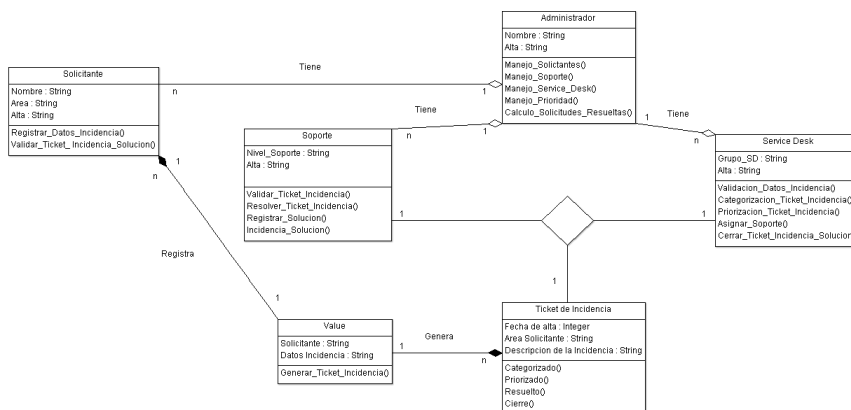


**Ilustración 15.**Caso de Uso Soporte Nivel 2

### Diagrama de Clases

Un diagrama de clases sirve para visualizar las relaciones entre las clases de un sistema, las cuales pueden ser asociativas, de herencia, de uso y de contenido.

En la ilustración 16, mostraremos el Diagrama de Clases de nuestro Sistema, en el cual se describe la relación existente entre las clases que lo comprenderán.



**Ilustración 16.**Diagrama de clases

# Capítulo 4

## Implementación de Métricas

*"Cuando se puede medir aquello de lo que se habla y se puede expresar en números, se conoce algo del tema; pero cuando no se puede medir, cuando no se puede expresar en números, el conocimiento es pobre e insatisfactorio: puede ser el principio del conocimiento"*  
Lord Kelvin

La medición del software es una disciplina relativamente joven, y no existe consenso general sobre la definición exacta de los conceptos y terminología que maneja.<sup>11</sup>

Las Métricas de Software se clasifican en dos grandes grupos principalmente, los cuales son:

Del Producto. Son obtenidas automáticamente tomando como entrada el código fuente: tamaño, estructura de datos y lógica.

Del Proceso. Dependen del entorno del desarrollo, de factores como: tiempo de desarrollo, reusabilidad y productividad.

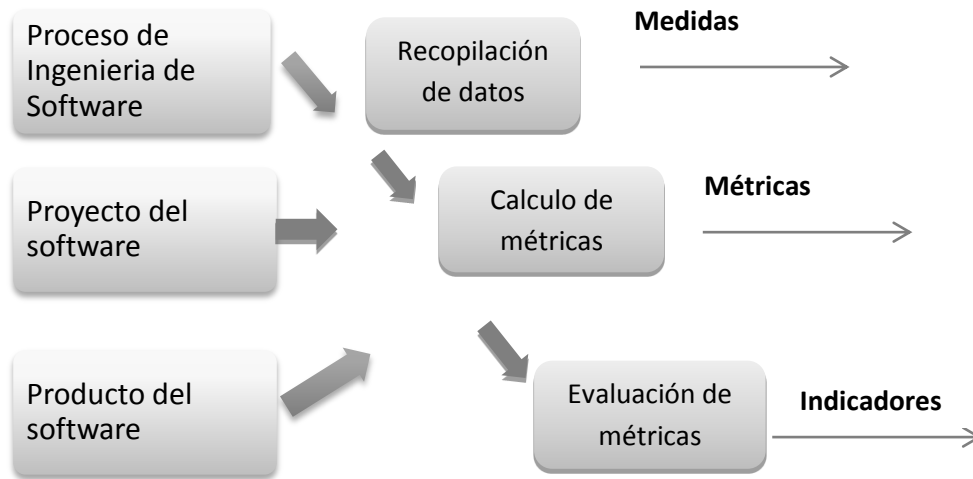
Proceso de Medición de Software propuesto por Roche:

- Formulación. Obtención de medidas y métricas de software apropiadas para la representación del software.
- Colección. Mecanismo empleado para acumular datos necesarios para obtener las métricas formuladas.
- Análisis. Cálculo de las métricas y aplicación de herramientas matemáticas.
- Interpretación. Evaluación de resultados de métricas en esfuerzo por conseguir una visión interna de la calidad de la representación.
- Realimentación (feedback). Recomendaciones obtenidas de la interpretación de métricas técnicas transmitidas al equipo que construye el software.

---

<sup>11</sup>[http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/gonzalez\\_d\\_h/capitulo6.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/gonzalez_d_h/capitulo6.pdf)

Proceso de la recopilación de las métricas de software.



**Ilustración 17.** Proceso Recopilación de Métricas

Los objetivos de la medición deben de ser claros, para cada uno de los involucrados, ya que las necesidades varían de acuerdo a las actividades que cada uno desempeña.

Se debe medir para:

- Indicar la calidad del producto.
- Evaluar la productividad de la gente involucrada en el proyecto.
- Evaluar los beneficios derivados del uso y herramientas de software.
- Ayudar al uso de nuevas herramientas.

Por ejemplo:

Rol	Mediciones
Project Manager	<ul style="list-style-type: none"> <li>• El costo de los diferentes procesos de la producción del software.</li> <li>• La productividad de la plantilla para determinar los pagos.</li> <li>• La calidad de los productos para poder comparar diferentes proyectos.</li> <li>• Definir objetivos medibles para los proyectos.</li> <li>• Medir repetidamente atributos de recursos y procesos con el fin de determinar los factores que afecten al coste y a la productividad</li> </ul>
Ingenieros	<ul style="list-style-type: none"> <li>• Monitorizar la calidad de la evolución de los sistemas a través de la medición de los procesos, esto incluye cambios hechos durante el diseño del sistema o los errores encontrados en la fase de testeo.</li> <li>• Especificar requisitos de calidad y realización en términos medibles estrictamente.</li> <li>• Atributos del proceso y atributos para la certificación</li> </ul>

**Tabla 8.** Objetivos por Roles de la Medición de Software

Así mismo podemos utilizar las mediciones en:

- La evaluación
- Predecir características importantes de los proyectos.

Y poder alcanzar:

- Medidas y modelos de estimación de coste y esfuerzo.
- Modelos y medidas de productividad.
- Recolección de datos
- Medidas y modelos de calidad.
- Modelos de fiabilidad.
- Complejidad computacional o algorítmica.

### **Clasificación de las Métricas de Calidad de Software**

Existen diferentes tipos de métricas en las cuales se basa la Calidad de Software entre las cuales podemos mencionar: <sup>12</sup>

- Métricas del Software, las que se relacionan con el desarrollo del software.
- Métricas de Calidad, como voy a medir que el sistema se adapte a los requisitos que me pide el cliente.
- Métricas de Productividad se basa en el rendimiento del proceso de la ingeniería del software.
- Métricas Orientadas a la Persona, proporciona información sobre la forma en la que la gente desarrolla software y la efectividad de las herramientas y métodos.
- Métricas Orientadas al Tamaño, es saber en que tiempo voy a terminar el software y cuantas personas voy a necesitar.
- Métricas Orientadas ala Función, son medidas indirectas del software se centran en la funcionalidad o utilidad del programa.
- Métricas Técnicas. Se centran en las características de software por ejemplo: la complejidad lógica, el grado de modularidad. Mide la estructura del sistema, el cómo está hecho.
- Métricas del Modelo de Análisis. En esta fase las métricas técnicas proporcionan una visión interna a la calidad del modelo de análisis con la intención de predecir el tamaño del sistema resultante así que es probable que el tamaño y la complejidad del diseño estén directamente relacionados.

Dicho lo anterior utilizaremos las Métricas Técnicas para ejemplificar nuestro Diseño.

### **Diseño de Métricas**

Ahora describiremos los pasos sugeridos por LemEjigo 1191 para diseñar métricas.

---

<sup>12</sup><http://fcqi.tij.uabc.mx/usuarios/luisgmo/data/5.5%20gestion%202007.pdf>

**Paso 1. Definiciones claras.**

Consiste en homogeneizar términos entre todos los involucrados, tanto usuarios como mantenimiento con el objetivo de tener claros los puntos a medir, escogiendo definiciones para todos.

**Paso 2. Definir el modelo**

En este paso el modelo nos define como calcular la métrica, consiste en incluir un elemento de simplificación, por lo que debemos recordar que el objetivo no es crear un modelo perfecto sino uno que cubra nuestras principales necesidades y que se apegue a éstas, podemos crear nuestro propio modelo, o en su defecto acomodar un modelo ya creado a nuestra necesidad.

**Paso 3. Establecer un criterio de conteo.**

Ahora debemos de dividir nuestro modelo, en métricas primitivas, es decir la definición de los primeros datos necesarios para implantar la métrica, por ejemplo, número de servidores, números de usuarios, etc.

**Paso 4. Decidir ¿Qué es bueno?**

Una vez decidido qué medir y cómo medirlo debemos definir qué haremos con el resultado, para esto es importante el apoyo de las cuatro metas de las métricas las cuales nos definen lo siguiente:

- La meta debe ser razonable, si la meta es irreal ésta deberá ser ignorada.
- La meta debe estar asociada a un marco de tiempo
- La meta debe fundamentarse en acciones sustentadas.
- La meta debe ser dividida en partes

**Paso 5. Reporte de la Métrica**

El siguiente paso consiste en decidir como reportar la métrica, es decir la definición del formato del reporte, la obtención de los datos, mecanismos de reporte de distribución y disponibilidad, el reporte se realiza en cuatro pasos:

- Ciclo de obtención de datos, define que la métrica requiere datos y en qué momento los elementos están disponibles para el cálculo de la métrica.
- Ciclo de reporte de datos, define con qué frecuencia el reporte es generado y cuándo se prepara para su distribución.
- Mecanismos de reporte, proyectan los mecanismos de entrega de las métricas.

- Distribución y disponibilidad, describe quienes reciben copias del reporte o tiene acceso a la métrica, también aquí se define cualquier restricción de acceso a la métrica.

**Paso 6. Calificadores Adicionales.**

Los calificadores añadidos proveen la información estadística necesaria para varios puntos de vista de la métrica, los calificadores necesitan ser definidos como parte del diseño de las métricas, ya que éstos determinan el segundo nivel de los requerimientos de la recolección de datos.

## **4.1 Métricas Aplicadas al diseño del Sistema de Gestión de Incidencias Orientado a Objetos**

Al parecer la credibilidad de las métricas aplicadas a un diseño se ve envuelta en la duda de su eficiencia, pero cómo saber si funciona o es eficiente, si realmente nadie toma esta medida previa y experimenta sus resultados, entonces así nunca podremos saber la efectividad de las mismas.<sup>13</sup>

Algunas de las métricas expuestas no contienen alguna validez teórica o experimental, a continuación describimos algunas de las métricas que se pueden aplicar a nuestro diseño, todo esto en relación a los diagramas que se describieron en el capítulo 3 página 22.

### **Métricas del Diseño de Software.**

Ahora definiremos algunas de las métricas que nos ayudaran para nuestro diseño de Sistema de Gestión de Incidencias Orientado a Objetos.

Con el objetivo de reducir la complejidad de programas cada vez más largos y difíciles de escribir y de leer, los diseñadores de software han hecho uso de las técnicas de modularización y diseño estructurado.

Algunas de las ventajas al momento de utilizar estas técnicas son:

- Comprensibilidad. Tanto programadores como usuarios pueden entender mejor el sistema.
- Manejabilidad. Los Project Manager tiene facilidad de asignar a sus recurso en cada modulo.
- Eficiencia. El esfuerzo de implementación se reduce.

---

<sup>13</sup><http://www.uv.mx/personal/asumano/files/2012/08/MetricasTecnicas.pdf>

#### 4.1 Métricas Aplicadas al diseño del Sistema de Gestión de Incidencias Orientado a Objetos

- Reducción de errores. Los planes de pruebas son simplificados.
- Reducción del esfuerzo de mantenimiento. La división de módulos favorece son llevados por diferentes módulos de trabajo.

