

## DECLARATORIA

Declaro que la investigación es absolutamente original, autentica, personal, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes. Las ideas, doctrinas resultados y conclusiones a los que he llegado son de mi absoluta responsabilidad.

---

Iván Naranjo Herrera

CC 1711186153

## CONTRATO DE CESIÓN SOBRE DERECHOS PROPIEDAD INTELECTUAL

Comparecen a la celebración del presente contrato de cesión y transferencia de derechos de propiedad intelectual, por una parte, el estudiante **IVAN NARANJO HERRERA**, por sus propios y personales derechos, a quien en lo posterior se le denominará el “CEDENTE”; y, por otra parte, el INSTITUTO SUPERIOR TECNOLÓGICO CORDILLERA, representado por su Rector el Ingeniero Ernesto Flores Córdova, a quien en lo posterior se lo denominará el “CESIONARIO”. Los comparecientes son mayores de edad, domiciliados en esta ciudad de Quito Distrito Metropolitano, hábiles y capaces para contraer derechos y obligaciones, quienes acuerdan al tenor de las siguientes cláusulas:

**PRIMERA: ANTECEDENTE.-** a) El Cedente dentro del pensum de estudio en la carrera de análisis de sistemas que imparte el Instituto Superior Tecnológico Cordillera, y con el objeto de obtener el título de Tecnólogo en Análisis de Sistemas, el estudiante participa en el proyecto de grado denominado “**CONTROL DE PEDIDOS E INVENTARIO DE PRODUCTOS MEDIANTE UNA APLICACIÓN WEB PARA LA EMPRESA DE MATERIAL ELÉCTRICO IMPORT**”, el cual incluye la creación y desarrollo del programa de ordenador o software, para lo cual ha implementado los conocimientos adquiridos en su calidad de alumno. b) Por iniciativa y responsabilidad del Instituto Superior Tecnológico Cordillera se desarrolla la creación del programa de ordenador, motivo por el cual se regula de forma clara la cesión de los derechos de autor que genera la obra literaria y que es producto del proyecto de grado, el mismo que culminado es de plena aplicación técnica, administrativa y de reproducción.

**SEGUNDA: CESIÓN Y TRANSFERENCIA.-** Con el antecedente indicado, el Cedente libre y voluntariamente cede y transfiere de manera perpetua y gratuita todos los derechos patrimoniales del programa de ordenador descrito en la cláusula anterior a favor del Cesionario, sin reservarse para sí ningún privilegio especial (código fuente, código objeto,

diagramas de flujo, planos, manuales de uso, etc.). El Cesionario podrá explotar el programa de ordenador por cualquier medio o procedimiento tal cual lo establece el Artículo 20 de la Ley de Propiedad Intelectual, esto es, realizar, autorizar o prohibir, entre otros: a) La reproducción del programa de ordenador por cualquier forma o procedimiento; b) La comunicación pública del software; c) La distribución pública de ejemplares o copias, la comercialización, arrendamiento o alquiler del programa de ordenador; d) Cualquier transformación o modificación del programa de ordenador; e) La protección y registro en el IEPI el programa de ordenador a nombre del Cesionario; f) Ejercer la protección jurídica del programa de ordenador; g) Los demás derechos establecidos en la Ley de Propiedad Intelectual y otros cuerpos legales que normen sobre la cesión de derechos de autor y derechos patrimoniales.

**TERCERA: OBLIGACIÓN DEL CEDENTE.-** El cedente no podrá transferir a ningún tercero los derechos que conforman la estructura, secuencia y organización del programa de ordenador que es objeto del presente contrato, como tampoco emplearlo o utilizarlo a título personal, ya que siempre se deberá guardar la exclusividad del programa de ordenador a favor del Cesionario.

**CUARTA: CUANTIA.-** La cesión objeto del presente contrato, se realiza a título gratuito y por ende el Cesionario ni sus administradores deben cancelar valor alguno o regalías por este contrato y por los derechos que se derivan del mismo.

**QUINTA: PLAZO.-** La vigencia del presente contrato es indefinida.

**SEXTA: DOMICILIO, JURISDICCIÓN Y COMPETENCIA.-** Las partes fijan como su domicilio la ciudad de Quito. Toda controversia o diferencia derivada de éste, será resuelta directamente entre las partes y, si esto no fuere factible, se solicitará la asistencia de un Mediador del Centro de Arbitraje y Mediación de la Cámara de Comercio de Quito. En el evento que el conflicto no fuere resuelto mediante este procedimiento, en el plazo de



diez días calendario desde su inicio, pudiendo prorrogarse por mutuo acuerdo este plazo, las partes someterán sus controversias a la resolución de un árbitro, que se sujetará a lo dispuesto en la Ley de Arbitraje y Mediación, al Reglamento del Centro de Arbitraje y Mediación de la Cámara de comercio de Quito, y a las siguientes normas: a) El árbitro será seleccionado conforme a lo establecido en la Ley de Arbitraje y Mediación; b) Las partes renuncian a la jurisdicción ordinaria, se obligan a acatar el laudo arbitral y se comprometen a no interponer ningún tipo de recurso en contra del laudo arbitral; c) Para la ejecución de medidas cautelares, el árbitro está facultado para solicitar el auxilio de los funcionarios públicos, judiciales, policiales y administrativos, sin que sea necesario recurrir a juez ordinario alguno; d) El procedimiento será confidencial y en derecho; e) El lugar de arbitraje serán las instalaciones del centro de arbitraje y mediación de la Cámara de Comercio de Quito; f) El idioma del arbitraje será el español; y, g) La reconvencción, caso de haberla, seguirá los mismos procedimientos antes indicados para el juicio principal.

**SÉPTIMA: ACEPTACIÓN.-** Las partes contratantes aceptan el contenido del presente contrato, por ser hecho en seguridad de sus respectivos intereses.

En aceptación firman a los \_\_\_\_ días del mes de \_\_\_\_ del dos mil catorce.

f) \_\_\_\_\_

C.C. N°

Cordillera

**CEDENTE**

f) \_\_\_\_\_

Instituto Superior Tecnológico

**CESIONARIO**



## Índice General

➤ Portada	
➤ Caratula	
➤ Declaración de Aprobación Tutor y Lector	i
➤ Declaración de autoría del estudiante	ii
➤ Declaración de cesión de derechos a la institución	iii
➤ Índice general	vi
➤ Índice de tablas	viii
➤ Índice de figuras	ix
➤ Índice de anexos	x
➤ Resumen Ejecutivo	xi
➤ Abstract	xii

Titulo	Página
Capítulo I: Antecedentes .....	1
1.01 Contexto .....	1
1.02 Justificación .....	2
1.03 Matriz T de análisis de fuerzas.....	2
1.03.01 Análisis de la matriz T .....	3
Capítulo II: Involucrados .....	5
2.01 Requerimientos .....	5
2.01.01 Descripción del sistema actual .....	5
2.01.2 Visión y alcance .....	5
2.01.3 Entrevistas .....	7
2.01.04 Matriz de Requerimientos .....	8
2.01.05 Descripción Detallada .....	9
2.02. Mapeos de Involucrados .....	12
2.02 Mapeos de Involucrados .....	13
2.03 Matriz de Análisis de Involucrados.....	14
Análisis de la matriz de involucrados .....	15
Capítulo III: Árbol de Problemas - Objetivos .....	16
3.01 Árbol de Problemas.....	16
Análisis del árbol de problemas. ....	17
3.01 Árbol de Objetivos .....	18



Análisis del árbol de objetivos. ....	18
4.01 Matriz de Análisis de Impactos de los Objetivos. ....	29
4.02. Estándares para el Diseño de Clases .....	29
4.03. Diagrama de clases.....	30
4.04. Modelo Lógico – Físico .....	32
4.05. Diagrama de Componentes .....	32
4.07. Matriz del Marco Logico .....	35
4.08 Vistas Arquitectonicas .....	35
4.08.01. Vista lógica .....	35
4.08.02. Vista física.....	35
4.08.03. Vista de desarrollo.....	36
Capítulo V: Propuesta .....	38
5.01. Especificación de estándares de programación .....	38
Programación Orientada a Objetos .....	38
5.02. Diseño de Interfaces de Usuario .....	39
5.03. Especificación de pruebas de unidad .....	41
5.04. Especificación de pruebas de aceptación .....	43
5.05. Especificación de pruebas de carga.....	45
5.06. Configuración del Ambiente mínima/ideal .....	46
5.06.01 Rendimiento .....	46
5.06.02 Características Técnicas .....	46
5.06.03 Componentes del Sistema .....	46
5.06.04 Requerimientos de Hardware y Software.....	46
5.06.04.01 Requerimientos de Hardware .....	47
5.06.04.02 Requerimientos de Software .....	47
CAPITULO VI.....	48
6.01.Recursos .....	48
6.02 Presupuesto .....	48
6.3 Cronograma de Actividades .....	49
Capítulo VII: Conclusiones y Recomendaciones.....	50
7.01 Conclusiones .....	50
7.02 Recomendaciones.....	50
ANEXOS.....	54
BIBLIOGRAFIA .....	102

## Índice de tablas

Titulo	Página
Tabla 1 Matriz T del Problema Central.....	3
Tabla 2 Entrevista determinada para el Técnico .....	7
Tabla 3 Matriz de requerimientos funcionales.....	8
Tabla 4 Matriz de requerimientos no funcionales .....	8
Tabla 5 Descripción detallada Requerimiento Funcional 001 .....	9
Tabla 6 Descripción detallada Requerimiento Funcional 002 .....	10
Tabla 7 Descripción detallada Requerimiento Funcional 003 .....	11
Tabla 8 Matriz De Requerimientos No Funcionales .....	12
Tabla 9 Matriz de involucrados.....	14
Tabla 10 Especificación del caso de uso de realización. Recepcion de incidencia.....	20
Tabla 11 Especificación del caso de uso de realización. Creación de incidencia .....	21
Tabla 12 Especificación del caso de uso de realización. Notificacion de incidencia.....	22
Tabla 13 Especificación del caso de uso de realización. Gestión de incidencia .....	23
Tabla 14 Especificación de caso de uso. Receptar de incidencia.....	25
Tabla 15 Especificación de caso de uso. Crear incidencia.....	26
Tabla 16 Especificación de caso de uso. Notificación incidencia.....	26
Tabla 17 Especificación de caso de uso. Gestionar incidencia .....	27
Tabla 18 Matriz de Análisis de Alternativas .....	28
Tabla 19 Diagrama de paquetes .....	36
Tabla 20 Especificación de pruebas de unidad .....	41
Tabla 21 Especificación de pruebas de unidad .....	42
Tabla 22 Especificación de pruebas de unidad .....	42
Tabla 23 Especificación de pruebas de unidad .....	43
Tabla 24 Especificación de pruebas de aceptación .....	43
Tabla 25 Especificación de pruebas de aceptación .....	44
Tabla 26 Especificación de pruebas de aceptación .....	44
Tabla 27 Pruebas de carga del sistema.....	45
Tabla 28 Recursos del proyecto .....	48
Tabla 29 Presupuesto.....	48

## Índice de Figuras

Titulo	Página
Figura 1: Mapeo de Involucrados: .....	13
Figura 2: Árbol de Problemas: .....	16
Figura 3: Árbol de Objetivos:.....	18
Figura 4. Diagrama de caso de uso de realización.....	19
Figura 5. Diagrama de caso de uso de realización.....	20
Figura 6. Diagrama de caso de uso de realización.....	21
Figura 7. Diagrama de caso de uso de realización.. ..	22
Figura 9. Diagrama de Secuencia.....	24
Figura 10. Diagrama de Secuencia.....	24
Figura 11. Diagrama de Secuencia. ....	24
Figura 12. Diagrama de Secuencia.....	25
Figura 13. Diagrama de clases.....	31
Figura 14. Diagrama de clases - Modelo Lógico. ....	32
Figura 15: Diagrama de Componentes.....	32
Figura 16: Diagrama de Componentes. ....	33
Figura 17:Diagrama de Estrategias. ....	34
Figura 18. Vista Lógica. ....	35
Figura 19. Vista Física. ....	35
Figura 20. Diagrama de actividades .....	36
Figura 21. Diagrama de actividades de notificación de incidencia. ....	37
Figura 22. Diagrama de actividades .....	37
Figura 23. Restricción de Acceso.....	39
Figura 24. Busqueda de cliente. ....	39
Figura 25. Creación de cliente. ....	40
Figura 26. Administrar motivos de incidencia. ....	40
Figura 27. Administrar prioridad de incidencia.....	40
Figura 28. Administrar responsable de incidencia. ....	41
Figura 29. Pruebas de carga. ....	45
Figura 30 Pruebas de carga. ....	46
Figura 31. Cronograma de actividades.....	49
Figura 32 Cronograma de actividades.....	49





## Índice de Anexos

Titulo	Página
A.01 Matriz de impactos de objetivos.....	52
A.02 Matriz Del Marco Lógico	
A.03 Manual de instalación.....	56
A.04 Manual de usuario .....	63
A.05 Manual técnico .....	66
A.06 Diccionario de datos:.....	92
BIBLIOGRAFIA .....	102

## Resumen Ejecutivo

El control de incidencias tiene como objetivo resolver, de la manera más rápida y eficaz posible, cualquier incidente que cause una interrupción de cualquier tipo de servicio.

Es frecuente que existan múltiples incidencias concurrentes, por lo que es necesario determinar un nivel de prioridad para la resolución de las mismas.

La priorización se basa esencialmente en cuatro parámetros: Urgente, alta, media, baja.

La clasificación de un incidente tiene como objetivo principal el recopilar toda la información que pueda ser utilizada para la resolución del mismo.

El proceso de clasificación consta, de los siguientes pasos:

**Categorización:** se asigna una categoría, dependiendo del tipo de incidente o del grupo de trabajo responsable de su resolución.

**Nivel de prioridad:** dependiendo del impacto y la urgencia se determina, según criterios preestablecidos.

**Asignación de recursos:** si el departamento técnico, no puede resolver el incidente en primera instancia, designará al personal de soporte técnico responsable de su resolución.

**Monitorización del estado y tiempo de respuesta esperado:** se asocia un estado al incidente.

## Abstract

The incident control aims to solve, as quickly and efficiently as possible, any incident causing an interruption of any service.

Often there are multiple concurrent incidents, so it is necessary to establish a level of priority for resolving them.

The prioritization is based essentially on four parameters: Urgent, High, Medium, Low.

The classification of an incident's main objective is to gather all the information that can be used to solve it.

The classification process consists of the following steps:

**Categorization:** a category is assigned, depending on the type of incident or group responsible for resolving labor.

**Priority level:** depending on the impact and urgency is determined according to criteria.

**Resource allocation:** if the technical department, can not resolve the incident in the first instance, shall appoint the technical support staff responsible for its resolution.

**Monitoring the status and expected response time:** a state associated to the incident.

## Capítulo I: Antecedentes

### 1.01 Contexto

La sociedad actual ha convertido a las TIC (Tecnologías de la información y la comunicación), en un instrumento de innovación y apoyo a varios campos propios de la vida laboral y personal, ofreciendo un nuevo espacio de innovación en ámbitos como la industria, los servicios, la salud, la administración, el comercio y la educación.

El crecimiento constante de las sociedades y sus poblaciones ha ido de la mano del uso y el mejoramiento constante de las TIC, que han influenciado directamente sobre el desarrollo diario de sus actividades y por ende ha influido en el desarrollo de nuevas aplicaciones tecnológicas.

En el ámbito laboral la influencia de las tecnologías de información de la comunicación se ve reflejada en la distribución del tiempo laboral y en la forma en que se desarrolla la producción y la naturaleza del trabajo realizado en los distintos tipos de sociedades.

Reduciendo en gran parte las horas de cómputo y orientando las horas al servicio a las personas e innovación constante en nuevos usos de la tecnología.

La implementación de la tecnología en el ambiente laboral ha incrementado los cambios en las actividades diarias dando un cambio constante y radical sobre los oficios ya existentes y que se verá reflejado en un corto plazo con la creación de nuevos oficios.

Un reflejo de la incorporación de las TI es la constante digitalización y la velocidad con la que está ocurriendo han impactado en la forma en la que se materializan las estrategias del negocio.

El ámbito económico de los servicios está intrínsecamente relacionado con la información y el conocimiento, y en este sector, las tecnologías de la información y las telecomunicaciones pueden tener efectos destructivos y creadores a la vez.

Cada vez es mayor la dimensión de las actividades en las que las personas trabajan en relación con otras personas y es aquí donde las tecnologías de la información y las

telecomunicaciones tendrán un papel transformador, creando un menor número de puestos de trabajo de mayor cualificación tecnológica.

## **1.02 Justificación**

Las organizaciones tienden a tener una mayor dependencia de las Tecnologías de la Información. Los departamentos de sistemas de información, y las actividades en ellos desarrolladas, han sido tradicionalmente vistos como una área de soporte al negocio, descuidando incluso muchas veces el uso de criterios racionales para medir su rentabilidad, eficacia y la calidad del servicio ofrecidos a toda la organización.

ITIL no implica una nueva manera de pensar y actuar.

Sí proporciona un marco en el cual plantea los métodos existentes y actividades en un contexto estructurado.

Se deben distinguir para esto:

Procesos y Tareas: Un Proceso es un conjunto de actividades o eventos que se realizan o suceden con un determinado fin son el más alto nivel para definir actividades y construir estándares en una organización.

Por otro lado una Tarea es una actividad definida es específica y contribuye al cumplimiento de la misión general para cada tarea

## **1.03 Matriz T de análisis de fuerzas**

En la siguiente matriz describelas situaciones mediante la cual describiremos los extremos a evitar, la problemática en el presente y lo que buscamos al hacer nuestra investigación.

Además describe las fuerzas mediante la cual son calificadas con la intensidad y el potencial de campo en la escala de 1 al 5.

Tabla 1

Matriz T del Problema Central

SITUACIÓN EMPEORADA	SITUACIÓN ACTUAL				SITUACIÓN MEJORADA
Cientes descontentos por información y atención que se le facilita.	Inadecuado control y registro de incidencias.				Mejorar la calidad de atención de la empresa a los clientes.
Fuerzas Impulsadoras	I	PC	I	PC	Fuerzas Bloqueadoras
Optimizar el proceso de control incidencias.	3	4	3	4	Procesos incorrectos en el control de incidencias.
Estabilizar las actividades de la empresa.	3	4	2	4	Confusión con la información incidencias.
El uso de la información de manera digital.	4	3	3	4	La ausencia de personal.
Mejorar el ambiente de trabajo.	2	3	2	3	No existe reporte de incidencias.
Apoyo al desarrollo de la empresa.	1	3			

Matriz de Análisis de Fuerzas T. En esta matriz detallamos las fuerzas bloqueadoras que nos impiden lograr la situación mejorada y las fuerzas impulsadoras que nos ayudaran a cumplir con la propuesta del proyecto y llegar así a la situación mejorada.

I= Intensidad

PC= Potencial de Cambio

### 1.03.01 Análisis de la matriz T

Después de realizar el Análisis de fuerzas en la matriz podemos dar a conocer la intensidad y el potencial de cambio de las fuerzas T, gracias a las actividades que se realiza se ha conseguido que la situación actual no empeore.

Los problemas que afronta la empresa son por las incidencias que se los realiza en forma manual, así como también reportes, por tal motivo el sistema permitirá optimizar los



procesos, actividades, estabilizar y mejorar el ambiente de trabajo e impulsando a la empresa a utilizar herramientas tecnológicas y realizar la capacitación necesaria.

Con la optimización del control de incidencias mediante una herramienta informática lograremos evitar errores en el registro de incidencias de igual manera la generación de información se realizara en un menor tiempo y con la menor cantidad de errores

## Capítulo II: Involucrados

### 2.01 Requerimientos

#### 2.01.01 Descripción del sistema actual

Actualmente la empresa "TRADESYSTEM", maneja el control de incidencias de forma manual, toda la información es almacenada en una hoja de cálculo de Microsoft Excel, trayendo como consecuencias un inadecuado seguimiento y control de incidencias.

La notificación de incidencia, se la realiza con una llamada telefónica en el cual el departamento de soporte se encarga de recibir la llamada y formalizar el registro.

Después que la notificación es analizada, un técnico es enviado de acuerdo al tipo de incidencia. Los tipos de incidencia pueden variar por eso es necesario que los especialistas encargados analicen y den solución al problema.

El técnico es el encargado de notificar de manera verbal, cuál de los siguientes servicios será facturado: Servicios profesionales, asistencia técnica, partes y piezas de computadoras

#### 2.01.2 Visión y alcance

El control de incidencias tiene como objetivo resolver, de la manera más rápida y eficaz posible, cualquier incidente que cause una interrupción de cualquier tipo de servicio.

Los técnicos no se preocuparán por encontrar, las causas subyacentes a un determinado incidente sino exclusivamente a restaurar el servicio.

Sin embargo, es evidente, que existe una fuerte interrelación entre ambas.

Es frecuente que existan múltiples incidencias concurrentes, por lo que es necesario determinar un nivel de prioridad para la resolución de las mismas.

La priorización se basa esencialmente en cuatro parámetros: Urgente, alta, media y baja.



Determinando así la importancia de la incidencia dependiendo de cómo ésta afecta a los procesos de negocio y/o del número de usuarios afectados.

Analizaremos un tiempo estimado que acepte el cliente para la resolución de la incidencia y/o el nivel de servicio acordado en el SLA.

Dependiendo de la prioridad, se asignarán los recursos necesarios para la resolución de la incidencia, la misma puede cambiar durante su ciclo de vida.

Por ejemplo, se pueden encontrar soluciones temporales que restauren aceptablemente los niveles de servicio y que permitan retrasar el cierre del incidente sin graves repercusiones.

La clasificación de un incidente tiene como objetivo principal el recopilar toda la información que pueda ser utilizada para la resolución del mismo.

El proceso de clasificación consta, de los siguientes pasos:

**Categorización:** se asigna una categoría, dependiendo del tipo de incidente o del grupo de trabajo responsable de su resolución.

**Nivel de prioridad:** dependiendo del impacto y la urgencia se determina, según criterios preestablecidos.

**Asignación de recursos:** si el departamento técnico, no puede resolver el incidente en primera instancia, designará al personal de soporte técnico responsable de su resolución (segundo nivel).

**Monitorización del estado y tiempo de respuesta esperado:** se asocia un estado al incidente (por ejemplo: abierto, en proceso, gestionado, cerrado, facturado) y se estima el tiempo de resolución del incidente en base al SLA correspondiente y la prioridad.

### 2.01.3 Entrevistas

Tabla 2

*Entrevista determinada para el Técnico.*

IDENTIFICADOR: 001		
PREGUNTAS	OBJETIVOS	ANÁLISIS POSTERIOR
¿Cuál es la problemática que se pretende automatizar a través del sistema de información?	Determinar cuáles son los procesos que se van a automatizar	Administración y registro de una incidencia.
¿Cuántas personas tienen acceso al control de incidencias?	Conocer las sesiones de usuarios con las cuales contara el sistema informático.	Las personas que tendrán acceso a la información son: Técnicos y responsables.
¿Quiénes son los encargados del control y administración de incidencias?	Establecer el número de involucrados.	El responsable de recibir la incidencia y los técnicos.
¿Cuál es el proceso para registrar la incidencia?	Conocer claramente cuál es el procedimiento un registro correcto	El responsable recepta la llamada y registra en una hoja de cálculo.

*Entrevista determinada para el técnico. En esta tabla detallamos la entrevista que se llevó a cabo para hacer el levantamiento de información. Las cuales permitieron conocer los permisos, restricciones y relevancia del control de incidencia en la Empresa TRADESYSTEM.*

## 2.01.04 Matriz de Requerimientos

Tabla 3

*Matriz de requerimientos funcionales*

Identificador	Descripción	Fuente	Prioridad	Tipo	Estado	Usuario Involucrados
<b>REQUERIMIENTOS FUNCIONALES</b>						
RF001	Creación de incidencias	Técnico	Alta	Sistema	En revisión	Centro de Servicios
RF002	Notificación de una incidencia	Técnico	Alta	Sistema	En revisión	Centro de Servicios
RF003	Monitoreo de una incidencia	Técnico	Alta	Sistema	En revisión	Centro de Servicios

*Matriz de Requerimientos. En esta matriz detallamos los requerimientos funcionales que obtuvimos con las entrevistas. Las cuales describen la interacción entre el sistema y su ambiente.*

Tabla 4

*Matriz de requerimientos no funcionales*

Identificador	Descripción	Fuente	Prioridad	Tipo	Estado	Usuarios Involucrados
<b>REQUERIMIENTOS NO FUNCIONALES</b>						
NRF001	La interfaz debe ser de fácil manejo para los usuarios.	Tecnico	Media	Usuario	En revisión	Centro de servicios, administración, empleados

*Matriz de Requerimientos. En esta matriz detallamos los requerimientos no funcionales que obtuvimos con las entrevistas. Los cuales describen aspectos del sistema que son visibles por el usuario.*

## 2.01.05 Descripción Detallada

Tabla 5

*Descripción detallada Requerimiento Funcional 001*

Creación de incidencias		Estado	Análisis	
<b>Creado por</b>	Ivan Naranjo		<b>Actualizado por</b>	Ivan Naranjo
<b>Fecha Creación</b>	14/06/2014	<b>Fecha de Actualización</b>	16/06/2014	
<b>Identificador</b>	RF001			
<b>Tipo de Requerimiento</b>	Crítico		<b>Tipo de Requerimiento</b>	Funcional
<b>Datos de Entrada</b>	<ul style="list-style-type: none"><li>➤ Datos del cliente</li><li>➤ Tipicacion, responsable</li><li>➤ Motivo y submotivo</li><li>➤ Fecha de gestión, estado de gestión</li><li>➤ Prioridad</li><li>➤ Nimero de factura, Valor</li><li>➤ Concepto</li><li>➤ Observaciones</li></ul>			
<b>Descripción</b>	Una vez ingresada la información de la incidencia se genera un nuevo registro			
<b>Datos de salida</b>	Incidencia ingresada con éxito			
<b>Resultados Esperados</b>	Incidencia almacenada correctamente.			
<b>Origen</b>	Tecnico			
<b>Dirigido a</b>	Control de servicios			
<b>Prioridad</b>	8			
<b>Requerimientos Asociados</b>	Ninguno			
<b>ESPECIFICACIÓN</b>				
<b>Precondiciones</b>	Para ejecutar el requerimiento el cliente debe estar creado			
<b>Poscondiciones</b>	Se podrá realizar una búsqueda de la nueva incidencia.			
<b>Criterios de Aceptación</b>	Permite que el usuario pueda establecer sus búsquedas.			

*Matriz del Detalle de Requerimientos. En esta matriz detallamos el requerimiento funcional que permitirá la creación de una incidencia.*

1 – 3 = Nivel de Prioridad Baja  
4 – 6 = Nivel de Prioridad Media  
7 – 10 = Nivel de Prioridad Alta

Tabla 6

*Descripción detallada Requerimiento Funcional 002*

Gestión de la incidencia		Estado	Análisis	
<b>Creado por</b>	Ivan Naranjo		<b>Actualizado por</b>	Ivan Naranjo
<b>Fecha Creación</b>	14/06/2014	<b>Fecha de Actualización</b>	16/06/2014	
<b>Identificador</b>	RF002			
<b>Tipo de Requerimiento</b>	Crítico		<b>Tipo de Requerimiento</b>	Funcional
<b>Datos de Entrada</b>	<ul style="list-style-type: none"><li>➤ Datos del cliente</li><li>➤ Fecha de gestión</li><li>➤ Motivo y submotivo</li><li>➤ Prioridad</li><li>➤ Estado de gestión</li><li>➤ Numero de factura</li><li>➤ Valor</li><li>➤ concepto</li><li>➤ Observaciones</li></ul>			
<b>Descripción</b>	Una vez ingresada la información de la incidencia se enviara un técnico.			
<b>Datos de salida</b>	El técnico será el encargado de cambiar el estado si la incidencia ha sido resuelta, dar prioridad y dar un correcto seguimiento.			
<b>Resultados Esperados</b>	Incidencia gestionada correctamente.			
<b>Origen</b>	Tecnico			
<b>Dirigido a</b>	Control de servicios			
<b>Prioridad</b>	8			
<b>Requerimientos Asociados</b>	Ninguno			
<b>ESPECIFICACIÓN</b>				
<b>Precondiciones</b>	Para ejecutar el requerimiento la incidencia debe estar creada.			
<b>Poscondiciones</b>	Se podrá cambiar el estado de la incidencia o asignar un nuevo responsable.			
<b>Criterios de Aceptación</b>	Permite que el usuario pueda establecer sus búsquedas.			

*Matriz del Detalle de Requerimientos. En esta matriz detallamos el requerimiento funcional que permitirá la recepción y notificación de una incidencia.*

1 – 3 = Nivel de Prioridad Baja  
4 – 6 = Nivel de Prioridad Media  
7 – 10 = Nivel de Prioridad Alta

Tabla 7

*Descripción detallada Requerimiento Funcional 003*

Monitoreo de una incidencia		Estado	Análisis	
Creado por	Ivan Naranjo		Actualizado por	Ivan Naranjo
Fecha Creación	14/06/2014	Fecha de Actualización	16/06/2014	
Identificador	RF003			
Tipo de Requerimiento	Crítico		Tipo de Requerimiento	Funcional
Datos de Entrada	<div>➤ Datos del cliente</div> <div>➤ Historial de incidencias</div>			
Descripción	Una vez ingresada la información de la incidencia se enviara un tecnico			
Datos de salida	<div>➤ Usuario</div> <div>➤ Cliente</div> <div>➤ Tipicacion</div> <div>➤ Motivo</div> <div>➤ Fecha</div> <div>➤ Tiempo</div> <div>Responsable</div>			
Resultados Esperados	Incidencia notificada con exito.			
Origen	Tecnico			
Dirigido a	Control de servicios			
Prioridad	8			
Requerimientos Asociados	Ninguno			
ESPECIFICACIÓN				
Precondiciones	Para ejecutar el requerimiento la incidencia debe estar creada.			
Poscondiciones	Se podrá cambiar el estado de la incidencia			
Criterios de Aceptación	Permite que el usuario pueda establecer sus búsquedas.			

*Matriz del Detalle de Requerimientos. En esta matriz detallamos el requerimiento funcional que permitirá el monitoreo de una incidencia.*

1 – 3 = Nivel de Prioridad Baja  
 4 – 6 = Nivel de Prioridad Media  
 7 – 10 = Nivel de Prioridad Alta

Tabla 8

*Matriz De Requerimientos No Funcionales*

Identificador	Descripción	Fuente	Prioridad	Tipo	Estado	Usuarios Involucrados
<b>REQUERIMIENTOS NO FUNCIONALES</b>						
NRF001	La interfaz debe ser de fácil manejo para los usuarios.	Tecnico	Media	Usuario	En revisión	Administrador or Centro de servicios

*Matriz de Requerimientos. En esta matriz detallamos los requerimientos no funcionales que obtuvimos con las entrevistas. Los cuales describen aspectos del sistema que son visibles por el usuario que no incluyen una relación directa con el comportamiento funcional del sistema*

**2.02.Mapeos de Involucrados**

Los Empleados revisan, analizan, consultan, realizan los informes de forma manual, indican el procedimientos que realizan. Se interesará en los módulos que se implementan en el proyecto, ayudará a solventar y definir los resultados esperados con la toma de decisiones que habitualmente realizadas en la empresa.

El Administrador es el responsable directo de lograr los objetivos, normas, mandatos, leyes, que rigen en la empresa y necesita ejercer su autoridad a través de factores en el cual ayuden a su ética, liderazgo y permita ejercer un cumplimiento considerable y satisfactorio.

Los Clientes son individuos que analizan el sector y describen que no hay la suficiente información sobre los procesos que realiza la empresa, por tal motivo el presente proyecto permitirá brindar la información necesaria a través de las disposiciones que emita la empresa.