##### **Métricas de Puntos de Función.**

Estas métricas predicen el tamaño de un sistema a partir de un Modelo de Análisis.

Utilizado en Diagramas de Flujo de Datos o Casos de Uso, la métrica permite determinar el tamaño final del sistema en base a los requerimientos, entre sus factores a medir encontramos:

- *Número de entradas del usuario.*
- *Número de salidas del usuario.*
- *Número de consultas del usuario.*
- *Número de archivos.*
- *Número de interfaces externas.*

Fórmula:

$$PF = (cuenta - total) * [(0.65 + 0.01) * \sum Fi]$$

donde:

*cuenta - total = suma de todas las entradas PF*

*Fi = valores de ajuste de complejidad*

Es independiente del lenguaje y la tecnología utilizada en la implementación del sistema, es una métrica de estimación, por lo cual la calidad del plan de desarrollo depende de la calidad de sus estimados, estos a su vez deben ser refinados en la medida que se conoce

La metodología de esta métrica es la siguiente:

- *Paso 1. Determinar el tipo de conteo y el alcance del conteo.*
- *Paso 2. Identificar la frontera a seguir.*
- *Paso 3. Contar la funcionalidad de datos.*
- *Paso 4. Contar la funcionalidad de transacciones.*
- *Paso 5. Determinar los puntos de función sin ajustar.*
- *Paso 6. Determinar el factor de valor de ajuste (VAF).*
- *Paso 7. Determinar el tipo de conteo y el alcance de conteo.*



**Paso 1. Determinar el tipo de conteo y el alcance del conteo.**

En la métrica de puntos de función, existen tres tipos de conteo los cuales son:

- Conteo de los proyectos por primera vez. Mide la funcionalidad del proyecto proporcionado al usuario
- Conteo por mantenimiento del proyecto. Mide las modificaciones por cambios de una aplicación.
- Conteo actualizado del proyecto. Mide una aplicación ya contada.

En este Paso elegimos el Conteo de los proyectos por primera vez, ya que nuestro sistema aun no se encuentra implementado.

**Paso 2. Identificar la frontera a seguir.**

La frontera de la aplicación<sup>14</sup> indica el límite entre el software que se está midiendo y el usuario.

En nuestro sistema podemos encontrar las siguientes fronteras, con sus correspondientes factores a medir:

Medidas para Solicitante:

- Número de entradas del solicitante:
- Número de salidas del solicitante.
- Número de consultas del solicitante.
- Número de archivos.
- Número de interfaces externas.

Medidas para Administrador:

- Número de entradas del administrador.
- Número de salidas del administrador:
- Número de consultas del administrador:
- Número de archivos.
- Número de interfaces externas.

Medidas para ServiceDesk:

- Número de entradas de servicedesk.
- Número de salidas de servicedesk.
- Número de consultas de servicedek.

---

<sup>14</sup>Anexo A

#### 4.1 Métricas Aplicadas al diseño del Sistema de Gestión de Incidencias Orientado a Objetos

- Número de archivos.
- Número de interfaces externas.

Medidas para Soporte:

- Número de entradas de soporte.
- Número de salidas de soporte.
- Número de consultas de soporte.
- Número de archivos.
- Número de interfaces externas.

Basaremos nuestro ejemplo en la frontera de ServiceDesk, debemos aclarar que la estimación será imprecisa debido a la escasa información que se tiene en un principio acerca del sistema.

ServiceDesk, funcionalidad:

- Revisar Inbox con las incidencias rechaza una solicitud errónea y esperar a que se reabra.
- Recibir notificaciones del estatus de la incidencia.
- Categorizar la incidencia de acuerdo al área de levantamiento.
- Priorizar la incidencia de acuerdo al escalado.
- Asignar el Nivel de Soporte de Solución.
- Encontrar una solución de segundo nivel que no genere costos.
- Recibir la solución de Soporte y enviarla al Administrador.
- Cerrar la incidencia una vez que recibe el Visto Bueno. del Administrador.

#### ***Paso 3. Contar la funcionalidad de datos.***

Representan la funcionalidad proporcionada al usuario de grupos de datos o información de control mantenidos dentro de la aplicación o leída desde otras aplicaciones.

- ALI. Archivos Lógicos Internos. Grupo identificable de datos relacionados lógicamente o de información de control que pertenecen al usuario. Mantenido dentro de las fronteras del sistema

Ejemplo. En nuestro ejemplo, donde los principales escenarios son: categorizar, priorizar, asignar, cerrar así como dar y recibir solución, podemos identificar como Archivos lógicos Internos, al Ticket de Incidencia y la solución debido a que estos son los datos que se relacionan con el sistema.

Por lo tanto: ALI = 2

- AIE. Archivos de interfaz Externa. Grupo identificable de datos relacionados lógicamente o de información de control que pertenece al usuario.

Referido por la aplicación, pero mantenido por las entradas externas de otras aplicaciones.

Ejemplo. En nuestro ejemplo un archivo de interfaz externa sería una solicitud errónea y encontrar una solución de segundo nivel ya que esta solución no se encuentra dentro de nuestro sistema pues únicamente se guardan las soluciones de primer nivel.

Por lo tanto AIE = 2

#### ***Paso 4. Contar la funcionalidad de transacciones***

- EE. Entradas Externas. Proceso que procesa datos o información de control que bien fuera de la frontera de la aplicación para mantener uno o más ALI. Altera el comportamiento del sistema.

Ejemplo. En nuestro ejemplo tenemos como límites de fronteras, los escenarios de recibir, encontrar solución y cerrar, ya que los datos que se manejan o cruzan la frontera de nuestro sistema, pues se relacionan fuera de esta frontera:

Por lo tanto EE= 3

- SE. Salidas Externas. Es un proceso lógico elemental o fórmula matemática crear y enviar datos o información fuera de control de la frontera de la aplicación.

Ejemplo. En nuestro ejemplo, para este caso no contamos con salidas externas debido a que ServiceDesk no genera algún reporte o proceso lógico elemental.

Por lo tanto SE = 0

- CE. Consultas Externas. Proceso elemental que envía datos o información de control fuera de la frontera de la aplicación.

Ejemplo. En nuestro ejemplo tenemos con consulta externa la solución de la incidencia, el ticket de incidencia.

Por lo tanto CE = 2

**Paso 5. Determinar los puntos de función sin ajustar**

Asignamos los niveles de dificultad para los archivos lógicos Internos o archivos de interfaz externa.

Primero vemos que tenemos de nuestro ejemplo los siguientes datos:

ALI = 2, ticket de incidencia y solución.

AIE = 2, solicitud errónea y encontrar solución.

EE= 3, recibir incidencia, encontrar solución, y cerrar ticket incidencia.

SE = 0

CE = 2, solución de incidencia, y ticket de incidencia.

Ahora asignaremos el valor de complejidad

Donde RA es el numero de Archivos Lógicos Internos (ALI) mantenidos por la entrada externa (EE), y TDE representa la cantidad de elementos que componen la Entrada Externa (EE).

	1 - 4TDE	5 - 15 TDE	15 o + TDE
0 - 1 RA	Simple	Simple	mediano
2 RA	<b>Simple</b>	Mediano	complejo
3 o + RA	Mediano	Complejo	complejo

**Tabla 9.**Niveles de dificultad Puntos de Fusión EE

Por lo tanto. TRE = 2 y TDE = 3, de acuerdo a los valores de ALI y EE, al hacer el cruce vemos que ambos son de complejidad simple.

Así que EE = complejidad simple, 2.

Para la clasificación de salidas y consultas externas emplearemos la siguiente tabla:

Donde RA representa el número de Archivos Lógicos Internos (ALI) o Archivos de interfaz Externa (AIE), vinculados con la salida externa o consulta externa y TDE, representa la cantidad combinada de elementos de datos de entrada y salida que componen la Salida Externa o Consulta Externa.

	1 – 5 TDE	6 - 19 TDE	19 o + TDE
0 - 1 RA	Simple	Simple	mediano
2 – 3 RA	<b>Simple</b>	Mediano	complejo
3 o + RA	Mediano	Complejo	complejo

**Tabla 10.** Asignación de pesos Puntos de Fusión CE y SE

Por lo tanto RA = 2 y TDE = 4 (2 ticket uno sin solución y otro con solución, 2 soluciones, la q se registra o la que ya estaba registrada), de acuerdo a los valores de CE y SE., al hacer el cruce en la tabla vemos que ambos son de complejidad simple.

Así que CE = complejidad simple, 4 y SE = complejidad simple, 2.

Para la clasificación de Archivos Lógicos Internos y Archivos de Interfaz Externos emplearemos la siguiente tabla, donde TRE representa un subgrupo de elementos reconocibles por el usuario, y TRE representa la cantidad de elementos de datos básicos (campos únicos) que componen el archivo.

	1 – 19 TDE	20 - 50 TDE	51 o + TDE
0 - 1 TRE	Simple	Simple	mediano
2 – 5TRE	<b>Simple</b>	Mediano	complejo
6 o + TRE	Mediano	Complejo	complejo

**Tabla 11.** Asignación de pesos Puntos de Fusión ALI y AIE

Por lo tanto TRE = 2 y TDE = 2 de acuerdo a los valores de ALI y AIE, al hacer el cruce vemos en la tabla que ambos son de complejidad simple.

Así que ALI = complejidad simple, 2 y AIE = complejidad simple, 2.

Asignación de valores numéricos. Los valores numéricos que se asignan a cada complejidad (Baja, Mediana o Alta), se muestran en la siguiente tabla, para cada uno de los tipos de transacción.

#### 4.1 Métricas Aplicadas al diseño del Sistema de Gestión de Incidencias Orientado a Objetos

Puntos de Fusión sin ajustar				
Indicador	Simple	Mediano	Complejo	SUMA
ALI	2 * 7	0 * 10	0 * 15	14
AIE	2 * 5	0 * 7	0 * 10	10
EE	2 * 3	3 * 4	0 * 6	6
SE	2 * 4	0 * 5	0 * 7	8
CE	4 * 3	0 * 4	0 * 6	12
				50

**Tabla 12.** Asignación de pesos Puntos de Fusión

Obtención de Puntos de Función sin Ajustar UFP o(*cuenta – total*)

Suma del total, por lo tanto (*cuenta – total*) = 50.

#### ***Paso 6. Determinar el factor de valor de ajuste (VAF).***

Consiste en el cálculo de un Factor de Ajuste en base a la cuantificación de ciertos coeficientes vinculados con las características deseadas del sistema, a cada una de estas características se le asigna un factor de peso que indica la importancia de la característica para el sistema bajo análisis.

1.- Comunicación de datos. Describe el grado en el que el aplicativo se comunica directamente con el procesador.

2.- Procesamiento distribuido de datos. Mide el grado en el que el aplicativo transfiere datos entre componentes de la aplicación.

3.- Rendimiento. Es crítico ya que tiene influencia sobre como diseñar, desarrollar o implementar.

4.- Configuración altamente usada. El software será implementado en un entorno existente y fuertemente utilizado.

5.- Promedio de Transacciones. Alto promedio de transacciones tiene influencia al diseño, desarrollo, implementación y soporte.

6.- Entrada de datos en línea. El software requiere entradas interactivas.

7. Eficiencia para el usuario final. Las funciones en línea proveídas tendrán que enfatizar un diseño para la eficiencia del usuario final.

8.- Actualización en línea. Se necesitara la actualización de archivos maestros en forma interactiva.

9.- Procesamiento complejo. Describe el grado en el cual el procesamiento lógico influencia en el desarrollo de la aplicación.

10. Reusabilidad. Describe el grado en el que la aplicación ha sido diseñada y desarrollada y soportadas para reutilizar.

11.- Facilidad de instalación. Describe el modo en el que los medios ambientes previos afectaran a la aplicación.

12. Facilidad de operación. Describe el grado de atención a aspectos operacionales tales como: salvar, y recuperación de datos.

14.- Varios sitios. Describe el grado en el que la aplicación será diseñada, desarrollada e implementada en múltiples localizaciones y organizaciones de usuarios.

15.- Facilidad de Cambios. Describe el grado en el que la aplicación ha sido desarrollada para la modificación fácil de procesamiento.