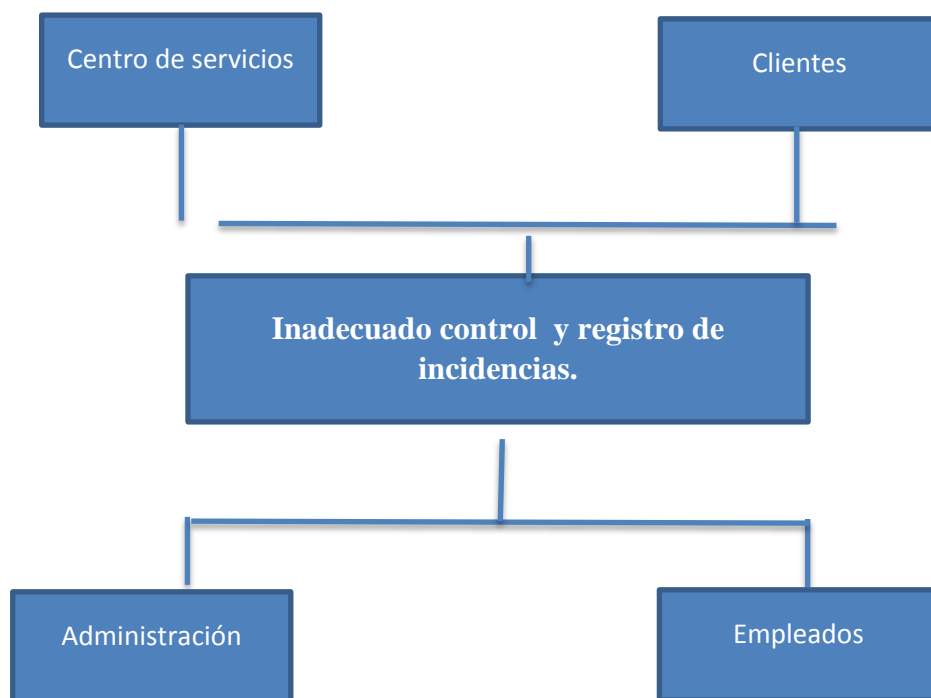
## Centro de servicios

Call Center: Su objetivo es gestionar un alto volumen de llamadas y redirigir a los usuarios, excepto en los casos más triviales, a otras instancias de soporte y/o comerciales.

Centro de Soporte (Help Desk): Su principal objetivo es ofrecer una primera línea de soporte técnico que permita resolver en el menor tiempo las interrupciones del servicio.

Centro de Servicios (Service Desk): es la encargada de ofrecer los servicios tanto a clientes como a los usuarios de TI, que brinda la organización. **SOPORTE DEL SERVICIO (SERVICE SUPPORT)** se centra en las tareas diarias de funcionamiento y soporte de los servicios de TI.

### 2.02 Mapeos de Involucrados



*Figura 1:* Mapeo de Involucrados: El análisis de involucrados es un instrumento esencial que permite: Identificar a aquellas personas interesadas en un determinado proyecto, establecer quienes pudieran ser afectadas por los objetivos del mismo, explorar quiénes y cómo pueden contribuir el logro de los objetivo.



### 2.03 Matriz de Análisis de Involucrados

Tiene como propósito analizar a los involucrados con su respectivo interés, los problemas percibidos, en el desarrollo de sus actividades sociales a través de los recursos que mantiene el proyecto para comprobar la efectividad de su producción.

Tabla 9

*Matriz de involucrados*

ACTORES INVOLUCRADOS	INTERESES EN EL PROBLEMA CENTRAL	PROBLEMAS PERCIBIDOS	RECURSOS Y CAPACIDADES	INTERESES SOBRE EL PROYECTO	CONFLICTOS POTENCIALES/CONSECUENCIAS DEL PROYECTO
<b>Operario</b>	Disminuir el tiempo de respuesta para incidentes.	Confusión con el tipo de incidencia	Recursos técnico, tecnológico, humano.	Optimizar el control de incidencias.	Cambio de políticas internas de la empresa sobre incidentes.
<b>Empleados</b>	El uso de la información de manera digital.	Procesos incorrectos en el control de incidencias.	Recursos humano.	Gestionar las incidencias a tiempo	No existe reporte de incidencias.
<b>Administrador</b>	Estabilizar las actividades de la empresa.	Administración y control incorrectos de las incidencias	Recursos técnico, tecnológico, humano	Administración y control de incidencias a tiempo.	Incorrecto uso del sistema informático en el control de incidencias.

Matriz de Análisis de Involucrados. En esta matriz detallamos más a fondo sobre la participación de los involucrados en el proyecto. Estableciendo el interés en el problema central quienes pudieran ser afectadas por los objetivos del mismo

### **Análisis de la matriz de involucrados**

En la matriz mostrada anteriormente se analizarán las estrategias que realizarán los diferentes tipos de actores mencionados en el mapeo de involucrados, buscando soluciones a los distintos problemas presentados para el desarrollo del proyecto.

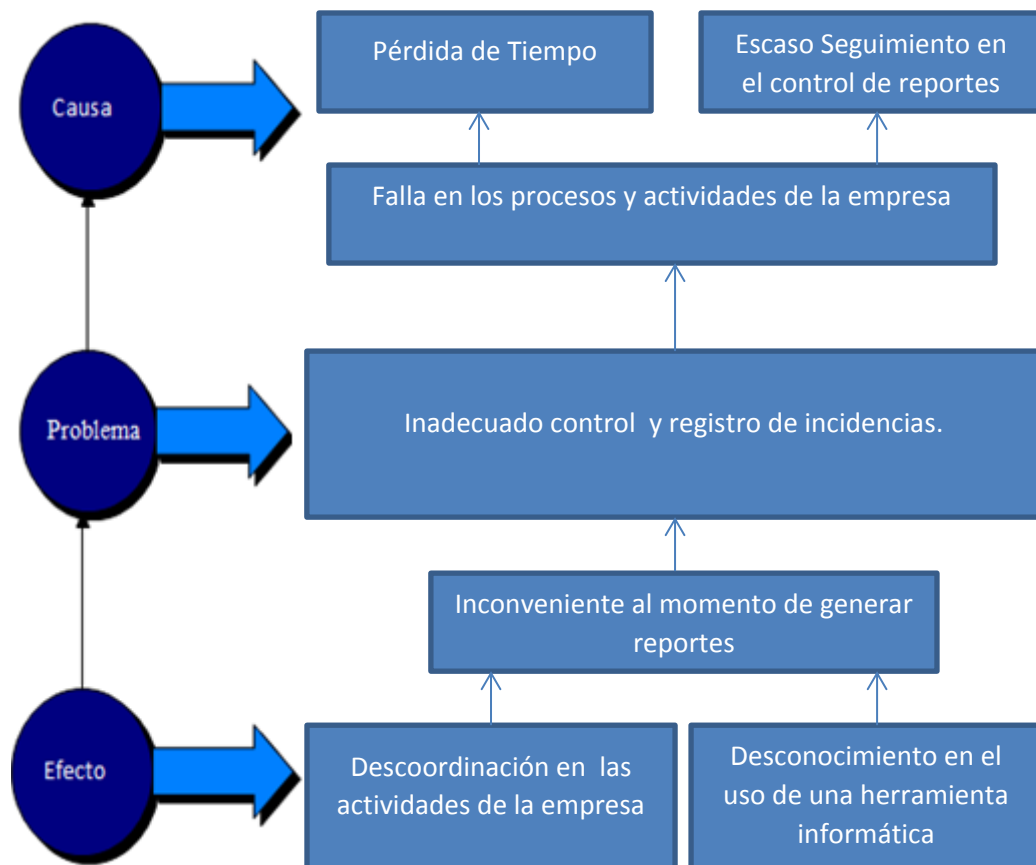
También se ha realizado el respectivo análisis de involucrados, los cuales muestran intereses sobre el problema central, generando diferentes interrogantes en cada uno de dichos involucrados permitiendo encontrar cuales son los puntos favorables y no favorables para su mejor desempeño dentro de la empresa, y así llegar a cumplir su objetivo, que es poder solucionar el problema central

Con los problemas observados y analizados mediante los involucrados, la solución que se considera en el presente proyecto, tiene como finalidad mejorar los procesos, análisis y control de incidencias, adicionalmente ayudando a al cliente en obtener una atención de calidad.

## Capítulo III: Árbol de Problemas - Objetivos

### 3.01 Árbol de Problemas

En el siguiente árbol de problemas se describe los aspectos necesarios que nos permiten entender de mejor manera la problemática a resolver, listando las causas y sus efectos para de esta manera poder identificar todo el conjunto de problemas sobre el cual se concentran enfocados los objetivos del proyecto.



*Figura 2:* Árbol de Problemas: Es importante porque nos ayuda a entender problemática a resolver. En él se expresan causa y efecto, las situaciones negativas observadas por los involucrados en relación con el problema en cuestión.

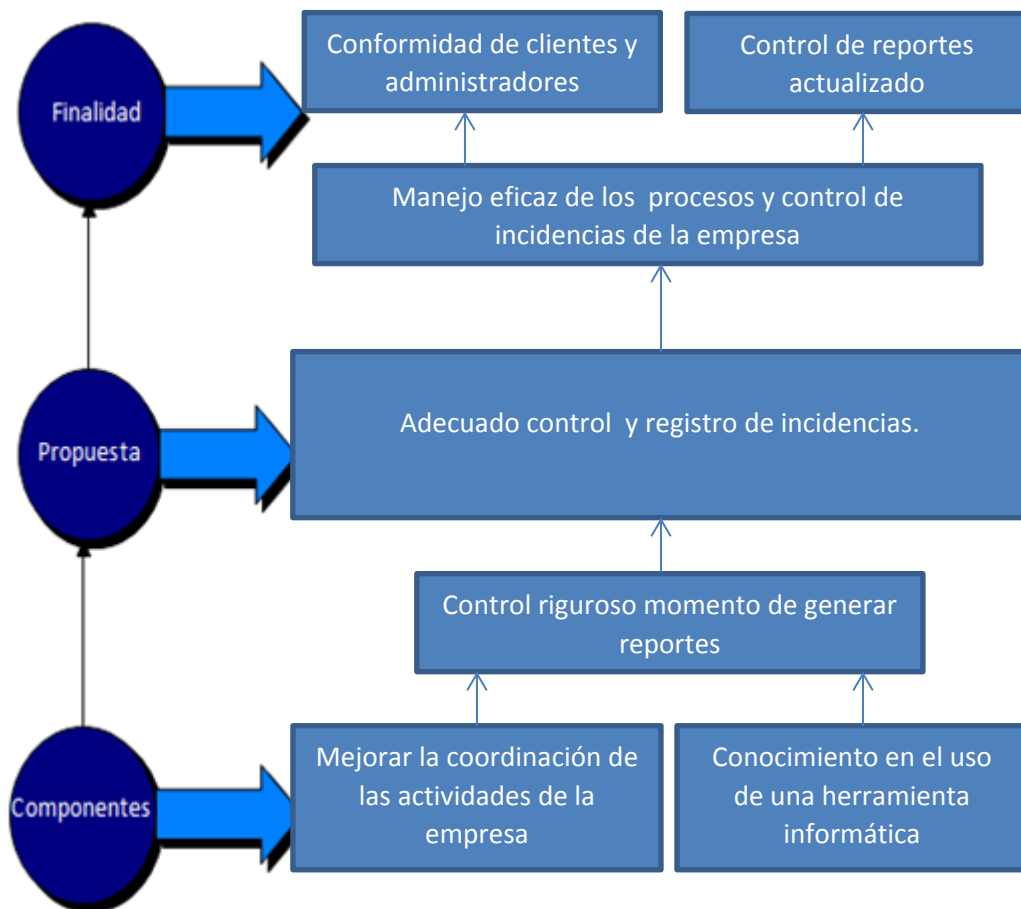
### **Análisis del árbol de problemas.**

El desconocimiento de nuevas herramientas tecnológicas, provoca la dificultad para dar un seguimiento a los requerimientos de un determinado cliente, así mismo el deficiente control de incidencias generan una defectuosa organización de los documentos e información de la empresa. Ocasionando que la atención de cada incidencia sea poca satisfactoria, incumpliendo con las expectativas del administrador, produciendo fallas en los principales procesos de la Empresa.

Los procesos mediante vía telefónica provocan una pérdida de tiempo tanto al cliente, como al administrador lo que ocasiona que disminuya la productividad. La pérdida de credibilidad en la gestión de cada proceso provoca un incremento de tiempo y disminución en la productividad, debido a la complejidad de los procesos y el excesivo papeleo. Los mismos que provocan un disgusto al administrador por la falta de organización en los documentos.

### 3.01 Árbol de Objetivos

A continuación se describe la solución a todos los inconvenientes identificados en el árbol de problemas, convirtiendo a los componentes en medios para encarar el problema de desarrollo identificado y proporcionar el sistema necesario para mejorar el proceso y control de incidencias de la empresa.



*Figura 3:* Árbol de Objetivos: Facilita la determinación de un objetivo del proyecto, como también permite; identificar los objetivos y alternativas para la realización del proyecto y describir la situación futura que será lograda cuando se resuelvan los problemas identificados anteriormente.

#### Análisis del árbol de objetivos.

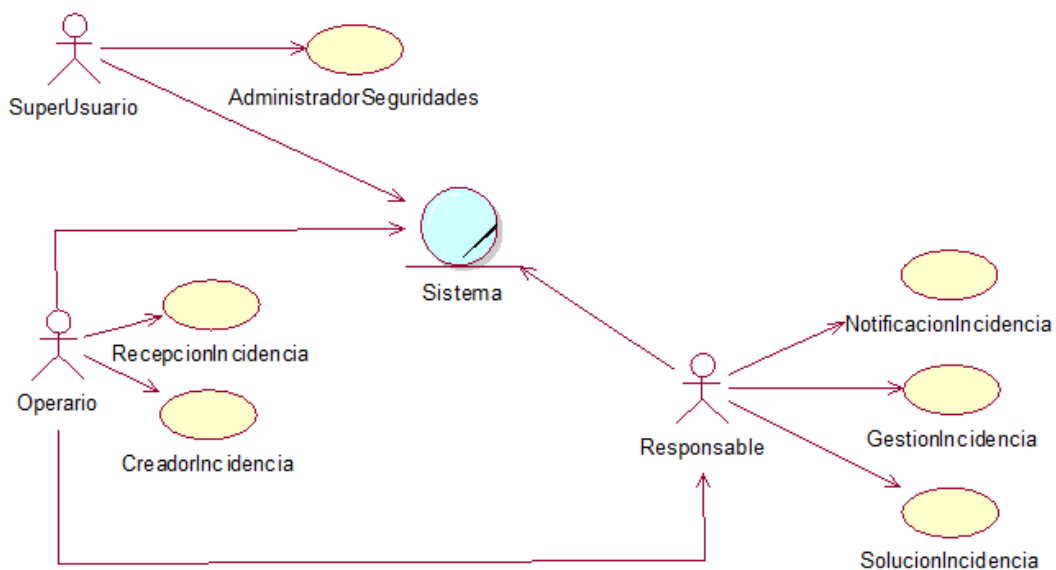
Se reduce el tiempo de respuesta, induce a la facilidad para el manejo de la información de cada incidencia, al igual que se genera los respectivos reportes e informes, fomentando una correcta organización en la empresa.

La satisfacción en los administradores y personal por la precisión en los procesos, se verá reflejada en el trabajo y desempeño de cada uno, gracias a la sencillez del manejo de información y el limitado papeleo; incrementando un control riguroso de la organización y manejo de la documentación.

### 3.03. Diagramas de casos de uso

Un diagrama de casos de uso es una especie de diagrama de comportamiento UML mejorado.

El Lenguaje de Modelado Unificado (UML), define una notación gráfica para representar casos de uso llamada modelo de casos de uso.



*Figura 4.* Diagrama de casos de uso. En esta figura detallamos. El control de incidencias, en el cual empieza solicitando el ingreso al sistema, el operario receptara y creara la incidencia. El responsable notificara, gestionara la incidencia.

### 3.04. Casos de uso de realización

#### 3.04.01 Diagrama de casos de uso de realización y especificaciones

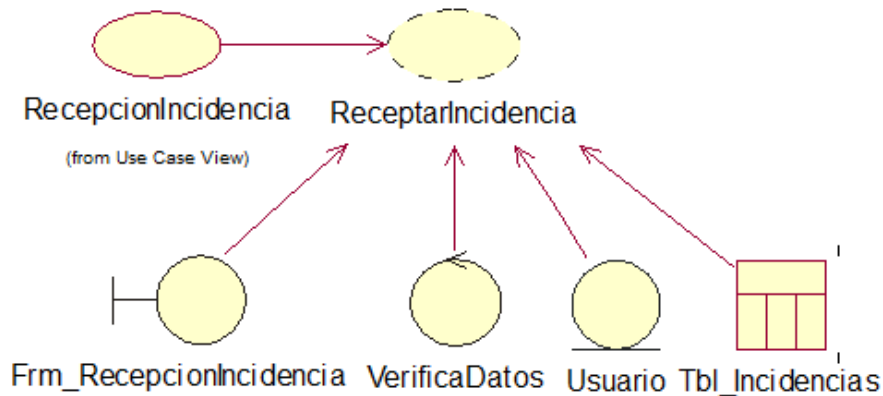


Figura 5. Diagrama de caso de uso de realización. Es esta figura detallamos la recepcion de incidencia, que se realiza en el caso de uso principal, el cual empieza en el formulario recepcionincidencia. Se verifican los datos, se ingresa la incidencia y se guarda en la tabla incidencias.

Tabla 10

*Especificación del caso de uso de realización. Recepcion de incidencia*

<b>NOMBRE</b>	Cliente
<b>IDENTIFICADOR</b>	CUR001
<b>Responsabilidades</b>	Operario
<b>Tipo</b>	Usuario
<b>Referencias de Casos de Uso</b>	Ninguno
<b>Referencias Requisitos</b>	Usado para la recepción de incidencias
<b>PRECONDICIONES</b>	
El operario debe haber recibido la incidencia	
<b>POSTCONDICIONES</b>	
El operario deberá confirmar la recepción de la incidencia	
<b>SALIDAS PANTALLA</b>	
El operario vera la incidencia requerida	

*Caso de uso de Realización. En esta tabla detallamos la recepción de incidencia, el cual permitirá la recepción de una nueva incidencia.*

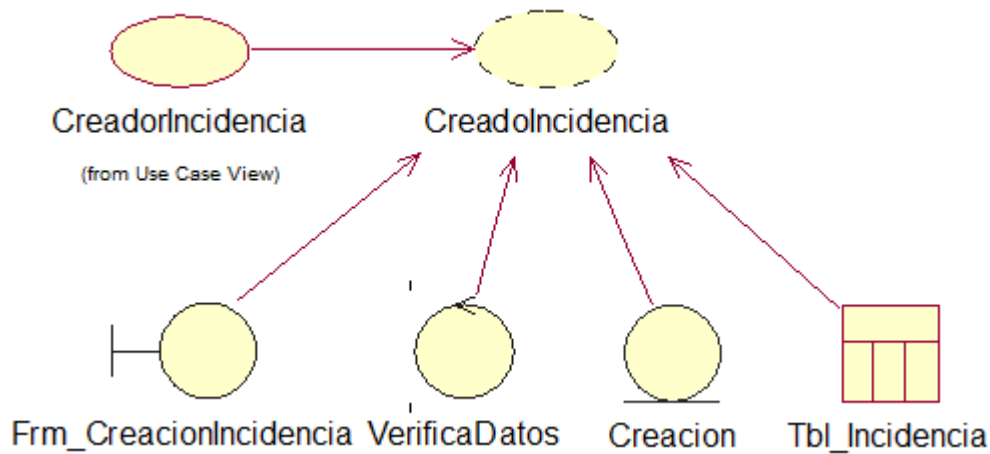


Figura 6. Diagrama de caso de uso de realización. Es esta figura detallamos la creación de una incidencia, que se realiza en el caso de uso principal, el cual empieza en el formulario creación incidencia. Se verifican los datos, se ingresa la incidencia y se guarda en la tabla incidencias.

Tabla 11

*Especificación del caso de uso de realización. Creación de incidencia*

<b>NOMBRE</b>	Cliente
<b>IDENTIFICADOR</b>	CUR002
<b>Responsabilidades</b>	Operario
<b>Tipo</b>	Usuario
<b>Referencias de Casos de Uso</b>	Ninguno
<b>Referencias Requisitos</b>	Usado para la creación de incidencias
<b>PRECONDICIONES</b>	
El operario deberá tener una incidencia que crear	
<b>POSTCONDICIONES</b>	
El operario deberá anotar detalladamente la incidencia	
<b>SALIDAS PANTALLA</b>	
El operario creara y revisara la incidencia	

*Caso de uso de Realización. En esta tabla detallamos la recepción de incidencia, el cual permitirá la creación de una nueva incidencia.*



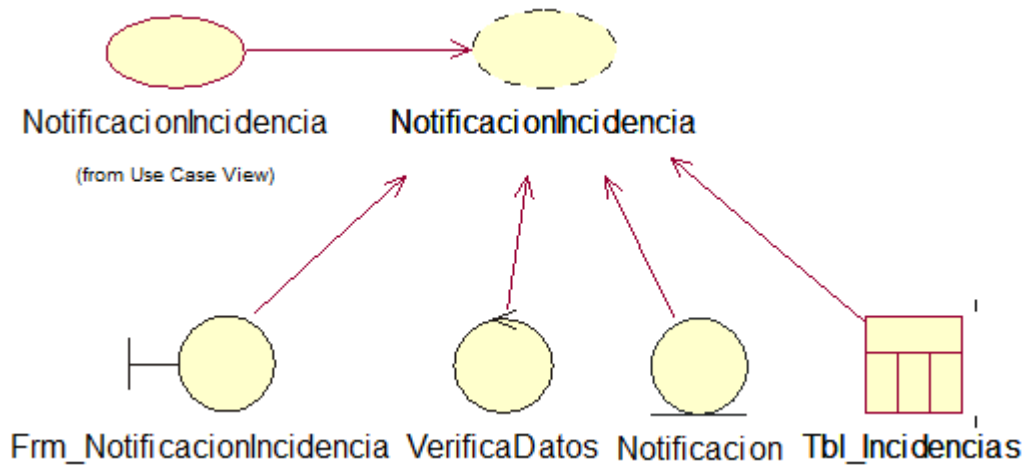


Figura 7. Diagrama de caso de uso de realización. Es esta figura detallamos la notificación de una incidencia, que se realiza en el caso de uso principal, el cual empieza en el formulario notificación incidencia. Se verifican los datos, se ingresa la incidencia y se guarda en la tabla incidencias.

Tabla 12

Especificación del caso de uso de realización. Notificación de incidencia

<b>NOMBRE</b>	Cliente
<b>IDENTIFICADOR</b>	CUR003
<b>Responsabilidades</b>	Responsable
<b>Tipo</b>	Usuario
<b>Referencias de Casos de Uso</b>	Ninguno
<b>Referencias Requisitos</b>	Usado para la recepción de la incidencia
<b>PRECONDICIONES</b>	
El Responsable deberá recibir una notificación por correo de que existe una incidencia	
<b>POSTCONDICIONES</b>	
El Responsable revisara la incidencia generada	
<b>SALIDAS PANTALLA</b>	
El Responsable podrá visualizar la incidencia	

Caso de uso de Realización. En esta tabla detallamos la recepción de incidencia, el cual permitirá la notificación de una nueva incidencia.

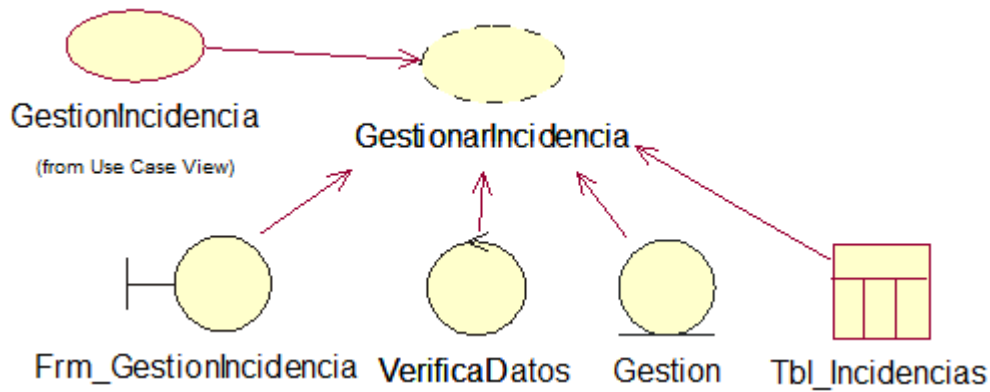


Figura 8. Diagrama de caso de uso de realización. Es esta figura detallamos la notificación de una incidencia, que se realiza en el caso de uso principal, el cual empieza en el formulario notificación incidencia. Se verifican los datos, se ingresa la incidencia y se guarda en la tabla incidencias.

Tabla 13

*Especificación del caso de uso de realización. Gestión de incidencia*

<b>NOMBRE</b>	Cliente
<b>IDENTIFICADOR</b>	CUR004
<b>Responsabilidades</b>	Responsable
<b>Tipo</b>	Usuario
<b>Referencias de Casos de Uso</b>	Ninguno
<b>Referencias Requisitos</b>	Usado para gestionar la incidencia
<b>PRECONDICIONES</b>	
El Responsable deberá tener una incidencia que gestionar	
<b>POSTCONDICIONES</b>	
El Responsable deberá proceder a gestionar la incidencia	
<b>SALIDAS PANTALLA</b>	
El Responsable podrá visualizar la descripción detallada de la incidencia	

*Caso de uso de Realización. En esta tabla detallamos la recepción de incidencia, el cual permitirá la gestión de una nueva incidencia.*

### 3.05. Diagrama de secuencias del sistema

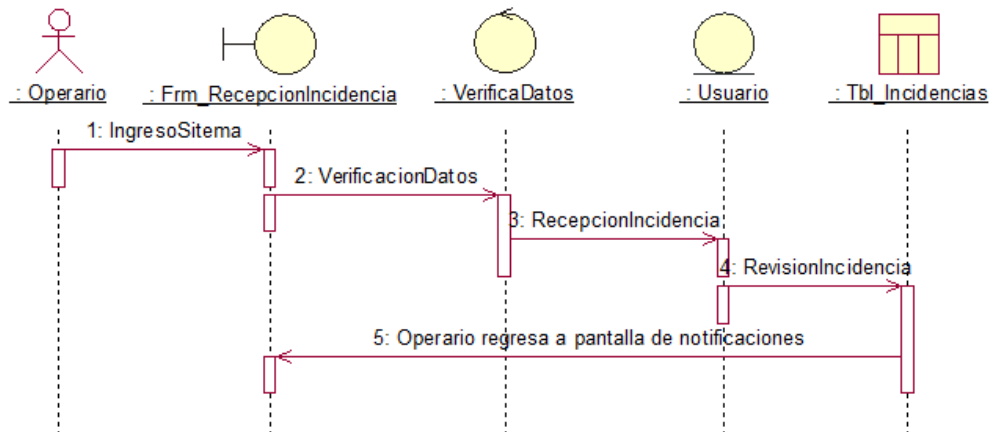


Figura 9. Diagrama de Secuencia. En esta Figura detallamos la recepción de una incidencia. El cual inicio con el ingreso al sistema, procede a la revisión de la incidencia y el operario regresa a la pantalla de notificaciones.

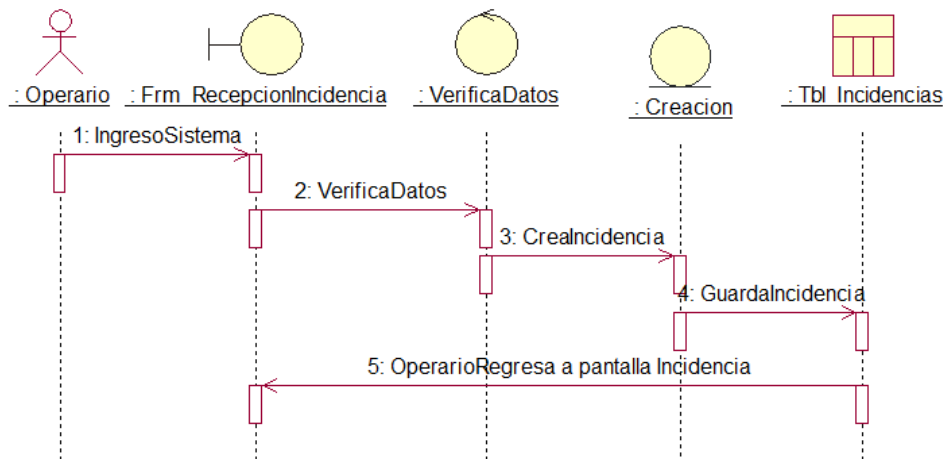


Figura 10. Diagrama de Secuencia. En esta Figura detallamos la creación de una incidencia. El cual inicio con el ingreso al sistema, verifica los datos, se procede a la creación de una incidencia si los datos son correctos y el operario regresa a la pantalla de incidencias.

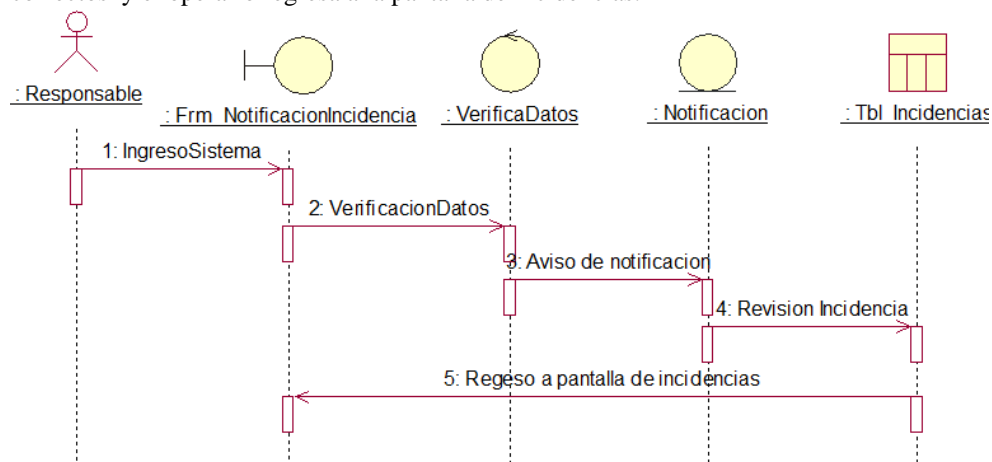


Figura 11. Diagrama de Secuencia. En esta Figura detallamos la notificación de una incidencia. El cual inicio con el ingreso al sistema, verifica los datos, aviso de una incidencia si los datos son correctos y el operario regresa a la pantalla de incidencias.

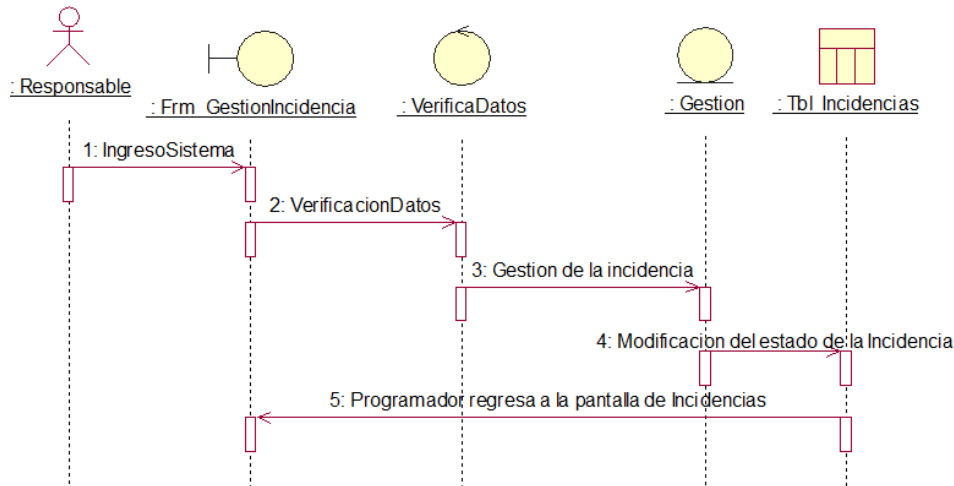


Figura 12. Diagrama de Secuencia. En esta Figura detallamos la gestión de una incidencia. El cual inicio con el ingreso al sistema, verifica los datos, gestión de una incidencia, realizada la modificación de la incidencia y el operario regresa a la pantalla de incidencias.