VALORES DE AJUSTE DE COMPLEJIDAD

0 = Sin Influencia

1 = Accidental

2 = Moderado

3 = Medio

4 = Significativo

5 = Esencial

Ahora aplicamos valores de complejidad, de acuerdo al valor que creamos es el más conveniente para cada descripción de factores:

Característica	Descripción	Valor de ajuste de complejidad
Comunicación de datos	Aplicación Web	5

4.1 Métricas Aplicadas al diseño del Sistema de Gestión de Incidencias Orientado a Objetos

Procesamiento distribuido de datos	Ticket de Incidencia y solución de la incidencia.	5
Rendimiento	Requisitos de rendimientos aprobados por el usuario.	5
Configuración altamente usada.	Hardware impuesto de Marca XXX Modelo YYY.	5
Promedio de Transacciones.	Diario	4
Entrada de datos en linera.	Todas la entradas son interactivas.	5
Actualización en línea	Todas	5
Procesamiento Complejo	BD de Conocimiento	3
Reusabilidad	Se pretende utilizar parte del sistema a futuro.	3
Facilidad de instalación	No Aplica	0
Facilidad de Operación	Efectiva a la solución de incidencias	5
Varios Sitios	Al menos para los usuarios finales	5
Facilidad de cambios	Mejor desarrollo a futuro	4

**Tabla 13.**Estimadores Puntos de Fusión

Grado total de Influencia, es la suma de los valores de complejidad.

$$\sum Fi = 54$$

**Paso 7. Determinar el tipo de conteo y el alcance de conteo.**

Aplicando la formula

$$PF = (cuenta - total) * [(0.65 + 0.01) * \sum Fi$$



$$(cuenta - total) = 50$$

$$\sum Fi = 54$$

Por lo tanto

$$PF = (50) * [0.65 + .01 * 54]$$

$$PF = (50) * [0.65 + .54]$$

$$PF = (50) * (1.19)$$

$$PF = 59.5 \text{ redondeado } PF = 60$$

Este resultado hace referencia a las líneas de código que se utilizaran para este modulo. Una vez obtenido este dato, se deben aplicar coeficientes que conviertan este valor a otros como el esfuerzo, el costo o el tiempo, estos coeficientes se obtienen de la información histórica de proyectos de la organización,, aunque pueden existir valores medios, recopilados estadísticamente de la industria del software.

### **Calidad de la Especificación**

Hace referencia a la calidad de la especificación de los requerimientos.

Incluye las características de: Especificidad, compleción, corrección, comprensión, verificación, consistencia, logro, concisión, trazabilidad, modificación, exactitud, reutilización.

Sea

$$Nr = Nf + Nnf$$

donde

$$Nr = \text{Número de requerimientos}$$

$$Nf = \text{Número de requerimientos funcionales}$$

$$Nnf = \text{Número de requerimientos NO funcionales}$$

Especificidad (Ausencia de Ambigüedad)

$$Q1 = \frac{Nui}{Nr}$$

#### 4.1 Métricas Aplicadas al diseño del Sistema de Gestión de Incidencias Orientado a Objetos

donde

*Nui = Num. de req. donde todas los revisores tuvieron inerpretaciones directas*

Ejemplificando la métrica de Calidad de la Especificación, en nuestro diseño vemos que:

Contando el número total de requerimientos funcionales incluyendo el de todos los usuarios involucrados es decir: ServiceDesk, Administrador, Soporte nivel 1, Soporte Nivel 2 y el Usuario, concluimos con un total de 37 requerimientos funcionales.

Y requerimientos no funcionales con un total de 2.

*Por lo tanto*

$$Nr = 37 + 2$$

$$Nr = 39$$

donde

*Nr = Numero de requerimientos*

$$Nf = 37$$

$$Nnf = 2$$

Especificidad (Ausencia de Ambigüedad)

$$Q1 = \frac{Nui}{Nr}$$

$$Q1 = \frac{8}{39}$$

$$Q1 = 0.25129$$

Al ser este un dato cualitativo resulta difícil la idea de poder ser medible, ya que su factor de total de requerimiento puede cambiar de acuerdo a las necesidades del usuario, pero por primera instancia tenemos que:

El valor de Q1 debe de ser más próximo a uno y con esto será menor la ambigüedad de la especificación, al obtener un resultado de:

$$Q1 = 0.25129$$

Por lo tanto nuestro diseño es próximo a uno, por lo tanto nuestra ambigüedad es de un 25%, multiplicando por 100, lo cual nos dice que tenemos 25 % de ambigüedad, esto quiere decir que nuestros requisitos poseen un 25% total de interpretación final.

## Métricas del Modelo de Diseño

### Métricas de diseño arquitectónico

Propuesta por Card y Glass, plantea Métricas de Complejidad:

#### Complejidad Estructural.

$$S(i) = f_{out}^2(i)$$

donde

$f_{out}(i)$   
*expansion del módulo i (número de módulos inmediatamente subordinados al módulo i)*

Aplicado a nuestro diseño vemos que:

Módulos en el diseño.

- Log In a la aplicación
- Registro Incidencia
- Categorización Incidencia
- Priorización incidencia
- Resolver Incidencia
- Cerrar Incidencia
- Administración de usuarios
- Log out
- Reportes mensuales de atención
- Estatus de la incidencia.
- Módulo de ServiceDesk
- Módulo de Administrador
- Módulo de Soporte Nivel 1
- Módulo de Soporte Nivel 2
- Módulo de Solicitante

Módulos Subordinados

#### 4.1 Métricas Aplicadas al diseño del Sistema de Gestión de Incidencias Orientado a Objetos

- Módulo de ServiceDesk. Log In a la aplicación, Categorización Incidencia, Priorización Incidencia, Cerrar Incidencia, Log out de la aplicación.
- Módulo de Administrador. Log in a la aplicación, administración de usuarios, cerrar incidencia, estatus incidencia, reportes mensuales de atención.
- Módulo de Soporte Nivel 1. Log in a la aplicación, resolver incidencia, cerrar incidencia, log out de la aplicación.
- Módulo de Soporte Nivel 2. Log in a la aplicación, resolver incidencia, cerrar incidencia, log out de la aplicación.
- Módulo de Solicitante. Log in a la aplicación, registro incidencia, status incidencia, cerrar incidencia, log out incidencia.

MóduloServiceDesk

$$f_{out}(1) = 5$$

Módulo Administrador

$$f_{out}(2) = 5$$

Módulo Soporte Nivel 1

$$f_{out}(3) = 4$$

Módulo Soporte Nivel 2

$$f_{out}(4) = 4$$

Módulo Solicitante

$$f_{out}(5) = 5$$

Tomamos el Modelo de ServiceDesk para ejemplificar nuestra fórmula, por lo tanto tenemos:  $f_{out}^2(2) = 25$

*Aplicando la formula de expansion por modulo i tenemos que:*

$$S(i) = f_{out}^2(i)$$

*Por lo tanto*

$$S(1) = 25$$

Por lo tanto el rango de módulos independientemente subordinados para el Modulo de ServicioDesk es de 25.

**Complejidad de Datos.** Sobre la interfaz interna del modulo, proporciona una indicación de la complejidad en la interfaz interna de un modulo  $i$ , y se define como:

$$D(i) = \frac{v(i)}{[f_{out}(i) + 1]}$$

donde

$v(i)$  = numero de variables que entran o salen del modulo  $i$

- Módulo de Administrador.  
Variables que entran o salen
  - ID ticket incidencia: registrado, asignado, resuelto, rechazado, cierre

$$v(1) = 5$$

Por lo tanto, para el modulo 1 tenemos que:

$$v(i) = 5$$

$$f_{out}(1) = 5$$

Aplicando la fórmula vemos que la complejidad de los datos se define por:

$$D(i) = \frac{v(i)}{[f_{out}(i) + 1]}$$

$$D(i) = \frac{5}{6}$$

$$D(i) = 0.83$$

Siendo  $D(i)$ , un valor muy próximo a 1, por lo cual el valor de complejidad de datos es menor y por consiguiente más fácil de programar.

#### 4.1 Métricas Aplicadas al diseño del Sistema de Gestión de Incidencias Orientado a Objetos

**Complejidad Total del Sistema.** Se define como la suma de las complejidades estructural y de datos.

$$CS = \sum_i (S(i) + D(i))$$

Siguiendo con nuestro ejemplo de ServiceDesk con valor de:

$$S(i) = 25$$

Aplicamos la fórmula observando que:

$$CS = \sum_i (25 + 0.83) = 25.83$$

Ahora debemos de aplicar la formula de la complejidad relativa del Sistema

$$CRS = Promedio(S(i) + D(I))$$

Aplicado a nuestros resultados

$$CRS = Promedio 25.83$$

$$CRS = 12.915$$

El crecimiento de los valores de complejidad es proporcional al crecimiento de la complejidad arquitectónica del sistema.

Por lo tanto:

Si  $CRS > 26.5$ , aumenta la probabilidad de un mayor esfuerzo innecesario para la integración y pruebas, y si  $CRS < 26.5$ , disminuye la probabilidad de un esfuerzo innecesarios para integración de pruebas.

Con nuestro resultado de 12.925 podemos concluir que realizar el módulo de ServiceDeskes factible ya que no tiene riesgo de aumentar su complejidad de integración y pruebas.

Esta serie de pasos debe de realizarse para cada modulo a trabajar, que se desee integrar al sistema.

### Métrica del Caso de Uso

**Marchesi**, esta métrica es utilizada para definir y estimar la complejidad del sistema, está basada en una relación entre los casos de uso y los puntos de fusión con el número de casos de uso ( $N_{CU}$ ), el número de actores ( $N_a$ ) y el número de inclusiones y extensiones.

Es así que Marchesi propone la métrica de UC4 y su ecuación es:

$$UC4 = K_1UCI^2 + UC3 + K_2[smm(|C|) - smm(|D|)]$$

donde:

*$K_1$  y  $K_2$  son constantes y son calculados empíricamente*

*UCI representan el número de casos de uso en los requerimientos de especificación.*

*UC3 es el total de comunicacion es decir extensiones e inclusiones*

$|C|$  es una matriz con dimensión  $N_a \times N_{CU}$  tiene un valor de 1, al ser una matriz de identidad, donde  $N_a$ , es el número de actores del Caso de Uso y  $N_{CU}$ , el número de inclusiones y extensiones.

$|E|$  es una matriz con dimensión  $N_a \times N_{CU}$  representando las relaciones entre los casos de uso y la eliminación de redundancias.<sup>15</sup>

### Propuesta de Karner

---

<sup>15</sup>Esta prueba no fue ejemplificada debido a que es dato empírico

#### 4.1 Métricas Aplicadas al diseño del Sistema de Gestión de Incidencias Orientado a Objetos

Karner (RationalSoftware) define el concepto de puntos de caso de uso análogos para la función de puntos. Los puntos de caso de uso estimado por la relación de hombres y horas en el desarrollo del proyecto.

UCP describe a los Casos de Uso adjuntados y UCCP a los casos de uso en ajustar.

$$UCP = UUCP * TCF * EF$$

*UUCP. caso de uso sin ajustar calucla la suma ponderad del numero de actores y de caso de usos'*

La complejidad de los actores dependen de dos factores; el número de clases de análisis correspondiente a el caso de uso.

Se utiliza principalmente en la obtención de horas-hombre que se verán involucrados en la implementación de un aplicativo, a base datos ponderados, lo cual nos describe hacer la suma total de involucrados y dividirlos ente estos valores.

Describiremos el valor de cada caso de uso, en base a la siguiente tala dada por la métrica, en las cuales observamos 3 tablas.

En la primera se describe al tipo de actor involucrado o a tratar, así como su descripción dentro del sistema.

En la segunda tabla hablamos de las transacciones ponderadas a utilizar por este mismo actor que se dividen entre simple, promedio y complejo dependiendo del número de estas.

Por último en la tercera tabla hablamos de las clases basadas en factores ponderadas, que a su vez se dividen del mismo modo, en simple, promedio y complejo, dependiendo del total de clases.

#### *Factores de Ponderación de Actor*

Tipo de Actor	Descripción	Factor
Simple	Interface de Programa	1
Promedio	Interactivo o protocolo de interfaz	2
Compleja	Interfaz gráfica	3



**Tabla 14.** Factores de Ponderación de Actor

Basadas en transacciones ponderados

Tipo de caso de uso	Descripción	Factor
Simple	3 o menos transacciones	5
Promedio	4 a 7 transacciones	10
Complejo	Mas de 7 transacciones	15

**Tabla 15.** Factores de Ponderación de Actor basada en transacciones

Análisis de clases basadas en factores ponderados

Tipo de Caso de uso	Descripción	Factor
simple	Menos de 5 clases de análisis	5
Promedio	De 5 a 10 clases de análisis	10
compleja	Mas de 10 clases de análisis	15

**Tabla 16.** Análisis de clases en Factores de Ponderación de Actor

*TCF.* Representa el factor de complejidad incrementa la complejidad de los requerimientos no funcionales.

*EF* Representa el nivel de experiencia técnica del personal que trabaja en el proyecto

Ejemplificando nuestro diseño vemos que:

*Factores de Ponderación de Actor: Administrador*

Tipo de Actor	Descripción	Factor
Compleja	Interfaz grafica	3

**Tabla 17.** Factores de Ponderación de Administrador

Basadas en transacciones ponderados

Tipo de Caso de uso	Descripción	Factor
Promedio	4 a 7 transacciones	10

**Tabla 18.** Factores de Ponderación de Administrador, basada en transacciones

Análisis de clases basadas en factores ponderados

Tipo de Caso de uso	Descripción	Factor
Simple	Menos de 5 clases de análisis	5

#### 4.1 Métricas Aplicadas al diseño del Sistema de Gestión de Incidencias Orientado a Objetos

**Tabla 19.** Análisis de clases en Factores de Ponderación de Administrador

##### *Factores de Ponderación de Actor: Service Desk*

Tipo de Actor	Descripción	Factor
Compleja	Interfaz grafica	3

**Tabla 20.** Factores de Ponderación de ServiceDesk

Basadas en transacciones ponderados

Tipo de caso de uso	Descripción	Factor
Complejo	Mas de 7 transacciones	15

**Tabla 21.** Factores de Ponderación de ServiceDesk basada en transacciones

Análisis de clases basadas en factores ponderados

Tipo de Caso de uso	Descripción	Factor
Promedio	De 5 a 10 clases de análisis	10

**Tabla 22.** Análisis de clases en Factores de Ponderación de ServiceDesk

##### *Factores de Ponderación de Actor: Solicitante*

Tipo de Actor	Descripción	Factor
Compleja	Interfaz grafica	3

**Tabla 23.** Factores de Ponderación de Solicitante

Basadas en transacciones ponderados

Tipo de caso de uso	Descripción	Factor
Simple	3 o menos transacciones	5

**Tabla 24.** Factores de Ponderación de Solicitante, basada en transacciones

Análisis de clases basadas en factores ponderados

Tipo de Caso de uso	Descripción	Factor
simple	Menos de 5 clases de análisis	5

**Tabla 25.** Análisis de clases en Factores de Ponderación de Solicitante

*Factores de Ponderación de Actor: Soporte Nivel 1*

Tipo de Actor	Descripción	Factor
Compleja	Interfaz grafica	3

**Tabla 26.** Factores de Ponderación de Soporte Nivel 1

Basadas en transacciones ponderados

Tipo de caso de uso	Descripción	Factor
Simple	3 o menos transacciones	5

**Tabla 27.** Factores de Ponderación de Soporte Nivel 1 basada en transacciones

Análisis de clases basadas en factores ponderados

Tipo de Caso de uso	Descripción	Factor
simple	Menos de 5 clases de análisis	5

**Tabla 28.** Análisis de clases en Factores de Ponderación de Soporte Nivel 1

*Factores de Ponderación de Actor: Soporte Nivel 2*

Tipo de Actor	Descripción	Factor
Compleja	Interfaz gráfica	3

**Tabla 29.** Factores de Ponderación de: Soporte Nivel 2

Basadas en transacciones ponderados

Tipo de caso de uso	Descripción	Factor
Simple	3 o menos transacciones	5

#### 4.1 Métricas Aplicadas al diseño del Sistema de Gestión de Incidencias Orientado a Objetos

**Tabla 30.** Factores de Ponderación de Soporte Nivel 2 basada en transacciones

Análisis de clases basadas en factores ponderados

Tipo de Caso de uso	Descripción	Factor
simple	Menos de 5 clases de análisis	5

**Tabla 31.** Análisis de clases en Factores de Ponderación de Soporte Nivel 2

- **Primer paso.** Factor de peso de los actores sin ajustar (UAW), de los actores que tenemos calculamos únicamente para los actores de peso 3 es decir complejo.

Los cuales de acuerdo a nuestras tablas son todo nuestros actores: Administrador, Solicitante, ServiceDesk, Soporte Nivel 1 y Soporte, ya que la interacción que éstos tienen con la interfaz gráfica es de un 100%.

Ahora basamos nuestro ejemplo de esta métrica en el caso de uso de Administrador, en la tabla número 16, 17 y 18 vemos que sus valores son:

##### *Factores de Ponderación de Actor: Administrador*

Tipo de Actor	Descripción	Factor
Compleja	Interfaz gráfica	3

Basadas en transacciones ponderados

Tipo de Caso de uso	Descripción	Factor
Promedio	4 a 7 transacciones	10

Análisis de clases basadas en factores ponderados

Tipo de Caso de uso	Descripción	Factor
Simple	Menos de 5 clases de análisis	5

Por lo tanto:

$$UAW = 1 * 3 = 3$$

- **Segundo paso.** Factor de peso de los casos de uso sin ajustar (UUCW), en este caso vemos el caso de Administrador en donde obtenemos que el número de transacciones que éste tiene son 4 transacciones por lo que en la tabla de

transacciones obtenemos un valor de 10 para este número de transacciones por lo que:

$$UCW = 4 * 10 = 40$$

- **Tercer paso.** Finalmente los Caso de Uso sin ajustar resultan:

$$UUCP = UAW + UUCW = 3$$

Ahora calculamos los Casos de Uso Ajustados, una vez que tenemos los casos de uso sin ajustar debemos de ajustar este valor mediante la ecuación:

$$UCP = UUCP * TCF * EF$$

donde

$$UUCP = \text{puntos de casos de uso sin ajustar}$$

$$TCF = \text{Factor de complejidad tecnica}$$

$$EF = \text{Factor de ambiente}$$

- **Cuarto paso.** Factor de complejidad técnica., obtenido de la tercera tabla.

Por lo tanto

$$TCF = (0.6 + 0.01) * \sum (Peso_i * Valor asignado_i)$$

$$TCF = (0.6 + 0.01) * 5$$

$$TCF = (0.6 + 0.01) * 5$$

$$TCF = (0.61) * 3.05$$

#### 4.1 Métricas Aplicadas al diseño del Sistema de Gestión de Incidencias Orientado a Objetos

- **Quinto paso.** Factor de ambiente. Este factor se basa principalmente en las habilidades técnicas de los involucrados es decir, en el factor humano que trabaja, el cual describe que:

Factores	VALOR ASIGNADO		
	0	3	5
E1 al E4	Sin Experiencia.	Experiencia media.	Amplia experiencia (experto).
E5	Sin motivación para el proyecto.	Motivación media.	Alta motivación.
E6	Requerimientos extremadamente inestables.	Estabilidad media.	Requerimientos estables sin posibilidad de cambios,
E7	Todo el personal es tiempo completo.	Personal mitad de tiempo completo, mitad de medio tiempo.	Todo el personal es medio tiempo.
E8	Lenguaje de programación fácil de usar.	Lenguaje de programación medio.	Lenguaje de programación exageradamente difícil.

**Tabla 32.** Factores de ambiente de Propuesta de Karner

La fórmula para calcular el factor de ambiente nos dice que:

$$EF = (1.4 - 0.03) * \sum (Peso_i * Valor\ asignado_i)$$

Supongamos que contamos con personal con motivación media, por lo tanto:

$$EF = (1.4 - 0.03) * 3$$

$$EF = (1.37) * 3$$

$$EF = 4.11$$

- **Sexto paso.** Finalmente los Puntos de Caso de Uso, para un caso de uso complejo con valor de complejidad técnica 5 y factor externo son:

$$UCP = UUCP * TCF * EF$$

De donde obtuvimos los valores:

$$UUCP = 3$$

$$TCF = 3.05$$

$$EF = 4.11$$

$$UCP = 3 + 30.5 + 4.11$$

$$UCP = 10.16$$

### Medidas para el Diagrama de Estados

Aplica para diagramas con OMT que también se pueden aplicar a diagramas de UML.

Atributos Medibles	Medidas
Complejidad del modelo	Número de escenarios
Complejidad de las transacciones de los estados	Número de estados Número de transacciones

**Tabla 33.** Medidas del diagrama de estados

### Medida de CRUZ-LEMUS ET AL (2002)

Mediante una tabla, explicaremos esta medida aplicada a nuestro diagrama.

Nombre	Siglas	Descripción
Nivel de anidación de los estados compuestos	NLCS	El número máximo de estados compuestos anidados en otros estados compuestos.
Numero de actividades	NA	El número total de actividades (do/Activity)
Número de estados compuestos	NCS	El número total de estados compuestos
Número de Transacciones complejas	NCT	Una transacción compleja a cumplido una serie de condiciones de guarda, eventos o acciones mientras que una transacción simple no.
	NCTG1	Número de transacciones que han cumplido una condición de guarda y un evento.
	NCTG2	Número de transiciones que han cumplido un evento y una lista de acciones.

#### 4.1 Métricas Aplicadas al diseño del Sistema de Gestión de Incidencias Orientado a Objetos

	NCTG3	Número de transacciones que han cumplido un evento, una condición de guarda y una lista de acciones.
Número de evento	NE	El número total de eventos y de cualquier tipo.
Número de acciones de entrada	NEntryA	El número total de acciones de entrada, es decir, las acciones que se realizan cada vez que se alcanza un determinado estado.
Número de acciones de salida	NExitA	El número total de acciones de salida, es decir, las acciones que se realizan cada vez que se deja un determinado estado.
Número de guardas	NG	El número de condiciones de guarda.
Número de acciones indirectas	NIA	Número de acciones que deben realizarse asociadas a una transacción.
Número de estados simples	NSS	El número total de estados, considerando también cada uno de los estados imples (sub.estados) que componen los estados compuestos.
Número de transacciones	NT	El número total de transiciones, considerando las transiciones (con diferentes estados en el origen y el destino) las transacciones iniciales y finales, transiciones sobre el mismo (siendo el origen y el destino el mismo estado) y transiciones internas (transiciones dentro de un estado que responden a un evento, pero sin salir del estado).

**Tabla 34.** Medidas de Cruz Lemus

Tabla ejemplificando nuestro diagrama

#### Medida de CRUZ-LEMUS ET AL (2002)

Nombre	Siglas	Totales
Nivel de anidación de los estados compuestos	NLCS	4
Número de actividades	NA	7
Número de estados compuestos	NCS	2
Número de Transacciones complejas	NCT	1
	NCTG1	4
	NCTG2	5
	NCTG3	4
Número de evento	NE	7
Número de acciones de entrada	NEntryA	6
Número de acciones de salida	NExitA	5
Número de guardas	NG	3
Número de acciones indirectas	NIA	3
Número de estados simples	NSS	7



Número de transacciones	NT	4
-------------------------	----	---

**Tabla 35.** Medidas de Cruz Lemus aplicadas a SGIOO

Con esta métrica podemos obtener cual será la complejidad de nuestro sistema, en relación al número de estados que se empleen en nuestro diagrama de estados, que describen el número de eventos, entradas y salidas en totalidad de nuestro sistema, a mayor número de estos será mayor el número.

### Medidas para Diagrama de Clases

Las propuestas por Chidamber y Kemerer (1994), Lorenz y Kidd (1994) Brito e Abreu y Carapucca (1994) entre otros, las más populares son las de Chidamber y Kemerer y se aplican a nivel conceptual como:

WMC (Weighted Methods per Classo Métodos ponderados por clase), se define:

$$WMC = \sum_{i=1}^n C_i$$

donde:

$C_1$  representa la complejidad de los métodos de una clase con métodos  $M_1 \dots M_n$

Si todos los métodos se consideran de complejidad 1, por que a nivel conceptual no se tienen el código de los métodos entonces  $WMC = n$ .