### 3.06. Especificación de los casos de uso

Tabla 14

Especificación de caso de uso. Recepar de incidencia

<b>CASOS DE USO</b>	Recepar Incidencia
<b>IDENTIFICADOR</b>	CU001

#### CURSO TIPICO DE EVENTOS

USUARIO	SISTEMA
Ingresar al sistema y verifica datos de recepción de incidencias	Tificación de incidencia Motivo y posible submotivo Fecha de gestión Prioridad Estado de gestión Concepto Observaciones

#### CURSOS ALTERNATIVOS

El operario debe haber recibido la incidencia

*Especificación del caso de uso. Recepar de incidencia. En esta tabla detallamos el caso de uso permite al operario recepar una incidencia.*

Tabla 15

Especificación de caso de uso. Crear incidencia

<b>CASOS DE USO</b>	Crear Incidencia
<b>IDENTIFICADOR</b>	CU002

**CURSO TÍPICO DE EVENTOS**

<b>USUARIO</b>	<b>SISTEMA</b>
	Tificación de incidencia
Verifica datos de creación de Incidencia	Motivo y posible submotivo
	Fecha de gestión
	Prioridad
	Estado de gestión
	Concepto
	Observaciones
	Guardar incidencia

**CURSOS ALTERNATIVOS**

Si no se guardo la Incidencia debe volver a ingresar datos de incidencia

---

*Especificación del caso de uso. Crear incidencia. En esta tabla detallamos el caso de uso permite al operario crear una incidencia.*

Tabla 16

Especificación de caso de uso. Notificación incidencia

<b>CASOS DE USO</b>	Notificación de Incidencia
<b>IDENTIFICADOR</b>	CU003

**CURSO TÍPICO DE EVENTOS**

<b>USUARIO</b>	<b>SISTEMA</b>
	Incendencia
El Responsable revisara la incidencia generada.	Responsable de resolver incidencia

**CURSOS ALTERNATIVOS**

Si no recibe notificación deberá revisar la incidencia generada.

---

*Especificación del caso de uso. Notificar incidencia. En esta tabla detallamos el caso de uso permite al operario notificar una incidencia.*

Tabla 17

Especificación de caso de uso. Gestionar incidencia

<b>CASOS DE USO</b>	<b>Gestionar Incidencia</b>
<b>IDENTIFICADOR</b>	CU004

**CURSO TIPO DE EVENTOS**

**Usuario** Sistema

El Responsable deberá proceder a gestionar la incidencia      Se dara solución a la Incidencia

**CURSOS ALTERNATIVOS**

Si no se tienes solución se deberá volver a verificar datos de gestión.

---

*Especificación del caso de uso. Gestionar incidencia. En esta tabla detallamos el caso de uso permite al operario gestionar una incidencia.*

## CAPÍTULO IV: ANÁLISIS DE ALTERNATIVAS

### 4.01. Matriz de Análisis de Alternativas

Tabla 18

*Matriz de Análisis de Alternativas*

Objetivos	Impact. del propósito	Factib. Técnica	Factib. Financ.	Factib. Social	Factib. política	Total	Catego ría
Solucionar necesidades, los cuales permitan elevar la calidad de solución a incidencias.	4	3	4	2	4	17	Alto
Automatizar y mejorar los procesos de gestión de incidencia.	4	4	4	3	2	17	Alto
Optimizar el proceso de registro de la información de incidencias solucionadas.	4	4	4	2	2	16	Alto
Conocimiento del estado en el que se encuentra la actividad, proceso o registro.	4	3	4	3	3	17	Alto
<b>TOTAL</b>	16	14	16	10	11	67	

Matriz de Análisis de Alternativas. En este análisis ponderamos cada una de las alternativas obtenidas del árbol de objetivos y puntuamos para obtener la prioridad de cada una.

0-5 =Bajo

6-10 =Medio Baja

11-15 = Media Alta

16- 20 = Alta

#### 4.01 Matriz de Análisis de Impactos de los Objetivos.

Matriz de Impactos de Objetivos. En este análisis ponderamos evaluar la factibilidad técnica, financiera, social y política de las estrategias representadas por los objetivos.

Véase anexo A.001

#### 4.02. Estándares para el Diseño de Clases

##### DIAGRAMA DE CLASES

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

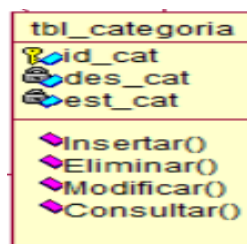
Un diagrama de clases está compuesto por los siguientes elementos:

- ✓ Clase: atributos, métodos y visibilidad.
- ✓ Relaciones: Herencia, Composición, Agregación, Asociación y Uso.

##### ELEMENTOS

**Clase.-**Es la unidad básica que encapsula toda la información de un objeto. A través de ella podemos modelar el entorno en estudio.

En UML, una clase es representada por un rectángulo que posee tres divisiones:




- ✓ **Superior:** Contiene el nombre de la Clase
- ✓ **Intermedio:** Contiene los atributos que caracterizan a la Clase
- ✓ **Inferior:** Contiene los métodos u operaciones, los cuales son la forma como interactúa el objeto con su entorno.




### ATRIBUTOS Y MÉTODOS:

**Atributos:** Los atributos o características de una Clase pueden ser de tres tipos, los que definen el grado de comunicación y visibilidad de ellos con el entorno, estos son:

- ✓ **public** (+, 

### Métodos:

Los métodos u operaciones de una clase son la forma en como ésta interactúa con su entorno, éstos pueden tener las características:

- ✓ **public** (+, 

### RELACIONES ENTRE CLASES:

Ahora ya definido el concepto de clase, es necesario explicar cómo se pueden interrelacionar dos o más clases. Antes es necesario explicar el concepto de cardinalidad de relaciones: En UML, la cardinalidad de las relaciones indica el grado y nivel de dependencia, se anotan en cada extremo de la relación y éstas pueden ser:

- ✓ **uno o muchos:** 1..\* (1..n)
- ✓ **0 o muchos:** 0..\* (0..n)
- ✓ **número fijo:** m (m denota el número).

#### 4.03. Diagrama de clases

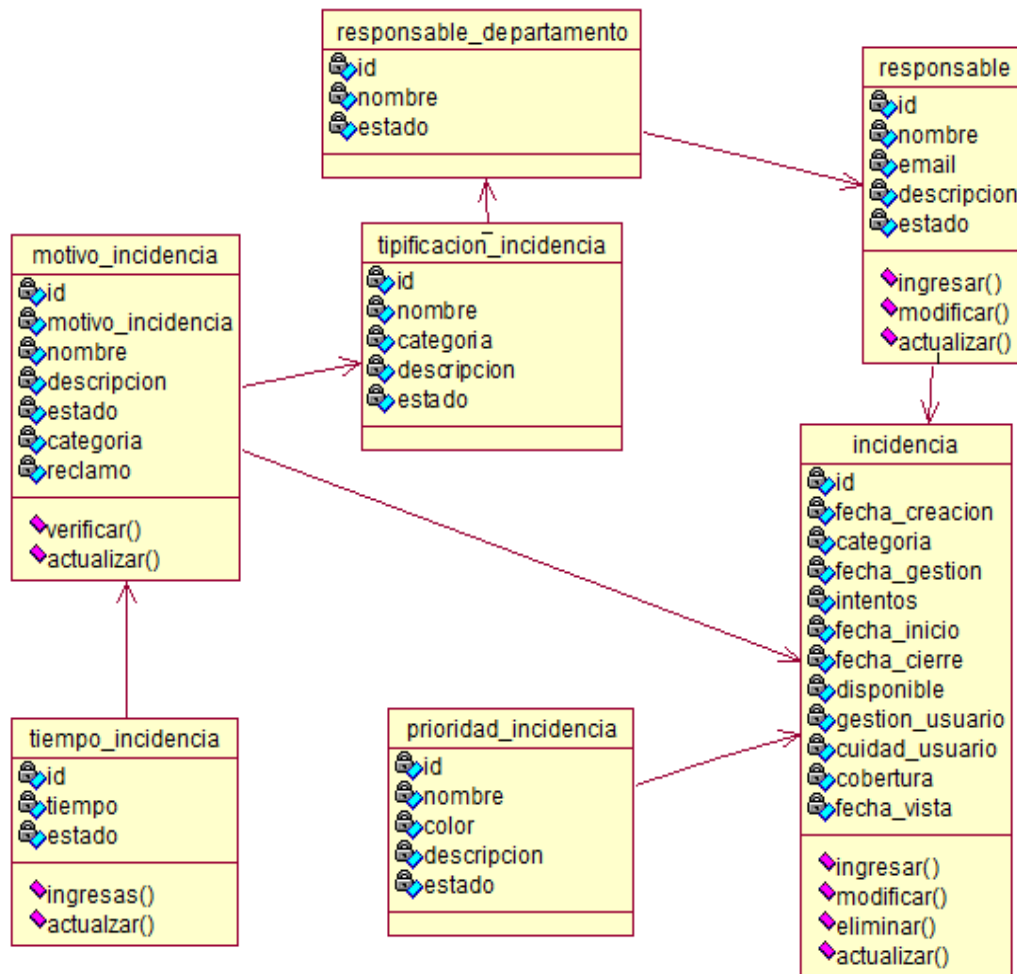


Figura 13. Diagrama de clases. En esta figura detallamos el conjunto de objetos que comparten los mismos atributos y relaciones. Podemos ver un conjunto de elementos estáticos..

#### 4.04. Modelo Lógico – Físico

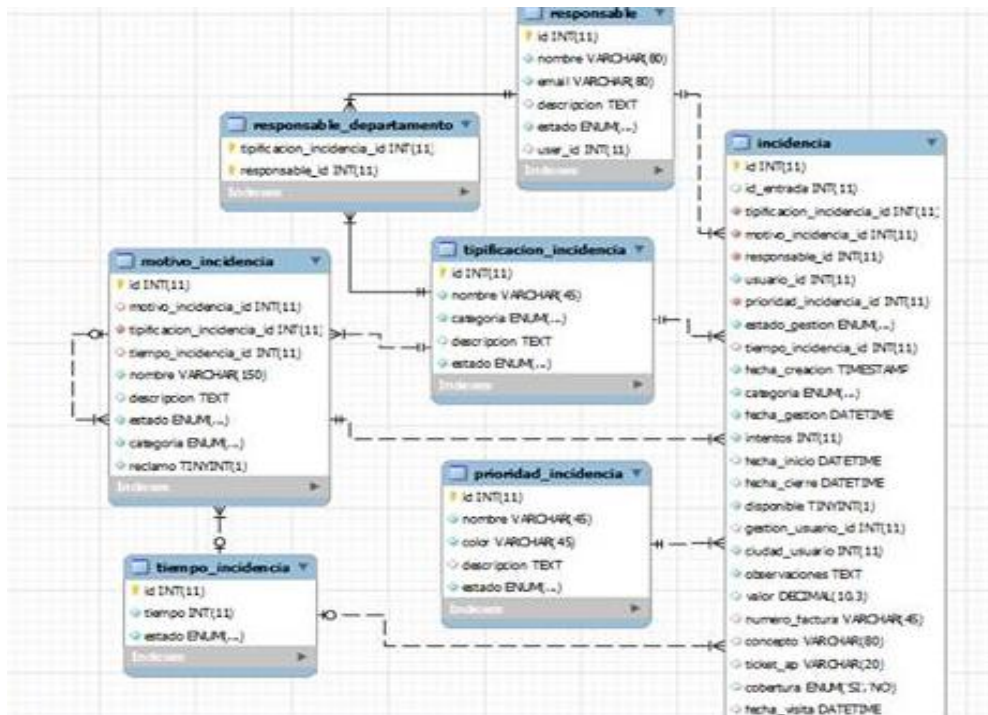


Figura 14. Diagrama de clases - Modelo Lógico. En este grafico detallamos el modelo lógico se usa en el UML, muestra las clases del proyecto con sus respectivos atributos y métodos, de manera lógica.

#### 4.05. Diagrama de Componentes

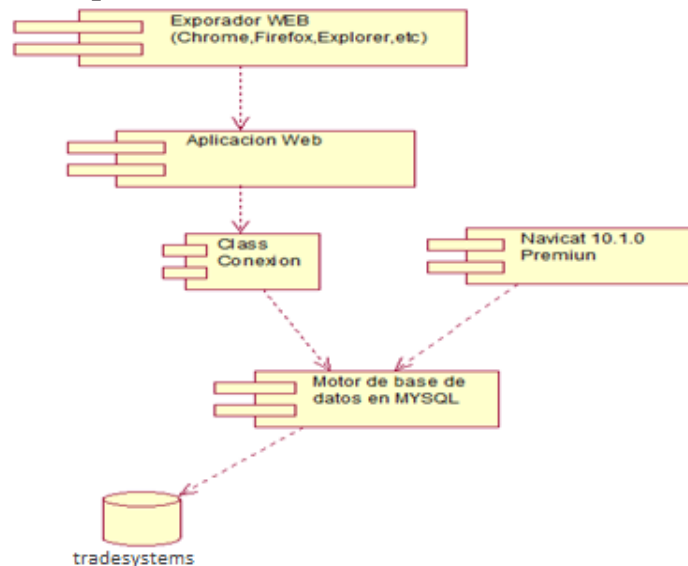


Figura 15: Diagrama de Componentes. En esta Figura se representa la capa acceso a datos

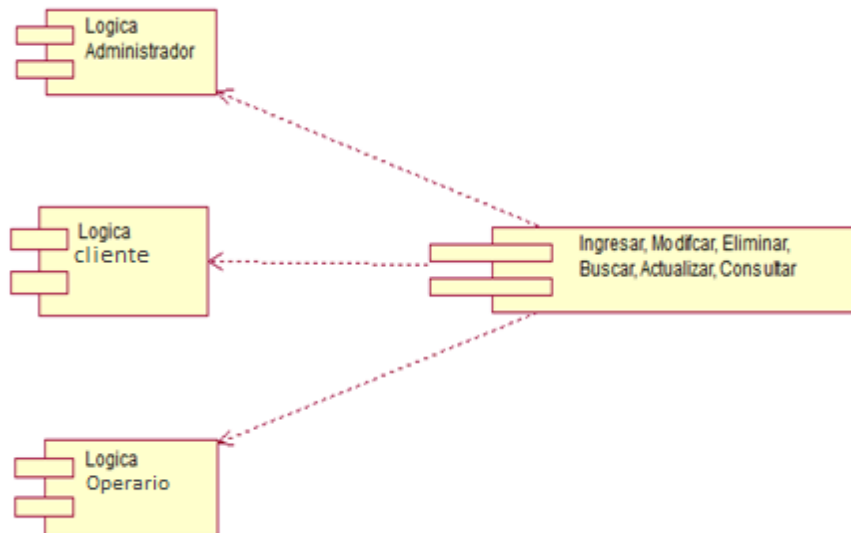


Figura 16: Diagrama de Componentes. En esta Figura se representa la capa de lógica de negocio