*DIT (Depth of Inheritance Tree o Profundidad del Arbol de Herencia)*. Se refiere a la distancia que existe entre dicha clase y la raíz del árbol de herencia, en caso de herencia múltiple se considera la distancia más larga.

*NOC (Number of Children o Numero de Hijos)*. Es el número de subclases que cuelgan de una clase dada de una Jerarquía de herencia.

Ejemplificando nuestro diseño, vemos que consideraremos a nuestro diagrama de complejidad 1 ya que no contamos con el código de los métodos por lo tanto:

#### 4.1 Métricas Aplicadas al diseño del Sistema de Gestión de Incidencias Orientado a Objetos

$$WMC = 1$$

$$DIT = 3$$

$$NOC = 2$$

DIT, mide el máximo nivel en la jerarquía de herencia, es la cuenta directa de los niveles en la jerarquía de herencia, en esta métrica DIT, queda definido como el la medida de complejidad de una clase, de este modo cuanto mas profunda se encuentra una clase en la jerarquía, mayor será la complejidad del diseño y el potencial de reusó.

Esto debido a que cuanto mas profunda se encuentra una clase en la jerarquía mayor será la probabilidad de heredar un mayor numero de métodos, y con esto aumentar la complejidad de testear y reusar.

Los bajos niveles de herencia pueden indicar que el código es un estilo funcional sin aprovechar.

La métrica de Lorens y Kidd sugiere un umbral de 6 niveles como indicador de un abuso en la herencia, por lo tanto al ser nuestro sistema un nivel de 3, podemos concluir que es factible para ser desarrollado, testeado y reusado

El detalle de esta métrica radica en la correcta definición de DIR.

NOC, es el número de subclases subordinadas a una clase en la jerarquía , es decir el numero de subclases que pertenecen a una clase, es un indicador de nivel de reusó, mide la posibilidad de haber creado abstracciones erróneas y es un indicador a nivel de test.

Su medida prácticamente nos indica que a mayor número de hijos representa una mayor reutilización de código por lo tanto éste solo nos indicaría:

- Mayor probabilidad de usar incorrectamente la herencia creando abstracciones erróneas.
- Mayor dificultad para modificar una clase ya que afecta su herencia.
- Se requiere mayor número de recursos para probar.

Cuando nuestro sistema tenga demasiada dependencia será mejor dividir las funcionalidades.

WMC, mide la complejidad de una clase, si todos los métodos son considerados igualmente de complejos entonces WMC, es el número de métodos definidos en una clase.

Se sugiere un umbral de 20 a 40, dependiendo si las clases son o no de interface de usuario, esta métrica debería de ser considerada como una medida de tamaño de una clase

En 1991 **Fenton** propone medidas de morfología simples basadas en la estructura de módulos jerárquicos del sistema, que permiten comparar diferentes arquitecturas mediante un conjunto de dimensiones directas.

De nuestro Diagrama de Clases descrito en la página 39 podemos ejemplificar nuestra fórmula con los siguientes datos:

$$\text{Tamaño} = n + a \text{ (numero de nodos + numeros de aristas)}$$

*Profundidad*

*Anchura*

$$\text{Relacion arco - nodo, } r = \frac{a}{n}$$

Donde

*Profundidad = Camino mas largo desde el nodo raiz a un nodo hoja.*

*Amplitud = numero maximo de nodos de cualquier nivel de la arquitectura*

*Relación arco-nodo = Mide la densidad de conectividad de la arquitectura y proporciona una indicación sencilla de acoplamiento de la arquitectura.*

*Ejemplificando nuestro diseño vemos:*

$$\text{Tamaño} = n + a$$

$$n = 6$$

$$a = 8$$

#### 4.1 Métricas Aplicadas al diseño del Sistema de Gestión de Incidencias Orientado a Objetos

Por lo tanto

$$\text{Tamaño} = 14$$

$$\text{Profundidad} = 5$$

$$\text{Anchura} = 6$$

$$\text{Relacion arco - nodo}, r = \frac{a}{n}$$

Por lo tanto

$$\text{Relacion arco - nodo}, r = \frac{8}{6}$$

$$\text{Relacion arco - nodo}, r = 0.7$$

# Conclusiones

Cualquier sistema que queramos desarrollar, es necesario que se origine a partir de la especificación y diseño, es por eso la importancia de mostrar el uso de las métricas de calidad en esta primera fase del ciclo de vida de un sistema, para la mejora de la calidad, además del uso de un método o modelo, que no sólo oriente a los analistas de sistemas sino a todo un equipo de trabajo que se encuentra detrás de este.

No debemos dejar de lado el hecho de que cambiar un método o proceso dentro de una empresa no es cosa fácil, pues éste también implica generar costos y tiempos altos o en caso contrario adaptar las métricas a un proceso ya existente tampoco será sencillo, a pesar de que este tema no es nuevo, sigue siendo un tema desconocido en el mundo empresarial actual, pues no consideran importante el hecho de determinar por este medio la calidad de sus productos.

Con este trabajo, demostramos que en un principio, se puede reducir los costos y tiempos que implica la construcción de un sistema, al trabajar con métricas que nos orientan a estos resultados, con los costos y recursos necesarios, y en segundo lugar identificar cual es la importancia de implementar un método o proceso que ayude al uso de métricas y a su vez a la mejora de calidad.

El resultado de este trabajo de investigación sugiere, que con una buena práctica y uso de métodos en relación a la implementación de métricas de calidad de software, se orienta a los involucrados de un proyecto, a definir mejor sus alcances, al conocer cuáles podrían ser sus posibles tiempos, y costos invertidos, así como el número de personas necesarias para su implementación, en algunos casos incluso es posible definir cuales serían los costos para otras etapas del ciclo de vida del software, no sólo en el análisis y diseño, sino también en desarrollo, pruebas y mantenimiento.

Un trabajo a futuro de esta investigación podría ser el desarrollo de una herramienta que calcule las métricas de calidad del diseño de forma automatizada y con esto quitarse de encima el trabajo que podría implicar el desarrollar las fórmulas, pues si bien hoy en día si existen, herramientas de métricas de calidad que determinen el tamaño de código, o el número de casos de prueba posibles, no hay alguna que dé como resultado la especificación y diseño.

Como alumno de la carrera de Matemáticas Aplicadas y Computación, he notado que el área de calidad de software cubre de manera significativa el perfil de un egresado, ya que se necesita contar no sólo con conocimientos de sistemas, sino además del análisis matemático que puede agilizar nuestra percepción del negocio y ayudarnos con esto, a desempeñar de manera satisfactoria la facilidad de poder analizar los riesgos que puede

implicar los incidentes encontrados en un sistema, además de que existen diversas pruebas en las cuales son necesarios análisis estadísticos previos y posteriores a los desarrollos que ayudan al desempeño de las BD, estos análisis son utilizados en las Métricas de Calidad en Software, sin embargo estas herramientas de análisis y medición no son conocidas y empleadas profesionalmente hablando, pero sin lugar a duda crecerán gracias a las necesidades globalizadas de las empresas para una mejor competencia, pues a mayor calidad de sistemas, mayores serán las ganancias y será de buen uso el saber implementar las medidas a fin de ayudar a las empresas a lograr sus objetivos.

El haber trabajado e investigado sobre este tema, amplió mis conocimientos hacia una nueva área que poco a poco crecerá y con esto estaré un paso adelante.

Sn embargo considero que el aprendizaje de ésta área es muy pobre en la educación a nivel licenciatura o ingeniería de sistemas relacionados con el área específica de Calidad de Software, y son pocas las carreras que especializan a gente dentro de este rubro, existen ingenieros en sistemas, informáticos, y demás especialistas, recomendaría a la jefatura de MAC, añadir una asignatura al programa de MAC con el nombre de Calidad de Software donde el objetivo sea incitar a los alumnos por una nueva opción de desarrollo profesional.

La calidad del software ha crecido con el tiempo, extendiendo hoy en día incluso certificaciones, sin embargo creo que mucha gente se dedica a esto, porque ya no le agrado programar o arreglar computadoras, cuando debería de elegir esta área desde antes de su desarrollo profesional, al ser elegida en un desarrollo educativo podríamos garantizar un mejor profesionista, y ampliar de esta manera su mundo de posibilidades profesionales.

# Bibliografía

1. Cuellar, L. R. (Mayo-Julio de 2009). La cultura de la calidad tomemos el reto. *Software Gúru No. 24*, pág. 10.
2. Gutiérrez, E. (Febrero-Marzo de 2009). Implementación de modelos de calidad diagnóstico de las empresas desarrolladoras de software en México. *Software Gúru No. 23*, págs. 44-46.
3. Marcela Genero, M. P. (2005). *Metrics For Software Conceptual Models*. España: Imperial College Press.
4. Mark, H. B. (2010). *Manual de Aplicaciones: Guía para manejar con maestría el ciclo de vida de las aplicaciones modernas*. Hewlett-packard.
5. Pressman, R. S. (2002). *Ingeniería del Software un enfoque práctico*. Madrid España: McGraw Hill 4ta Edición.
6. Sicilia, M. A. (8 de Enero de 2009). *Connexions*. Recuperado el 14 de Diciembre de 2011, de Métricas de la Calidad del Diseño Orientado a Objetos del Software: <http://cnx.org/content/m17463/latest/>
7. Universidad Politecnica de Madrid ( 2005-2006). *Presentacion del curso 2005-2006*. Recuperado el 20 de Enero de 2012, de Proyectos Informaticos <http://www.oei.eui.upm.es/Asignaturas/PInformaticos/ficheros/temario/PROYINF-2.pdf>
8. Booch Grady, R. J., J. I. The Unified Modeling Language User Guide, Herman Steffen (octubre de 2008), *Las pruebas en el desarrollo del Software* , Seminario FUNTEC Recuperado el 13 de Marzo de 2012, de Fundacion para el desarrollo de las nuevas tecnologías,Buenos Aires, diapositivas 87-91,127-128, [www.funtec.org.ar](http://www.funtec.org.ar)
9. Olsina Luis (2003), *Metricas e Indicadores: Dos conceptos claves para Medicion y Evaluacion*, GIDIS, Facultad de Ingenieria UNLPam, La Pampa Argentina, Recuperado el 13 febrero 2012 <http://gidis.ing.unlpam.edu.ar>
10. Osiatis The IT service experts, Curso ITIL:Gestion de Servicio TI Recuperado el 20 de Febrero 2012 de TIL-Gestión de Servicios TI Osiatis, Gestion de Incidentes [http://itil.osiatis.es/Curso\\_ITIL/](http://itil.osiatis.es/Curso_ITIL/)
11. Gracia Romero Claudia Ivette ,Tesis: El Modelo de Capacidad de Madurez y su Aplicación en Empresas Mexicanas, Recuperado el 11 de Enero del 20012 de Colección de Tesis Digitales Universidad de las Americas Puebal, capitulo 2 Modelos y Estandares de Calidad de Software. [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/garcia\\_r\\_ci/capitulo2.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/garcia_r_ci/capitulo2.pdf)
12. Qualitrian Express, Aseguramiento de la calidad del Software, consultoria TI Blog Qualitrian , Recuperado el 12 de Abril del 2012 de Pagina Oficial Qualitrian Express,

- Procesos de Software, Aseguramiento de la Calidad del Software.  
<http://www.qualitrain.com.mx/Aseguramiento-de-la-Calidad-de-Software>
13. Heidi González Doria, Las Métricas de Software y su Uso en la Región Recuperado el 7 de julio del 2012 de UPLAP Bibliotecas.  
[http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/gonzalez\\_d\\_h/capitulo6.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/gonzalez_d_h/capitulo6.pdf)
  14. Dra. María de los Ángeles Sumano López Pagina Personal Universidad Veracruzana, Recuperado el 26 de Abril del 2012 <http://www.uv.mx/personal/asumano/files/2012/08/MetricasTecnicas.pdf>
  15. Facultad de Ciencias Químicas e Ingeniería Campus Tijuana Recuperado el 717 de junio del 2012. <http://fcqi.tij.uabc.mx/usuarios/luisgmo/data/5.5%20gestion%202007.pdf>
  16. Facultad de Ingeniería Universidad de Buenos Aires ,Maestria en Ingeniería en Calidad , “Estudio Comparativo de los modelos y Estandares de Calidad de Software” Fernanda Scalone Recuperado el 23 de Febrero 2012 <http://laboratorios.fi.uba.ar/lsi/scalone-tesis-maestria-ingenieria-en-calidad.pdf>



# ANEXO A

## Glosario de términos

Ahora definamos algunos términos que serán utilizados en nuestro diseño y con esto tener más claro el proceso que definiremos más adelante.

- **Rol.** Conjunto de actividades y responsabilidades asignadas a una persona o grupo una persona o grupo puede desempeñar más de una actividad.
- **KPI.** Key Performance Indicators, Indicadores de clave de rendimiento, ayudan a las empresas a entender lo bien que se esta realizando el trabajo, en relación a sus metas y objetivos estratégicos.
- **Acoplamiento.** Grado de interdependencia entre las unidades de software (módulos, funciones, subrutinas, bibliotecas, etc.) de un sistema informático. El acoplamiento da la idea de lo dependiente que son las unidades de software entre sí, es decir, el grado en que una unidad puede funcionar sin recurrir a otras
- **Cohesión.** Forma en que se agrupan las unidades de software (módulos, subrutinas...) en una unidad mayor.
- **Escalado:** Cuando un Centro de Servicios no es capaz de resolver n incidente se debe recurrir a un especialista o algún superior que pueda tomar decisiones que se escapan de su responsabilidad.
- **Incidencia.** *“Cualquier evento que no forma parte de la operación estándar de un servicio y que causa, o puede causar una interrupción o una reducción en la calidad del mismo”.*
- **Indicador.** Magnitud utilizada para medir o comparar los resultados efectivamente obtenidos, en la ejecución de un proyecto, programa o actividad. Resultado cuantitativo de comparar dos variables
- **Medición.** Es el acto por el cual números o símbolos son asignados a atributos de entidades del mundo real de modo de poder describirlos acorde a reglas claramente definidas.
- **Medida.** Proporciona una indicación, cuantitativa de la cantidad, dimensiones o tamaño de algunos atributos de un producto.
- **Métrica.** Es una medida del grado en que un sistema, componente o proceso posee un atributo dado.
- **Modulo.** Es un software que agrupa un conjunto de subprogramas y estructuras de datos. Los módulos son unidades que pueden ser compiladas por separado y los hace reusables y permite que múltiples programadores trabajen en diferentes módulos en forma simultánea, produciendo ahorro en los tiempos de desarrollo.
- **SLA.** Acuerdo de Nivel de Servicio es un contrato escrito entre un proveedor de servicio y su cliente con el objetivo de fijar el nivel acordado, ayuda a ambas

partes a llegar a un conceso en términos de nivel de calidad del servicio en aspectos tales como tiempo de respuesta, disponibilidad horaria, documentación, personal asignado etc.

- **Frontera de la aplicación.** Define lo que es externo de la aplicación, actúa como “membrana” a través de la cual las transacciones entran o salen de la aplicación, ayuda a identificar los datos mantenidos y referenciados, depende del punto de vista del usuario y no de consideraciones técnicas y/o implementación.
- **SPICE.** Software Process Improvement and Assurance Standards Capability Determination.
- **OMT.** Object Modeling Technique, es una metodología de análisis y diseño orientada a objetos, es de carácter abierto, que le permite ser de dominio publico y en consecuencia sobrevivir con enorme vitalidad.
- **ERS.** Especificación de Requisito de Software.
- **Especificidad.** Si todo requisito posee una sola interpretación.
- **Compleción.** Una ERS es completa si todo lo que se supone que el software debe hacer esta incluido, todas las posibles respuestas a todas las posibles entradas y en todas las situaciones posibles.
- **Corrección.** Todo requisito de la ERS contribuye a satisfacer una necesidad real.
- **Comprensión.** Todo tipo de lectores entienden la ERS.
- **Verificación.** Si para cada requisito expresado en la ERS existe un procedimiento de prueba finito y no costoso para demostrar que el futuro sistema lo satisface.
- **Consistencia.** No existen subconjuntos de requisitos contradictorios, ninguno de los requisitos esta en contradicción con lo expresado en los documentos de nivel superior.
- **Logro.** Dados los actuales recursos, la ERS es implementarle.
- **Concisión.** La ERS debe ser lo más breve posible.
- **Trazabilidad.** Si esta claro el origen de cada requisito.
- **Modificación.** Los cambios son fáciles de introducir.
- **Exactitud.** Una ERS es precisa si se hace uso de valores numéricos para precisar las características del sistema.
- **Reutilización.** Si ciertas secciones de la ERS se pueden reutilizar.

## ANEXO B

### DIAGRAMAS DE CASOS DE USO

#### CASO DE USO ADMINISTRADOR

##### *CASO de USO. Manejo Solicitantes*

	<b>Manejo Solicitantes</b>	
<b>Objetivos asociados</b>	Se debe de tener un control sobre los datos del registro, modificación y eliminación de Solicitantes.	
<b>Requisitos asociados</b>	Datos de Solicitantes. La BD del Sistema debe contener un catalogo de los Grupos Solicitantes, creados de acuerdo a datos proporcionados por el usuario, es como se crearon.	
<b>Descripción</b>	Administrador debe de controlar los datos del Solicitante.	
<b>Actor</b>	Administrador	
<b>Precondición</b>	Administrador tiene privilegios en la BD para realizar, selects, updates, deletes. Deben de crearse datos del Solicitante, para ser manipulados.	
<b>Secuencia</b>  <b>Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Administrador ingresa al Sistema.
	2	El sistema muestra los Registros que se han levantado y generado Roles de Solicitante, los cuales contienen los siguientes datos:  Nombre del Solicitante <ul style="list-style-type: none"><li>• Área Solicitante</li><li>• Puesto del Solicitante</li><li>• Ubicación Física</li><li>• Teléfono</li><li>• E-mail</li><li>• Jefe Directo</li><li>• Gerencia a la que pertenece</li><li>• Aplicación que maneja</li></ul>
	3	Administrador asigna los Grupos de Solicitante de acuerdo a sus datos.
	4	Administrador controla el número de Solicitantes Registrados. (Flujos Alternos 4a y 4b)

	7	Administrador cierra la sesión .
<b>Postcondición</b>		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	4a	Si el Solicitante Modifica los datos de su registro, el Administrado debe de controlar la modificación sin que se afecte la BD.  En caso de Modificaciones el Sistema Notifica al Administrador.
	4b	Si el Solicitante Da de Baja su registro, el Administrador debe de controlar la baja sin que se afecte la BD.  En caso de Bajas el Sistema Notifica al Administrador.
<b>Rendimiento</b>	<b>Paso</b>	<b>Cota de tiempo</b>
	4	5 segundos
<b>Frecuencia esperada</b>	5 veces/día	
<b>Estabilidad</b>	Alta	
<b>Comentarios</b>	La frecuencia podría ser mayor en el primer mes del sistema se espera que baje, con la BDK.	

#### **CASO de USO. Manejo Soporte**

	<b>Manejo Soporte</b>
<b>Objetivos asociados</b>	Se debe de tener un control sobre los datos del registro, modificación y eliminación de Soporte
<b>Requisitos asociados</b>	Datos de Soporte.La BD del Sistema debe contener un catalogo de los Grupos Soporte, creados de acuerdo a datos proporcionados por el usuario, es como se crearon.
<b>Descripción</b>	Administrador debe de controlar los datos del Soporte.
<b>Actor</b>	Administrador
<b>Precondición</b>	Administrador tiene privilegios en la BD para realizar, selects, updates, deletes.  Deben de crearse datos de Soporte, para ser manipulados.

<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>
<b>Normal</b>	1	Administrador ingresa al Sistema.
	2	El sistema muestra los Registros que se han levantado y generado el Rol de Soporte, los cuales contienen los siguientes datos: <ul style="list-style-type: none"> <li>• Nombre</li> <li>• Área</li> <li>• Puesto</li> <li>• Ubicación Física</li> <li>• Teléfono</li> <li>• E-mail</li> <li>• Jefe Directo</li> <li>• Gerencia a la que pertenece</li> <li>• Aplicación que maneja</li> </ul>
	3	Administrador asigna los Grupos de Solicitante correspondientes de acuerdo a sus datos.
	4	Administrador controla el número de Registros. (Flujos Alternos 4a y 4b)
	7	Administrador cierra la sesión.
	<b>Postcondición</b>	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	4a	Si el Solicitante Modifica los datos de su registro, el Administrado debe de controlar la modificación sin que se afecte la BD.  En caso de Modificaciones el Sistema Notifica al Administrador.
4b	Si el Solicitante Da de Baja su registro, el Administrador debe de controlar la baja sin que se afecte la BD.  En caso de Bajas el Sistema Notifica al Administrador.	
<b>Rendimiento</b>	<b>Paso</b>	<b>Cota de tiempo</b>
	4	5 segundos
<b>Frecuencia esperada</b>	5 veces/día	
<b>Estabilidad</b>	Alta	

<b>Comentarios</b>	La frecuencia podría ser mayor en el primer mes del sistema se espera que baje, con la BDK.
--------------------	---

**CASO de USO. Manejo ServiceDesk**

		<b>Manejo ServiceDesk</b>
<b>Objetivos asociados</b>	Se debe de tener un control sobre los datos del registro, modificación y eliminación de ServiceDesk	
<b>Requisitos asociados</b>	Datos de ServiceDesk. La BD del Sistema debe contener un catalogo de los Grupos ServiceDesk, creados de acuerdo a datos proporcionados por el usuario, es como se crearon.	
<b>Descripción</b>	Administrador debe de controlar los datos del ServiceDesk.	
<b>Actor</b>	Administrador	
<b>Precondición</b>	Administrador tiene privilegios en la BD para realizar, selects, updates, deletes.  Deben de crearse datos de ServiceDesk, para ser manipulados.	
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>
<b>Normal</b>	1	Administrador ingresa al Sistema.
	2	El sistema muestra los Registros que se han levantado y generado el Rol de ServiceDesk, los cuales contienen los siguientes datos: <ul style="list-style-type: none"> <li>• Nombre</li> <li>• Área</li> <li>• Puesto</li> <li>• Ubicación Física</li> <li>• Teléfono</li> <li>• E-mail</li> <li>• Jefe Directo</li> <li>• Gerencia a la que pertenece</li> <li>• Aplicación que maneja</li> </ul>
	3	Administrador asigna el Grupo de ServiceDesk correspondiente de acuerdo a sus datos.
	4	Administrador controla el número de Registros. (Flujos Alternos 4a y 4b)

	7	Administrador cierra la sesión.
<b>Postcondición</b>		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	4a	Si el Solicitante Modifica los datos de su registro, el Administrador debe de controlar la modificación sin que se afecte la BD.  En caso de Modificaciones el Sistema Notifica al Administrador.
	4b	Si el Solicitante Dar de Baja su registro, el Administrador debe de controlar la baja sin que se afecte la BD.  En caso de Bajas el Sistema Notifica al Administrador.
<b>Rendimiento</b>	<b>Paso</b>	<b>Cota de tiempo</b>
	4	5 segundos
<b>Frecuencia esperada</b>	5 veces/día	
<b>Estabilidad</b>	Alta	
<b>Comentarios</b>	La frecuencia podría ser mayor en el primer mes del sistema se espera que baje, con la BDK.	