#### 4.6. Diagrama de estrategias

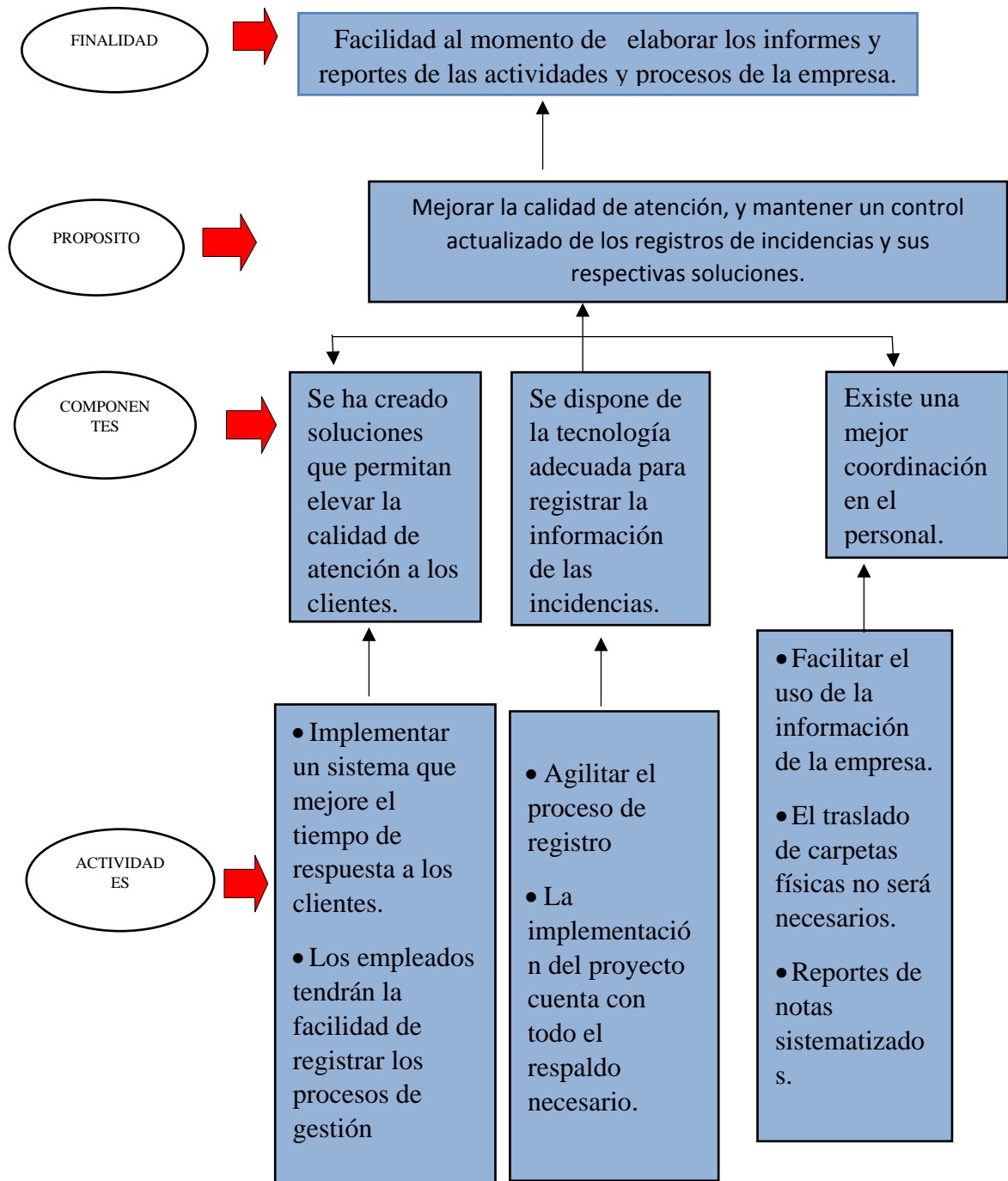


Figura 17: Diagrama de Estrategias. Este diagrama muestra los componentes, las actividades, y los objetivos específicos de dichas actividades.

#### 4.07. Matriz del Marco Logico

La matriz de Marco Lógico es una herramienta conceptual que ayuda en las etapas de diseño de proyectos, monitoreo, ejecución y evaluación. Véase anexo A.002

#### 4.08 Vistas Arquitectonicas

La arquitectura seleccionada para el desarrollo del sistema informatico de la Empresa “TRADESYSTEM” ha sido la arquitectura de tres niveles o arquitectura tres capas.

##### 4.08.01. Vista lógica

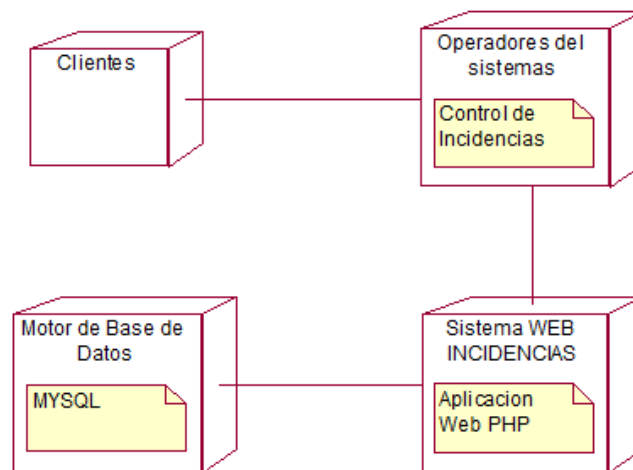


Figura 18. Vista Lógica. Este diagrama muestra los componentes de la vista lógica del sistema.

##### 4.08.02. Vista física

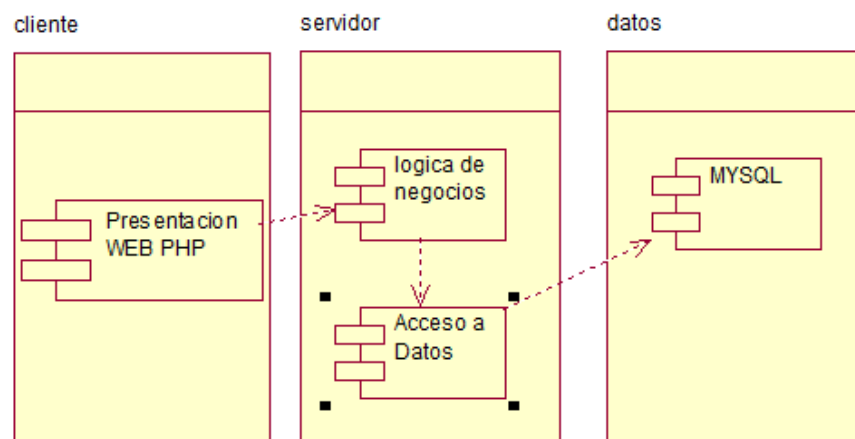
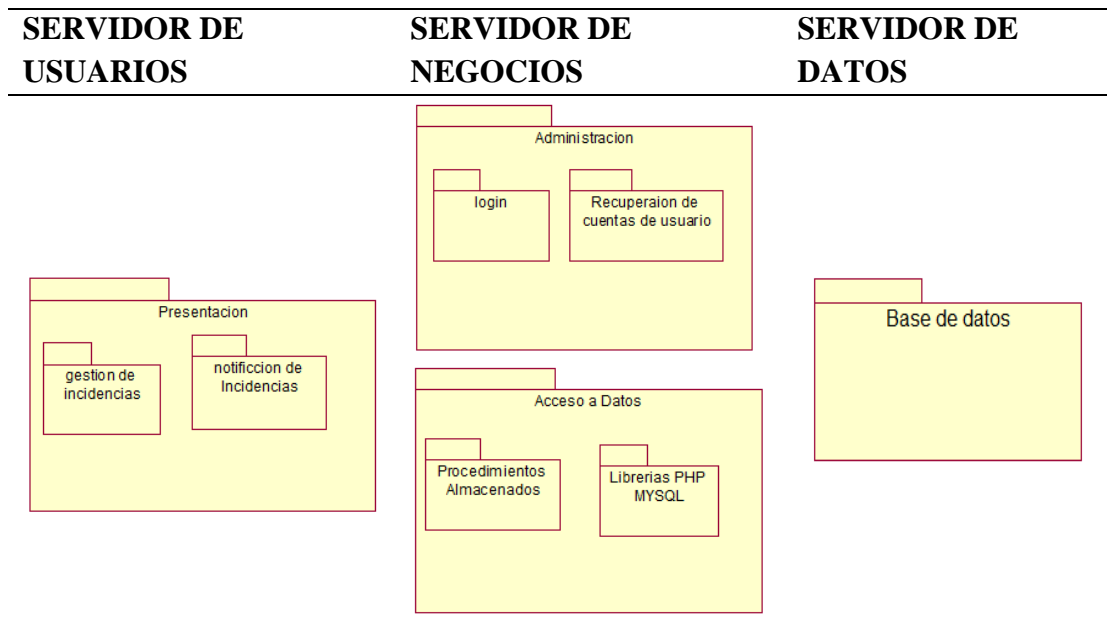


Figura 19. Vista Física. En esta Figura se detalla la arquitectura en tres capas del sistema informatico.

#### 4.08.03. Vista de desarrollo

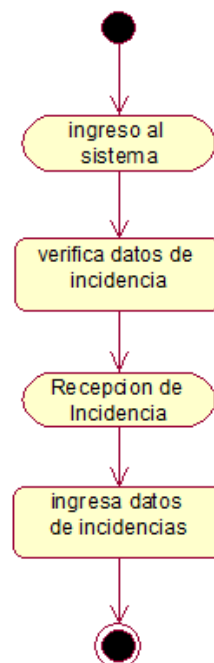
Tabla 19

*Diagrama de paquetes*



En esta tabla detallamos el diagrama de paquetes del sistema informatico

#### 4.09.04. Vista de procesos



*Figura 20. Diagrama de actividades en esta figura se detalla del proceso de recepción de incidencia. Muestra las actividades que se realizaran dentro del sistema para la ejecución de cada proceso.*

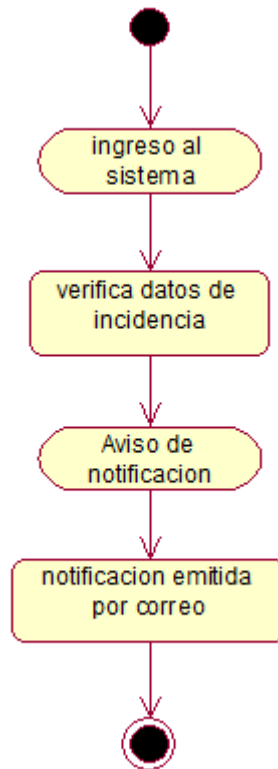


Figura 21. Diagrama de actividades de notificación de incidencia. Muestra las actividades que se realizarán dentro del sistema para la ejecución de cada proceso.

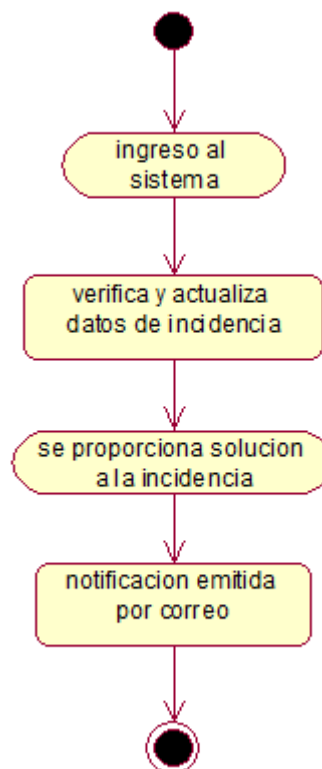


Figura 22. Diagrama de actividades En esta Figura se detalla el del proceso de gestión de incidencia. Muestra las actividades que se realizarán dentro del sistema para la ejecución de cada proceso.



## Capítulo V: Propuesta

### 5.01. Especificación de estándares de programación

PHP es un lenguaje de programación muy potente que, junto con html, permite crear sitios web dinámicos. Php se instala en el servidor y funciona con versiones de Apache, Microsoft IIs, Netscape Enterprise Server y otros.

#### Programación Orientada a Objetos

PHP tiene un conjunto muy completo de aspectos que facilitan la programación orientada a objetos (OOP) que incluye la habilidad de crear clases, clases abstractas, interfaces, herencia, constructores.

#### Clases

La definición básica de clases comienza con la palabra clave *class*, seguido por un nombre de clase, continuado por un par de llaves que encierran las definiciones de las propiedades y métodos pertenecientes a la clase.

El nombre de clase puede ser cualquier etiqueta válida, siempre que no sea una palabra reservada de PHP

Una clase puede tener sus propias constantes, variables (llamadas "propiedades"), y funciones (llamadas "métodos").

#### Constantes

Es posible definir valores constantes en función de cada clase manteniéndola invariable. Las constantes se diferencian de variables comunes en que no utilizan el símbolo \$ al declararlas o usarlas.

#### Variables

Las variables pertenecientes a clases se llaman "propiedades". También se les puede llamar usando otros términos como "atributos" o "campos", pero para los propósitos de esta referencia se va a utilizar "propiedades". Éstas se definen usando una de las palabras clave *public*, *protected*, o *private*, seguido de una declaración normal de variable.

## Instancia de una clase

Para crear una instancia de una clase, la palabra clave *new* debe ser usada. Un objeto siempre se creará a menos que el objeto tenga un constructor que arroje una excepción en caso de error. Las clases deberían ser definidas antes de la instanciación

## Constructor

Permite a los desarrolladores declarar métodos constructores para las clases. Aquellas que tengan un método constructor lo invocarán en cada nuevo objeto creado, lo que lo hace idóneo para cualquier inicialización que el objeto pueda necesitar antes de ser usado.

## 5.02. Diseño de Interfaces de Usuario

Ingreso al Sistema: el usuario realiza un ingreso de datos al login para ingresar al sistema.



*Figura 23.* Restricción de Acceso. El usuario al momento de ingresar a la pantalla principal no podrá visualizar los módulos sin antes registrarse. Ya que el sistema le muestra una pantalla de error el mismo permitirá realizar sus funciones determinadas y con sus debidas restricciones de usuario.



*Figura 24.* Búsqueda de cliente. El usuario al momento de ingresar podrá realizar la búsqueda de un cliente para crear la incidencia.

### Crear Cliente TradeSystem

Codigo Cliente *	<input type="text"/>	R.Social	<input type="text"/>
Ruc	<input type="text"/>	Direccion Titular	<input type="text"/>
Telefono Titular	<input type="text"/>	Telefono2 Titular	<input type="text"/>
Grupo Clientes	<input type="text"/>	Mail	<input type="text"/>
Nombre Contacto	<input type="text"/>	Apellido Contacto	<input type="text"/>
Email Contacto	<input type="text"/>		

Seleccione Usuario

Usuario

Figura 25. Creación de cliente. El usuario al momento de ingresar podrá realizar la creación de un nuevo cliente, los campos con \* de color rojo son obligatorios.

### Administrar Motivos de Incidencia

Negocio	Tipificación	Motivo	Tiempo	Estado	
<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>	
REPSOLGAS	Soporte Técnico	Telefonia		Activo	
REPSOLGAS	Soporte Técnico	Equipos informáticos	48	Activo	
REPSOLGAS	Pedidos	Equipos informáticos	48	Activo	
REPSOLGAS	Pedidos	Materiales	48	Activo	
REPSOLGAS	Pedidos	Partes y Piezas	48	Activo	
REPSOLGAS	Pedidos	Componentes de Telefonía	48	Activo	
REPSOLGAS	Mantenimiento Preventivo	Frecuencia Programada	48	Activo	
REPSOLGAS	Mantenimiento Preventivo	No programado	48	Activo	
REPSOLGAS	Servicios Técnicos	Gestión de Garantía	48	Activo	
REPSOLGAS	Servicios Técnicos	Inspecciones	48	Activo	

← 1 2 3 →

Figura 26. Administrar motivos de incidencia. El usuario al momento de ingresar podrá realizar modificación y crecion de motivos de incidencia.

### Administrar Prioridades de Incidencia

Nombre	Estado	Color	
<input type="text" value=""/>	<input type="text" value=""/>		
Alto	Activo		
Bajo	Activo		
Medio	Activo		
Urgente	Activo		

Figura 27. Administrar prioridad de incidencia. El usuario al momento de ingresar podrá realizar modificación y crecion de motivos de incidencia.

**Administrar Responsables**

Nombre	Email	Estado	Usuario
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Andres Barreno	owk666@gmail.com	Activo	andres.barreno
Christian Villareal	owk666@gmail.com	Activo	christian.villareal
Gabriel Mejía	owk666@gmail.com	Activo	gabriel.mejia
Giovanni Granda	owk666@gmail.com	Activo	giovanni.granda
Kevin Hidalgo	owk666@gmail.com	Activo	kevin.hidalgo
Mauricio Arcentales	owk666@gmail.com	Activo	mauricio.arcentales
Roberto Vozmediano	owk666@gmail.com	Activo	roberto.vozmediano
Roy Chavez	owk666@gmail.com	Activo	roy.chavez
Santiago Benitez	owk666@gmail.com	Activo	santiago.benitez
Sonia Quipo	owk666@gmail.com	Activo	sonia.quipo

Figura 28. Administrar responsable de incidencia. El usuario al momento de ingresar podrá realizar modificación y crecion de responsables de incidencia.

### 5.03. Especificación de pruebas de unidad

Tabla 20

#### Especificación de pruebas de unidad

<b>IDENTIFICADOR DE LA PRUEBA</b>	PUC001
<b>MÉTODO A PROBAR</b>	Inicio de sesión
<b>Objetivo de la Prueba</b>	Verificar que ingresen correctamente datos en la pantalla de inicio de sesión
<b>Datos de Entrada</b>	
Nombre de usuario o email	
Contraseña	
<b>Resultado Esperado</b>	
El sistema verifique que los campos sean ingresados correctamente	
Si se encuentran vacios, emitirá un mensaje de que los campos son obligatorios.	
Si los datos son incorrectos, emitirá un mensaje de "el usuario no fue hallado"	
<b>Comentarios</b>	
El proceso funciona al 100% al momento de ejecutarlo	

*Pruebas de unidad. En esta tabla detallamos el procedimiento de la realización de las pruebas de unidad. Las pruebas unitarias desarrolladas en este procedimiento tienen como objetivo validar el ingreso al sistema*

Tabla 21

*Especificación de pruebas de unidad*

<b>IDENTIFICADOR DE LA PRUEBA</b>	PUC002
<b>MÉTODO A PROBAR</b>	Metodo "Nuevo"
<b>Objetivo de la Prueba</b>	Crear nuevo cliente correctamente
<b>Datos de Entrada</b>	
<ul style="list-style-type: none"> <li>✓ Código cliente</li> <li>✓ Razón Social , Ruc</li> <li>✓ Dirección Titular</li> <li>✓ Teléfono Titular , Teléfono 2 Titular</li> <li>✓ Grupo clientes</li> <li>✓ Email</li> <li>✓ Nombre Contacto, Apellido Contacto</li> <li>✓ Email Contacto</li> </ul>	
<b>Resultado Esperado</b>	
<p>El sistema verifique que los campos sean ingresados correctamente</p> <p>Si se encuentran vacios, emitirá un mensaje de que los campos son obligatorios.</p> <p>Si los datos son incorrectos, emitirá un mensaje de "el usuario no fue hallado"</p>	
<b>Comentarios</b>	
El proceso funciona al 100% al momento de ejecutarlo	

*Pruebas de unidad. En esta tabla detallamos el procedimiento de la realización de las pruebas de unidad. Las pruebas unitarias desarrolladas en este procedimiento tienen como objetivo validar el método nuevo.*

Tabla 22

*Especificación de pruebas de unidad*

<b>IDENTIFICADOR DE LA PRUEBA</b>	PUC003
<b>MÉTODO A PROBAR</b>	Metodo "Nuevo"
<b>Objetivo de la Prueba</b>	Crear nueva incidencia correctamente
<b>Datos de Entrada</b>	
<ul style="list-style-type: none"> <li>✓ Tipificación</li> <li>✓ Responsable</li> <li>✓ Motivo , submotivo</li> <li>✓ Fecha de gestión , estado de gestión</li> <li>✓ Prioridad</li> </ul>	
<b>Resultado Esperado</b>	
<p>El sistema verifique que los campos sean ingresados correctamente</p> <p>Si se encuentran vacios, emitirá un mensaje de que los campos son obligatorios.</p> <p>Si los datos son incorrectos, emitirá un mensaje de "el usuario no fue hallado"</p>	
<b>Comentarios</b>	
El proceso funciona al 100% al momento de ejecutarlo	

*Pruebas de unidad. En esta tabla detallamos el procedimiento de la realización de las pruebas de unidad. Las pruebas unitarias desarrolladas en este procedimiento tienen como objetivo validar el método nuevo.*

Tabla 23

*Especificación de pruebas de unidad*

<b>IDENTIFICADOR DE LA PRUEBA</b>	PUC004
<b>MÉTODO A PROBAR</b>	Metodo "Modificar"
<b>Objetivo de la Prueba</b>	Gestionar responsables correctamente
<b>Datos de Entrada</b>	
✓ Nombre	
✓ Mail	
✓ Estado	
✓ Usuario	
<b>Resultado Esperado</b>	
El sistema debe realizar una búsqueda de los responsables.	
El sistema realiza una búsqueda completa en la base de datos.	
<b>Comentarios</b>	
El proceso funciona al 100% al momento de ejecutarlo	

*Pruebas de unidad. En esta tabla detallamos el procedimiento de la realización de las pruebas de unidad. Las pruebas unitarias desarrolladas en este procedimiento tienen como objetivo validar el método modificar.*

**5.04. Especificación de pruebas de aceptación**

Tabla 24

*Especificación de pruebas de aceptación*

<b>IDENTIFICADOR DE LA PRUEBA</b>	PA001
<b>CASO DE USO</b>	Perfiles de usuario
<b>Tipo de Usuario</b>	Administrador
<b>Objetivo de la Prueba</b>	Registrar los usuarios que operan el sistema
<b>Secuencia de eventos</b>	
Ingrese datos del empleado	
Asignar rol	
Asignar login y password	
<b>Resultados Esperados</b>	
Registrar a usuario	
Brindar acceso a un usuario al sistema	
<b>Estado</b>	
Aceptado	

*Especificación de pruebas de aceptación. Perfiles de usuario. En esta tabla detallamos la prueba de aceptación en un escenario de utilización del sistema y el comportamiento que de él se espera, desde la perspectiva del cliente, usuario o sistema externo que interactúa con el programa*

Tabla 25

*Especificación de pruebas de aceptación*

<b>IDENTIFICADOR DE LA PRUEBA</b>	PA002
<b>CASO DE USO</b>	Creación de incidencia
<b>Tipo de Usuario</b>	Administrador
<b>Objetivo de la Prueba</b>	El técnico deberá crear la incidencia a solucionar
<b>Secuencia de eventos</b>	El operario deberá anotar detalladamente la incidencia
<b>Resultados Esperados</b>	La creación de la incidencia o problema de cliente
<b>Estado</b>	Aceptado

*Especificación de pruebas de aceptación. Perfiles de usuario. En esta tabla detallamos la prueba de aceptación en un escenario de utilización del sistema y el comportamiento que de él se espera, desde la perspectiva del cliente, usuario o sistema externo que interactúa con el programa*

Tabla 26

*Especificación de pruebas de aceptación*

<b>IDENTIFICADOR DE LA PRUEBA</b>	PA002
<b>CASO DE USO</b>	Perfiles de usuario
<b>Tipo de Usuario</b>	Técnico
<b>Objetivo de la Prueba</b>	El técnico deberá asignar un responsable para la incidencia
<b>Secuencia de eventos</b>	La incidencia deberá tener un responsable a cargo
<b>Resultados Esperados</b>	Asignar un responsable a una incidencia
<b>Estado</b>	Aceptado

*Especificación de pruebas de aceptación. Perfiles de usuario. En esta tabla detallamos la prueba de aceptación en un escenario de utilización del sistema y el comportamiento que de él se espera, desde la perspectiva del cliente, usuario o sistema externo que interactúa con el programa*

## 5.05. Especificación de pruebas de carga

Tabla 27

### Pruebas de carga del sistema

<b>IDENTIFICADOR DE LA PRUEBA</b>	PC001
<b>Tipo de Prueba</b>	Carga de usuarios
<b>Objetivo de la Prueba</b>	Enviar 30 sesiones concurrentes 40 veces
<b>Descripción</b>	
<i>Herramienta:</i>	LoadUIWeb 2
<i>Descripción:</i>	Se hace la simulación de un tráfico de peticiones de 10 sesiones 20 veces
<b>Resultados Esperados</b>	
	Que el sistema soporte el tráfico de usuarios
<b>Comentarios</b>	
	El sistema soporte la entrada n números de usuarios

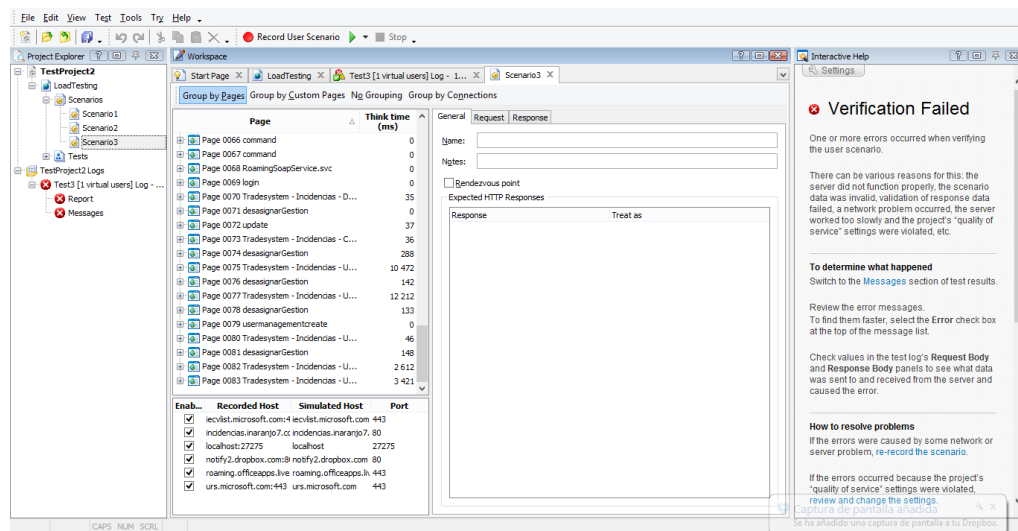


Figura 29. Pruebas de carga. En esta figura los formularios utilizados en las pruebas de carga.



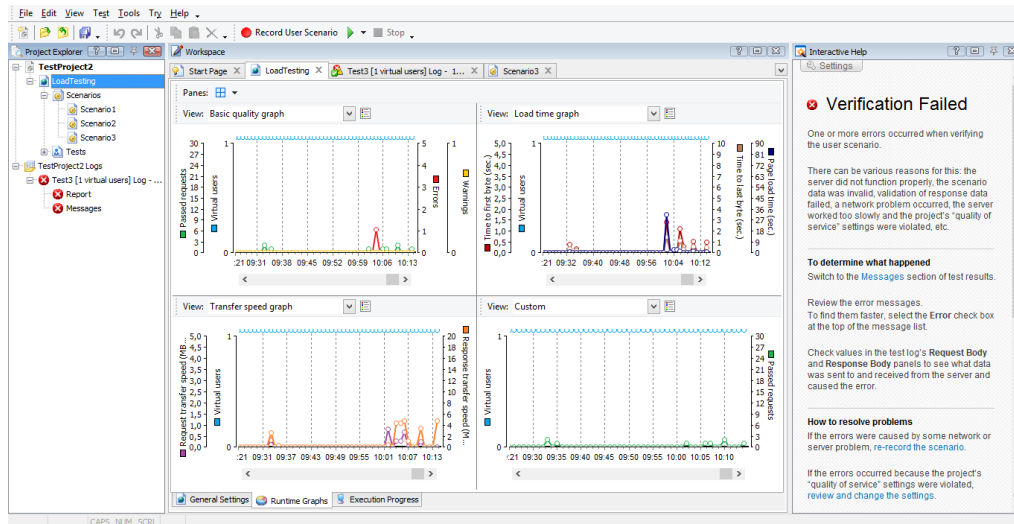


Figura 30 Pruebas de carga. En esta figura muestra el resultado de las pruebas de carga.

## 5.06. Configuración del Ambiente mínima/ideal

### 5.06.01 Rendimiento

El sistema está optimizado para que obtener una respuesta rápida. La interfaz de usuario se diseñó con el objetivo de facilitar y optimizar el tiempo en los procesos.

### 5.06.02 Características Técnicas

Funciona como Servicio Web desde cualquier servidor que soporte MySQL y PHP.

### 5.06.03 Componentes del Sistema

El sistema está constituido por los módulos de administración, actividades, catalogo de pedidos, catalogo de incidencias.

### 5.06.04 Requerimientos de Hardware y Software

Los requerimientos mínimos recomendados, tanto para servidores como para estaciones de trabajo.

La resolución recomendada para la utilización es 1024x760 pixeles. Una resolución menor puede provocar una visualización incorrecta.

#### **5.06.04.01 Requerimientos de Hardware**

##### **EQUIPO SERVIDOR**

**Procesador** Core Duo 3 GHz .

**RAM** 2 GB mínimo

**Disco duro** 80 Gb espacio libre

##### **EQUIPO CLIENTE**

**Procesador** Core Duo 1+ GHz

**RAM** 1 Gb mínimo

**Disco duro** 60 Gb espacio libre

#### **5.06.04.02 Requerimientos de Software**

##### **EQUIPO SERVIDOR**

**Sistema operativo** Windows 7-8

**Manejadores de base de datos** MySQL 5.0 -

**Browser** Google Chrome, Internet Explorer

##### **EQUIPO CLIENTE**

**Sistema operativo** Windows 7-8

**Manejadores de base de datos** MySQL 5.0 -

**Browser** Google Chrome, Internet Explorer

## CAPITULO VI

### 6.01. Recursos

Tabla 28

*Recursos del proyecto*

TIPO	DESCRIPCION
<b>Físicos</b>	Equipo de Computo Impresora
<b>Económicos</b>	Alimentos-Refrigerios Transporte
<b>Renovables</b>	Hojas Energía Anillados Empastados
<b>Didáctico</b>	Internet
<b>Otros</b>	Programa de profesionalización

*En esta tabla detallamos los recursos utilizados para la ejecución de este proyecto.*

### 6.02 Presupuesto

Tabla 29

*Presupuesto*

CANTIDAD	DESCRIPCIÓN	VALOR UNITARIO	TOTAL
540	Impresiones Color	\$ 0,25	\$ 135,00
540	Impresiones B/N	\$ 0,10	\$ 54,00
1	Materiales de Oficina	\$ 20,00	\$ 20,00
<b>320</b>	Transporte	\$ 0,25	\$ 80,00
<b>2</b>	Anillados	\$ 2,50	\$ 5,00
<b>2</b>	Empastados	\$ 3,50	\$ 7,00
<b>6</b>	Internet	\$ 21,00	\$ 126,00
<b>1</b>	Seminarios	\$ 720,00	\$ 720,00
<b>1</b>	Tutorías	\$ 200,00	\$ 200,00
	<b>TOTAL</b>		<b>\$ 1.347,00</b>