### ***CASO de USO. Calculo Solicitudes Incidencias Registradas***

	<b>Calculo Solicitudes Incidencias Resgistradas</b>	
<b>Objetivos asociados</b>	Calcular Índices de acuerdo a la Solución y Cierres de Tickets de Incidencia.	
<b>Requisitos asociados</b>	KPI	
<b>Descripción</b>	Administrador debe de controlar los datos del Solicitudes de Incidencias Registradas.	
<b>Actor</b>	Administrador	
<b>Precondición</b>	Administrador tiene privilegios en la BD para realizar, selects, updates, deletes. Deben de crearse datos de ServiceDesk, para ser manipulados.	
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>

<b>Normal</b>	1	Administrador ingresa al Sistema.
	2	El sistema muestra los Registros que se han levantado y generado el Rol de ServiceDesk, los cuales contienen los siguientes datos: <ul style="list-style-type: none"> <li>• Nombre</li> <li>• Área</li> <li>• Puesto</li> <li>• Ubicación Física</li> <li>• Teléfono</li> <li>• E-mail</li> <li>• Jefe Directo</li> <li>• Gerencia a la que pertenece</li> <li>• Aplicación que maneja</li> </ul>
	3	Administrador asigna el Grupo de ServiceDesk correspondiente de acuerdo a sus datos.
	4	Administrador controla el número de Registros. (Flujos Alternos 4a y 4b)
	7	Administrador cierra la sesión.
<b>Postcondición</b>		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	4a	Si el Solicitante Modifica los datos de su registro, el Administrador debe de controlar la modificación sin que se afecte la BD.  En caso de Modificaciones el Sistema Notifica al Administrador.
	4b	Si el Solicitante Dar de Baja su registro, el Administrador debe de controlar la baja sin que se afecte la BD.  En caso de Bajas el Sistema Notifica al Administrador.
<b>Rendimiento</b>	<b>Paso</b>	<b>Cota de tiempo</b>
	4	5 segundos
<b>Frecuencia esperada</b>	5 veces/día	
<b>Estabilidad</b>	Alta	
<b>Comentarios</b>	La frecuencia podría ser mayor en el primer mes del sistema se espera que baje, con la BDK.	



## CASO DE USO SOLICITANTE

### CASO de USO. Registrar Incidencia

	<b>Registro Incidencia</b>	
<b>Objetivos asociados</b>	Generar Ticket de Incidencia.	
<b>Requisitos asociados</b>	<p>Información sobre solicitantes</p> <p>Validación Datos Incidencia</p> <p>El Sistema muestra las opciones para el Rol Solicitante.</p>	
<b>Descripción</b>	El solicitante ingresa a la aplicación para realizar un alta de incidencia.	
<b>Actor</b>	Solicitante	
<b>Precondición</b>	El solicitante se encuentra registrado en la BD con el rol de Solicitante y asignado a un grupo de acuerdo a su área.	
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>
<b>Normal</b>	1	El solicitante ingresa al sistema, da clic en "Incidencia".
	2	El solicitante ingresa al sistema, da clic en "Registrar Incidencia".
	3	<p>El sistema muestra datos por default:</p> <ul style="list-style-type: none"> <li>• Nombre del Solicitante.</li> <li>• Grupo al que pertenece.</li> <li>• Fecha de Alta Incidencia.</li> </ul>

	4	<p>El sistema solicita los siguientes datos, en un formulario:</p> <ul style="list-style-type: none"> <li>• Nombre de la Incidencia.</li> <li>• Tipo de Incidencia <ul style="list-style-type: none"> <li>○ Funcional</li> <li>○ BD</li> <li>○ Grafica/Diseño</li> </ul> </li> <li>• Código de la Aplicación (aplicación en la que se detecto la incidencia).</li> <li>• Descripción de la Incidencia.</li> <li>• Nivel de Gravedad <ul style="list-style-type: none"> <li>○ Alta</li> <li>○ Media</li> <li>○ Baja</li> </ul> </li> <li>• Mail <ul style="list-style-type: none"> <li>○ Solicitante</li> <li>○ Involucrados</li> </ul> </li> <li>• Agregar Documentación. (evidencias)</li> <li>• Agregar Observaciones.</li> <li>• Nuevo Proyecto (Flujo Excepción 4a)</li> </ul>
	5	El solicitante da clic en Guardar.
	6	El solicitante da clic en Enviar.
	7	La aplicación genera el Ticket de incidencia y lo envía por mail a los roles con privilegios. (Flujo Excepción 7a)
<b>Postcondición</b>		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	4a	Si se detecto la incidencia en un desarrollo se indica como nuevo proyecto, ya que este aun no se encuentra en el listado de códigos de proyecto.
	7a	Se envía el mail solo a las personas seleccionadas en la lista y a las personas con roles con privilegios.
<b>Rendimiento</b>	<b>Paso</b>	<b>Cota de tiempo</b>
	5	5 segundos
<b>Frecuencia esperada</b>	5 veces/día	
<b>Estabilidad</b>	Alta	
<b>Comentarios</b>	La frecuencia podría ser mayor en el primer mes del sistema se espera que baje, con la BDK.	

**CASO de USO. Validación Solución Ticket Incidencia**

	<b>Valida Solucion Ticket Incidencia</b>	
<b>Objetivos asociados</b>	Cerrar Ticket de Incidencia.	
<b>Requisitos asociados</b>	Solución Ticket Incidencia	
<b>Descripción</b>	El solicitante ingresa a la aplicación para Cerrar el Ticket de Incidencia.	
<b>Actor</b>	Solicitante	
<b>Precondición</b>	El solicitante recibe un mail notificando que la incidencia con numero "XXXX" ha sido solucionada, y se encuentra en espera de su Vo.Bo.	
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>
<b>Normal</b>	1	El solicitante ingresa al sistema, da clic en "Incidencia".
	2	El solicitante da clic en Incidencias Resueltas
	3	El Solicitante ingresa el número del Ticket de su Incidencia.
	4	<p>El sistema muestra datos del Registro de Incidencia:</p> <ul style="list-style-type: none"> <li>• Nombre del Solicitante.</li> <li>• Grupo al que pertenece.</li> <li>• Nombre de la Incidencia.</li> <li>• Tipo de Incidencia <ul style="list-style-type: none"> <li>○ Funcional</li> <li>○ BD</li> <li>○ Grafica/Diseño</li> </ul> </li> <li>• Código de la Aplicación (aplicación en la que se detecto la incidencia).</li> <li>• Descripción de la Incidencia.</li> <li>• Nivel de Gravedad <ul style="list-style-type: none"> <li>○ Alta</li> <li>○ Media</li> <li>○ Baja</li> </ul> </li> <li>• Mail <ul style="list-style-type: none"> <li>○ Solicitante</li> <li>○ Involucrados</li> </ul> </li> <li>• Agregar Documentación. (evidencias)</li> <li>• Agregar Observaciones.</li> <li>• Nuevo Proyecto (Flujo Excepción 2a)</li> </ul>

	5	El solicitante da clic a la pestaña Solución del Ticket de Incidencia.
	4	La aplicación muestra la Solución a la Incidencia.
	5	El Solicitante ingresa fecha fin de Validación de la Solución (Flujo Excepción 5a)
	6	El Solicitante envía su Vo.Bo. en comentarios. (Flujo Externo 6a)
	7	El Solicitante Cierra la Incidencia.
<b>Pos condición</b>	5a Es necesario introducir un fecha compromiso de Cierre de Ticket después de la Validación.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	6a	Si el solicitante no aprueba la solución del Ticket Incidencia, se cierra el Ticket Incidencia sin ser aprobado.
<b>Rendimiento</b>	<b>Paso</b>	<b>Cota de tiempo</b>
	5	5 segundos
<b>Frecuencia esperada</b>	10 veces/día	
<b>Estabilidad</b>	Alta	
<b>Comentarios</b>	La frecuencia será mucho mayor durante los dos primeros meses, probablemente 100 veces/día	

## CASO DE USO SERVICE DESK

### *CASO de USO. Validar Datos Incidencia*

	<b>Validar Datos Incidencia</b>
<b>Objetivos asociados</b>	Validar Datos de Incidencia, ingresados por el Solicitante.
<b>Requisitos asociados</b>	Datos de Incidencia registrados.
<b>Descripción</b>	ServiceDesk, valida los datos introducidos por el Solicitante, los cuales deben de ser correctos, y corresponder a las áreas asignadas.
<b>Actor</b>	ServiceDesk

<b>Precondición</b>	El Solicitante debe haber Registrado una Incidencia.  ServiceDesk, debe tener los roles correspondientes del Grupo Asignado para validar los datos de la incidencia.	
<b>Secuencia</b>  <b>Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	ServiceDesk ingresa al sistema y da clic en Inbox.
	2	El sistema muestra los Registros que se han levantado para ese Grupo de Service Desk1, en forma de tabla con los siguientes datos: <ul style="list-style-type: none"> <li>• Ticket de Incidencia</li> <li>• Nombre de la Incidencia</li> <li>• Tipo de Incidencia</li> <li>• Código de la Aplicación</li> <li>• Descripción de la Incidencia</li> <li>• Nivel de Gravedad</li> <li>• Documentación</li> <li>• Observaciones</li> <li>• Solicitante</li> <li>• Área Solicitante</li> </ul>
	3	ServiceDesk, analiza los Datos Incidencia, y Aprueba los Datos Incidencia, dando clic en el botón "Aprobar". (Flujo alterno 3a)
	4	El Sistema envía mensaje de Generar Ticket Incidencia, ServiceDesk "Acepta".
	5	El Sistema envía mensaje de notificación de Generación de Ticket Incidencia al Solicitante.
	6	El Sistema Genera el Ticket de Incidencia con los Datos Validados y Aprobados.
	7	ServiceDesk ciérrala sesion .
<b>Postcondición</b>		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3a	Si no se aprueban los Datos Incidencia ingresados por el Usuario, ServiceDesk da clic en "Rechazar".  El Sistema rechaza los datos y envía mensaje de notificación al Solicitante

<b>Rendimiento</b>	<b>Paso</b>	<b>Cota de tiempo</b>
	4	5 segundos
<b>Frecuencia esperada</b>	5 veces/día	
<b>Estabilidad</b>	Alta	
<b>Comentarios</b>	La frecuencia podría ser mayor en el primer mes del sistema se espera que baje, con la BDK.	

### ***CASO de USO. Categorización Ticket Incidencia***

	<b>Categorización Ticket Incidencia</b>	
<b>Objetivos asociados</b>	Categorizar el Ticket de Incidencia, generado por el Sistema.	
<b>Requisitos asociados</b>	Ticket de Incidencia generado por el Sistema.	
<b>Descripción</b>	ServiceDesk, Categoriza la incidencia, de acuerdo al Grupo de Soporte correspondiente al Área Solicitante.	
<b>Actor</b>	ServiceDesk	
<b>Precondición</b>	El Sistema debe haber Generado un Ticket de Incidencia, aprobado por ServiceDesk.	
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>
<b>Normal</b>	1	ServiceDesk ingresa al sistema y da clic en Inbox.
	2	<p>El sistema muestra los Ticket de Incidencia, generados por Grupo de ServiceDesk que ha iniciado sesión, cada Ticket presenta su información, en forma de tabla con los siguientes datos:</p> <ul style="list-style-type: none"> <li>• Ticket de Incidencia</li> <li>• Nombre de la Incidencia</li> <li>• Tipo de Incidencia</li> <li>• Código de la Aplicación</li> <li>• Descripción de la Incidencia</li> <li>• Nivel de Gravedad</li> <li>• Documentación</li> <li>• Observaciones</li> <li>• Solicitante</li> <li>• Área Solicitante</li> </ul>

	3	ServiceDesk, analiza el Ticket de Incidencia de acuerdo al Área Solicitante y Categoriza el Ticket a un Grupo de Soporte.
	4	El Sistema envía mensaje "Se va a Categorizar el Ticket de Incidencia al Grupo de Soporte XXX"
	5	ServiceDesk da clic en "Aceptar" la Categorización. (Flujo Alternativo 5a)
	6	El Sistema notifica a ServiceDesk y Soporte la Categorización.
	7	ServiceDesk cierra la sesión .
<b>Postcondición</b>		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3a	Si ServiceDesk, cometió un error al seleccionar el Grupo de Soporte tiene la opción de "Rechazar" la Categorización.
<b>Rendimiento</b>	<b>Paso</b>	<b>Cota de tiempo</b>
	4	5 segundos
<b>Frecuencia esperada</b>	5 veces/día	
<b>Estabilidad</b>	Alta	
<b>Comentarios</b>	La frecuencia podría ser mayor en el primer mes del sistema se espera que baje, con la BDK.	

#### **CASO de USO. Priorización Ticket Incidencia**

	<b>Priorización Ticket Incidencia</b>
<b>Objetivos asociados</b>	Priorizar el Ticket de Incidencia, Categorizado por ServiceDesk.
<b>Requisitos asociados</b>	Ticket de Incidencia Categorizado por ServiceDesk.
<b>Descripción</b>	ServiceDesk, Categoriza la incidencia, de acuerdo al Grupo de Soporte correspondiente al Área Solicitante.
<b>Actor</b>	ServiceDesk
<b>Precondición</b>	El Ticket de Incidencia, debe estar Categorizado por ServiceDesk.

<b>Secuencia</b>  <b>Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	ServiceDesk ingresa al sistema y da clic en Inbox.
	2	El sistema muestra los Ticket de Incidencia, Categorizados por ServiceDesk que ha iniciado sesión, cada Ticket presenta su información, en forma de tabla con los siguientes datos: <ul style="list-style-type: none"> <li>• Ticket de Incidencia</li> <li>• Nombre de la Incidencia</li> <li>• Tipo de Incidencia</li> <li>• Código de la Aplicación</li> <li>• Descripción de la Incidencia</li> <li>• Nivel de Gravedad</li> <li>• Documentación</li> <li>• Observaciones</li> <li>• Solicitante</li> <li>• Área Solicitante</li> </ul>
	3	ServiceDesk, Prioriza el Ticket de acuerdo a una tabla de Escalamiento.
	4	El Sistema envía mensaje “Priorizar Ticket de Incidencia como XXX”
	5	ServiceDesk da clic en “Aceptar” la Categorización. (Flujo Alternativo 5a)
	6	El Sistema notifica a ServiceDesk y Soporte la Priorización, para ejecutar la Solucion.
	7	ServiceDesk cierra la sesión .
<b>Postcondición</b>		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3a	Si ServiceDesk, cometi6 un error al priorizar el Ticket de Incidencia el Sistema muestra la opci6n de “Rechazar” la Priorizaci6n.
<b>Rendimiento</b>	<b>Paso</b>	<b>Cota de tiempo</b>
	4	5 segundos
<b>Frecuencia esperada</b>	5 veces/día	



<b>Estabilidad</b>	Alta
<b>Comentarios</b>	La frecuencia podría ser mayor en el primer mes del sistema se espera que baje, con la BDK.

**CASO de USO. Asignar- Resolver Ticket Incidencia**

	<b>Asignar-Resolver Ticket Incidencia</b>	
<b>Objetivos asociados</b>	Asignar-Resolver Ticket de Incidencia, categorizado y priorizado.	
<b>Requisitos asociados</b>	Ticket de Incidencia priorizado y categorizado por ServiceDesk.	
<b>Descripción</b>	ServiceDesk, asigna el Grupo de Soporte de Nivel 1 o Nivel 2, de acuerdo a las especificaciones de la incidencia.	
<b>Actor</b>	ServiceDesk	
<b>Precondición</b>	El Ticket de Incidencia, debe ser categorizado y aprobado por ServiceDesk.	
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>
<b>Normal</b>	1	ServiceDesk ingresa al sistema y da clic en Inbox.
	2	El sistema muestra los Ticket de Incidencia, generados por Grupo de ServiceDesk que ha iniciado sesión, cada Ticket presenta su información, en forma de tabla con los siguientes datos: <ul style="list-style-type: none"> <li>• Ticket de Incidencia</li> <li>• Nombre de la Incidencia</li> <li>• Tipo de Incidencia</li> <li>• Código de la Aplicación</li> <li>• Descripción de la Incidencia</li> <li>• Nivel de Gravedad</li> <li>• Documentación</li> <li>• Observaciones</li> <li>• Solicitante</li> <li>• Área Solicitante</li> </ul>
	3	ServiceDesk, analiza el Ticket de Incidencia de acuerdo al Área Solicitante, así como la prioridad y la categorización.
	4	ServiceDesk asigna el Ticket de Incidencia a Soporte Nivel 1, dando clic en la Opción "Asignar Soporte Nivel 1". (Flujo Alternativo 4a)

	5	ServiceDesk da clic en “Aceptar” la Asignacion. (Flujo Alternativo 5a)
	6	El Sistema notifica a ServiceDesk, Soporte y Solicitante la Asignación.
	7	ServiceDesk cierra la sesión.
<b>Postcondición</b>		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	4a	Si Soporte Nivel 1 no puede resolver el Ticket de incidencia de acuerdo a sus responsabilidades y acciones, ServiceDesk debe de re-asignar, el Ticket de Incidencia a Soporte Nivel 2 dando clic en “Asignar Soporte Nivel 2”. De acuerdo a su análisis.  (Se realiza el Flujo Básico para Asignar a Soporte Nivel 2)
<b>Rendimiento</b>	<b>Paso</b>	<b>Cota de tiempo</b>
	4	5 segundos
<b>Frecuencia esperada</b>	5 veces/día	
<b>Estabilidad</b>	Alta	
<b>Comentarios</b>	La frecuencia podría ser mayor en el primer mes del sistema se espera que baje, con la BDK.	

**CASO de USO. Cierre Ticket Incidencia**

	<b>Cerrar Ticket Incidencia</b>
<b>Objetivos asociados</b>	Cierre del Ticket de Incidencia, asignado a un Nivel de Soporte por ServiceDesk.
<b>Requisitos asociados</b>	Ticket de Incidencia asignado por ServiceDesk y validado por el Solicitante.  El sistema habilita un botón de “Cierre Ticket Incidencia” solo si tiene el Vo.Bo. del Solicitante.
<b>Descripción</b>	ServiceDesk, cierra la Solución del Ticket de Incidencia validada por el Solicitante.

<b>Actor</b>	ServiceDesk	
<b>Precondición</b>	El Solicitante debe haber enviado su Vo.Bo. en relación a la Solución de Soporte, para el Ticket de Incidencia asignado.	
<b>Secuencia</b>  <b>Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	ServiceDesk ingresa al sistema y da clic en Inbox.
	2	El sistema muestra los Ticket de Incidencia, solucionados, cada Ticket presenta su información, en forma de tabla con los siguientes datos: <ul style="list-style-type: none"> <li>• Ticket de Incidencia</li> <li>• Nombre de la Incidencia</li> <li>• Tipo de Incidencia</li> <li>• Código de la Aplicación</li> <li>• Descripción de la Incidencia</li> <li>• Nivel de Gravedad</li> <li>• Documentación</li> <li>• Observaciones</li> <li>• Solicitante</li> <li>• Área Solicitante</li> <li>• Solución Ticket Incidencia.</li> </ul>
	3	El Sistema habilita un botón para cerrar el Ticket de Incidencia, que cuente con el Vo.Bo. del Solicitante.
	4	ServiceDesk da clic en el botón "Cerrar Ticket de incidencia".
	5	El Sistema envía mensaje de Advertencia "Esta seguro de cerrar el Ticket". (Flujo Alterno 5a)
	6	ServiceDesk cierra el ticket de Incidencia.
	7	ServiceDesk cierra la sesión .
<b>Postcondición</b>		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	5a	Si ServiceDesk, cometió un error al seleccionar otro Ticket de Incidencia , puede cancelar la opción de "Cerrar Ticket de incidencia".
<b>Rendimiento</b>	<b>Paso</b>	<b>Cota de tiempo</b>
	4	5 segundos

<b>Frecuencia esperada</b>	5 veces/día
<b>Estabilidad</b>	Alta
<b>Comentarios</b>	La frecuencia podría ser mayor en el primer mes del sistema se espera que baje, con la BDK.

## CASO DE USO SOPORTE NIVEL 1

### *CASO de USO. Resolver Ticket Incidencia*

	<b>Validar Ticket Incidencia</b>	
<b>Objetivos asociados</b>	Resolver Ticket de Incidencia.	
<b>Requisitos asociados</b>	Ticket de incidencia asignado por ServiceDesk.  El Sistema muestra las opciones para el Rol Soporte Nivel 1.	
<b>Descripción</b>	Soporte Nivel 1 ingresa a su Inbox, para ver sus Tickets de Incidencias asignados por ServiceDesk y resolver el Ticket.	
<b>Actor</b>	Soporte Nivel 1	
<b>Precondición</b>	Soporte Nivel 1 debe tener los recursos asignados para resolver la incidencia de acuerdo a su grupo asignado.	
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>
<b>Normal</b>	1	Soporte Nivel 1 ingresa al sistema y da clic en Inbox.
	2	El sistema muestra los Tickets Asignados a ese Soporte Nivel 1, en forma de tabla con los siguientes datos: <ul style="list-style-type: none"> <li>• Ticket de Incidencia</li> <li>• Nombre de la Incidencia</li> <li>• Tipo de Incidencia</li> <li>• Código de la Aplicación</li> <li>• Descripción de la Incidencia</li> <li>• Nivel de Gravedad</li> <li>• Documentación</li> <li>• Observaciones</li> <li>• Solicitante</li> <li>• Área Solicitante</li> </ul>

	3	Soporte Nivel 1, analiza la incidencia, y revisa en la BDK si existe una incidencia similar resuelta. (Flujo alterno 3a)
	4	Soporte Nivel 1 agrega la Solución de la BDK
	5	Soporte Nivel 1 cierra temporalmente el Ticket de Incidencia, previo al Solicitante.
<b>Postcondición</b>		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3a	Si la solución no se encuentra en la BDK, se renvía el mail a ServiceDesk, para que pueda ser re asignado a otro Nivel Soporte.
<b>Rendimiento</b>	<b>Paso</b>	<b>Cota de tiempo</b>
	4	5 segundos
<b>Frecuencia esperada</b>	5 veces/día	
<b>Estabilidad</b>	Alta	
<b>Comentarios</b>	La frecuencia podría ser mayor en el primer mes del sistema se espera que baje, con la BDK.	

## CASO DE USO SOPORTE NIVEL 2

### *CASO de USO. Resolver Ticket Incidencia*

	<b>Solucionar Ticket Incidencia</b>	
<b>Objetivos asociados</b>	Resolver Ticket de Incidencia.	
<b>Requisitos asociados</b>	Ticket de incidencia re asignado.  El Sistema muestra las opciones para el Rol Soporte Nivel 2.	
<b>Descripción</b>	Soporte Nivel 2 ingresa a su Inbox, para ver sus Tickets de Incidencias asignados por ServiceDesk y resolver el Ticket.	
<b>Actor</b>	Soporte Nivel 2	
<b>Precondición</b>	Soporte Nivel 2 debe tener los recursos asignados para resolver la incidencia de acuerdo a su grupo asignado.	
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>

<b>Normal</b>	1	Soporte Nivel 2 ingresa al sistema y da clic en Inbox.
	2	El sistema muestra los Tickets Asignados a ese Soporte Nivel 1, en forma de tabla con los siguientes datos: <ul style="list-style-type: none"> <li>• Ticket de Incidencia</li> <li>• Nombre de la Incidencia</li> <li>• Tipo de Incidencia</li> <li>• Código de la Aplicación</li> <li>• Descripción de la Incidencia</li> <li>• Nivel de Gravedad</li> <li>• Documentación</li> <li>• Observaciones</li> <li>• Solicitante</li> <li>• Área Solicitante</li> </ul>
	3	Soporte Nivel 2, analiza la incidencia, y resuelve el Ticket Incidencia. (Flujo alterno 3a)
	4	Soporte Nivel 2 agrega la Solución de la BDK
	5	Soporte Nivel 2 cierra temporalmente el Ticket de Incidencia, previo al Solicitante.
<b>Postcondición</b>		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3a	Si no se tiene una solución para el error se cierra el Ticket Incidencia con una observación indicando el motivo por el cual no se pudo resolver el Ticket.
<b>Rendimiento</b>	<b>Paso</b>	<b>Cota de tiempo</b>
	4	5 segundos
<b>Frecuencia esperada</b>	5 veces/día	
<b>Estabilidad</b>	Alta	
<b>Comentarios</b>	La frecuencia podría ser mayor en el primer mes del sistema se espera que baje, con la BDK.	