*En esta tabla detallamos el presupuesto para el proyecto.*

### 6.3 Cronograma de Actividades

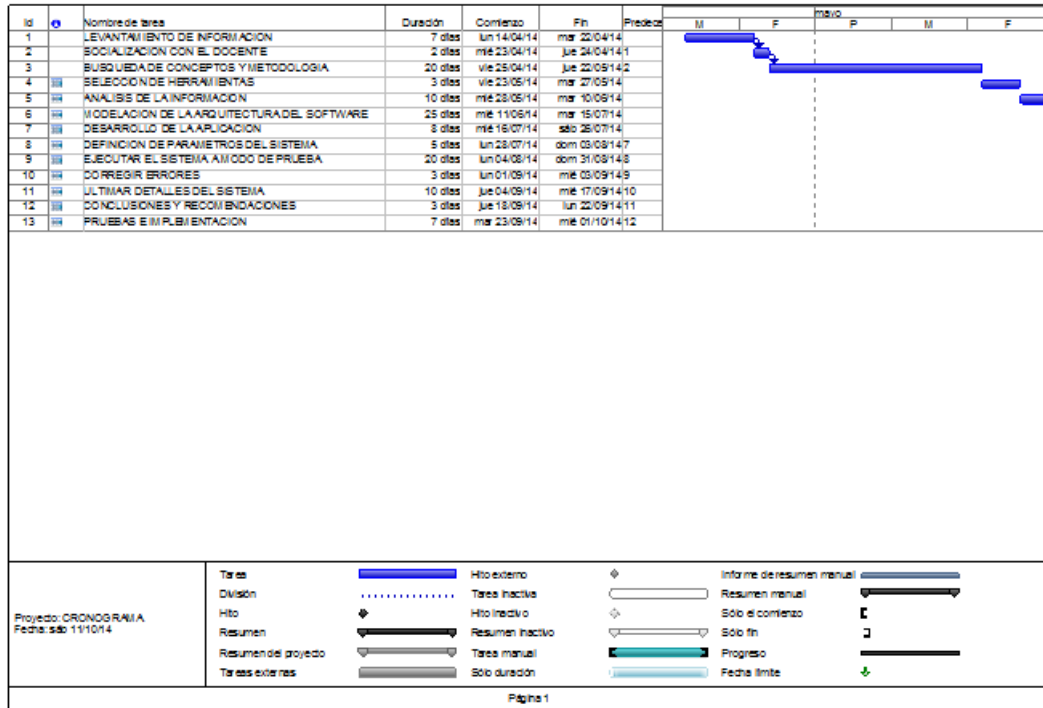


Figura 31. Cronograma de actividades. En esta figura se detalla la descripción y fechas de las actividades

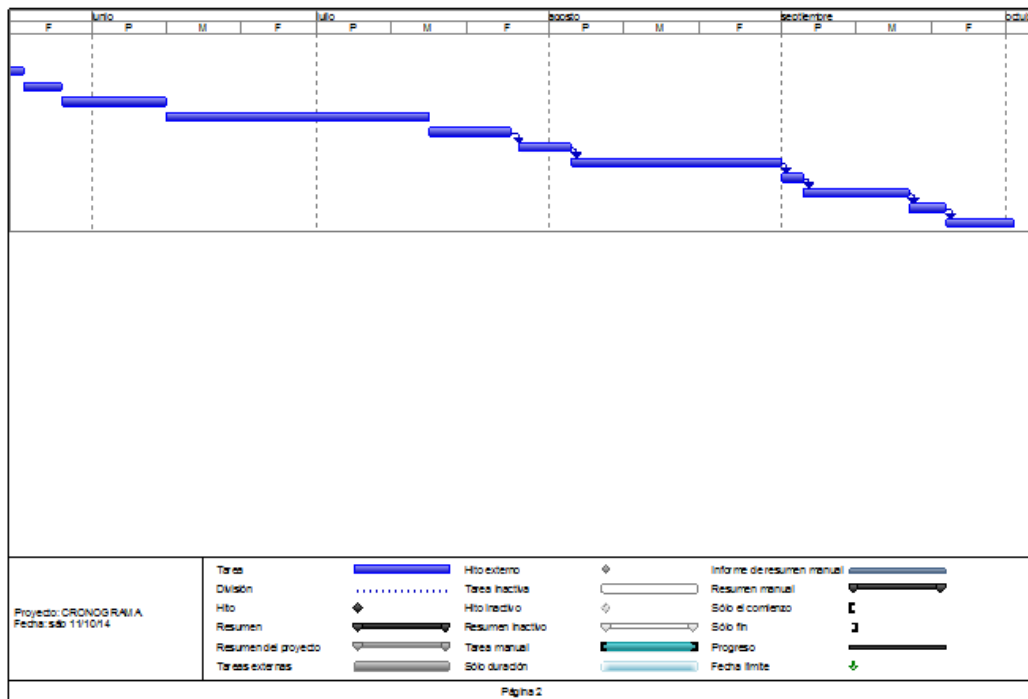


Figura 32 Cronograma de actividades. En esta figura se detalla fechas de las actividades y diagram de Gantt

## Capítulo VII: Conclusiones y Recomendaciones

### 7.01 Conclusiones

- Con la implementación del sistema lograremos la optimización los procesos de la empresa a través de los módulos con los que cuenta el sistema los cuales permiten el funcionamiento de la misma.
- Los administradores tendrán la opción de realizar la búsqueda de la información que genera el sistema y agiliza los procesos operacionales que se realizan en la actualidad, aportando nuevas estrategias en el desarrollo de las actividades.
- El sistema permitirá un control de cada cliente, incidencia y responsable para facilitar la información con eficiencia y rapidez.
- Esta Empresa contara con el aporte tecnológico beneficiando a sus clientes y empleados.

### 7.02 Recomendaciones

- El personal y los administradores tiene la obligación de revisar los manuales para el buen desempeño del sistema.
- Los errores que se presenten al momento de la manipulación del sistema se deberá reportar al personal técnico para la solución inmediata.
- Es necesario que los equipos tecnológicos que contaran el personal debe estar actualizados con la última tecnología para el desempeño de sus funciones.
- El acceso a los mantenimientos del sistema deben ser exclusivos del administrador, con el propósito de mantener la persistencia y la integridad de los datos, permitiendo que la seguridad del software se encuentre en un estado óptimo.



## ANEXOS

# ANEXOS

## A.01 Matriz del Marco Logico

RESUMEN NARRATIVO	INDICADORES	MEDIOS DE VERIFICACIÓN	SUPUESTOS
<b>Finalidad</b>  Facilidad al momento de elaborar los informes y reportes de las actividades y procesos de la empresa.	En el tiempo de uso del sistema se espera que incrementen un 60% la satisfacción de las nuevas estrategias de registro y control.	Reportes del historial de incidencias	Los administradores y el personal, no tendrán la dificultad en la generación de informes y reportes.
<b>Propósito</b>  Mejorar la calidad de atención, y mantener un control actualizado de los registros de incidencias y sus respectivas soluciones.	Aumento de un 70% en el manejo eficaz en los procesos y actividades de la empresa. - El número de demora en la atención se reduce en (+/- 10 minutos)	Historial de incidencias	Se logra optimizar el tiempo de actividades y procesos de gestión y registros.

<b>Componentes</b>	Los registros y el control de incidencias aumentaran un 75% con el proceso de automatización de todas las actividades. El acceso a la información aumentara 90% con la implementación del sistema tecnológico.	Informes sobre el estado en que se encuentra la empresa y el tipo de atención que realizan a los clientes. Consultas sobre el interés que tiene el personal sobre la tecnología a implementar.	Los administradores y el personal optaron con nuevos conocimientos y están dispuestos a utilizar la tecnología mediante la cual van a realizar sus labores con la dedicación de brindar comodidad a todos los usuarios.
Se ha creado soluciones que permitan elevar la calidad de atención a los clientes.			
Se dispone de la tecnología adecuada para registrar la información de las incidencias.			
Existe una mejor coordinación en el personal.			
<b>Actividades</b>			
Implementar un sistema que mejore el tiempo de respuesta a los clientes.	Aumento de un 95% en la coordinación de los procesos y actividades de la empresa.	-Se confirman en los informes de los procesos que se realizan en la empresa. Informe sobre los equipos de desarrollo a utilizar.	El personal se encuentra 100% capacitado para ofrecer los servicios con las nuevas estrategias en sus procesos de control.
Facilitar el uso de la información de la empresa.		Verificar módulo de seguridad.	Personal altamente capacitado para el manejo operativo.
Agilizar el proceso de registro.	<b>Presupuesto</b>		
Los empleados tendrán la facilidad de registrar los procesos de gestión.	Materiales	Verificar módulo mantenimiento y módulo de reglas de negocio.	Beneficiarios directos satisfactorios con la aplicación proporcionada.
Los empleados podrán tener acceso a la información desde cualquier lugar.	Impresiones Color (\$ 135,00)		
El traslado de carpetas físicas no será necesarios.	Impresiones B/N (\$ 54,00)	Cuantificar las entrevistas realizadas	Agilidad y seguridad en los procesos operativos de la empresa.
El personal obtendrá los conocimientos necesarios sobre el uso de la herramienta.	Materiales de Oficina (\$ 20,00)	Información disponible para la realización del sistema por parte de los empleados de la empresa.	Incremento de ingresos para la empresa.
La implementación del proyecto cuenta con todo el respaldo	Alimentos-Refrigerios (\$ 180,00)		
	Transporte (\$ 80,00)		
	Anillados (\$ 5,00)		
	Empastados (\$ 7,00)		
	Internet (\$ 126,00)		





---

necesario.	Taller
	(\$ 520,00)
	Tutorías
	\$ 200,00
	Total
	Presupuesto
	(\$ 1327.00)

---

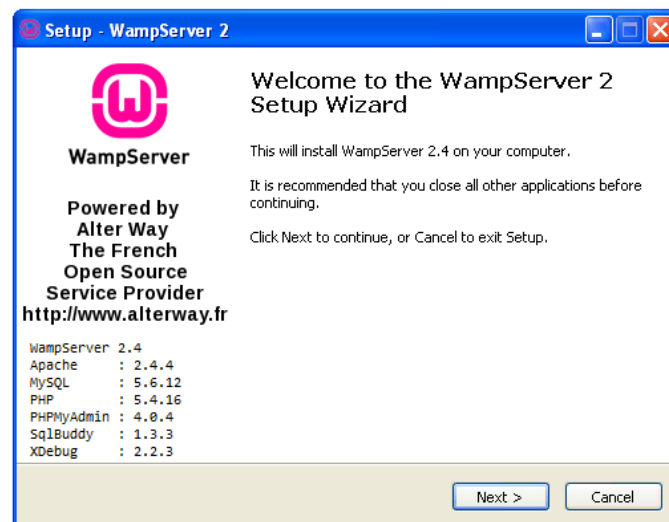
## A.02 Matriz de impactos de objetivos

77AUMENTO DE LA PRODUCTIVIDAD Y CONFORMIDAD EN LAS ACTIVIDADES Y PROCESOS, DEBIDO A LA AGILIDAD DE PEDIDOS Y CONTROL DE INVENTARIOS.	<i>Factibilidad de Lograse</i> (Alta-Media-Baja) ( 4 - 2 - 1 )	<i>Relevancia</i> (Alta-Media-Baja) ( 4 - 2 - 1 )	<i>Sostenibilidad</i> (Alta-Media-Baja) ( 4 - 2 - 1 )	<i>Total</i>
	<ul style="list-style-type: none"> <li>-Los beneficios son mayores que los costos.</li> <li>- Cuenta con financiamiento por parte de la empresa.</li> <li>-Protege el uso de los recursos.</li> <li>-Es aceptable y conveniente para los beneficiarios</li> <li>-Existe tecnología adecuada para su realización</li> <li>-Se cuenta con soporte político-institucional.</li> <li>-Mejora el entorno social con los clientes.</li> </ul> <p><b>28 Puntos</b></p>	<ul style="list-style-type: none"> <li>-Responde a las expectativas de los beneficiarios</li> <li>-Es una prioridad sentida por los beneficiarios</li> <li>-Beneficia a grupos de mayor carencia y vulnerabilidad</li> <li>-Los beneficios son deseados por los beneficiarios</li> </ul> <p><b>16 puntos</b></p>	<ul style="list-style-type: none"> <li>-Fortalece la participación de los empleados.</li> <li>-Fortalece la Organización local</li> <li>-Los beneficiarios están interesados en el proyecto</li> <li>-La Empresa respalda el financiamiento del proyecto a futuro</li> </ul> <p><b>16 puntos</b></p>	<p><b>60 puntos</b></p> <p>22 a 32 Baja</p> <p>33 a 44 Media baja</p> <p>45 a 66 Media Alta</p> <p>67 a 88 Alta</p>

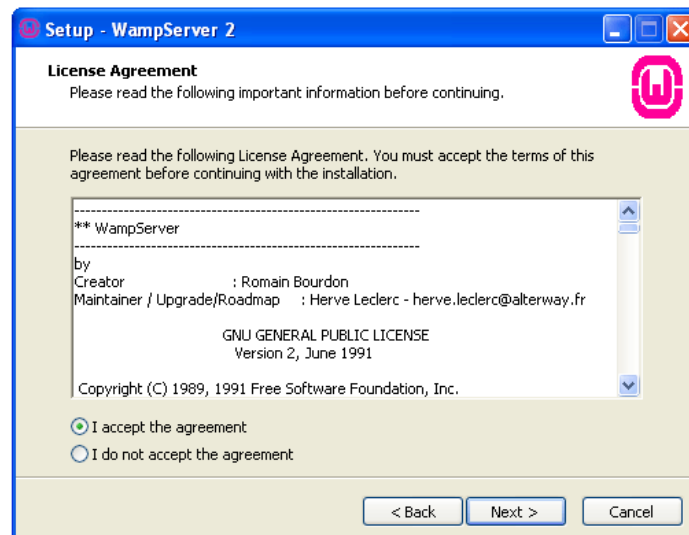
## A. 03 MANUAL DE INSTALACIÓN

### Instalación de WAMPSEVER

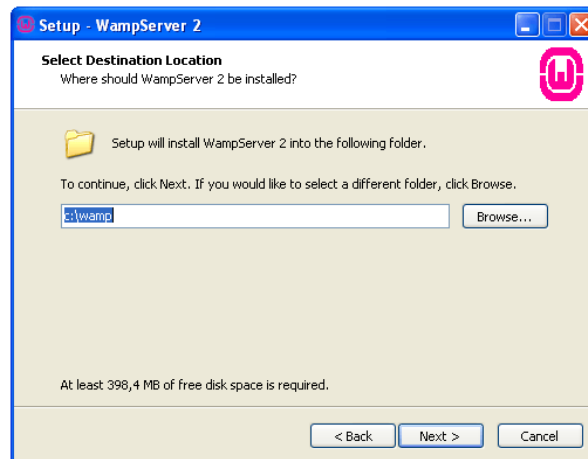
Lo ejecutamos con doble click sobre él. Nos debería aparecer una ventana de asistente de instalación muy parecida a esta:



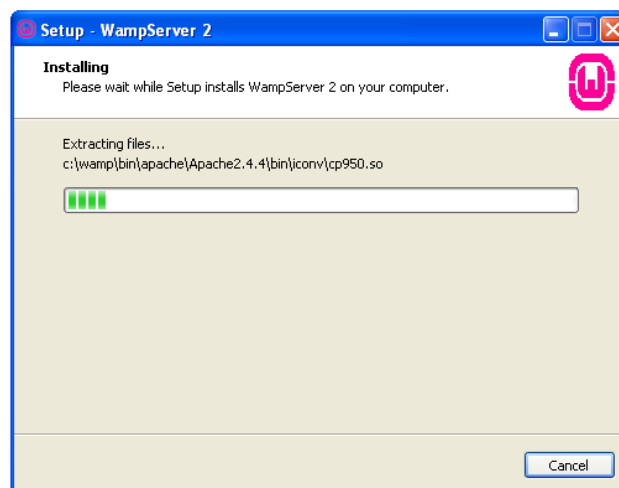
Le damos al boton “Siguiente” para avanzar al siguiente paso y aceptamos el contrato de licencia:



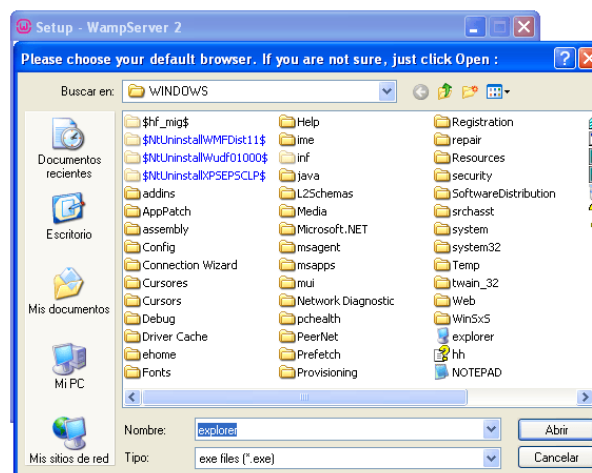
Ahora seleccionamos la ruta por defecto donde se instalara WAMPSEVER y pulsamos “Siguiente” para avanzar:



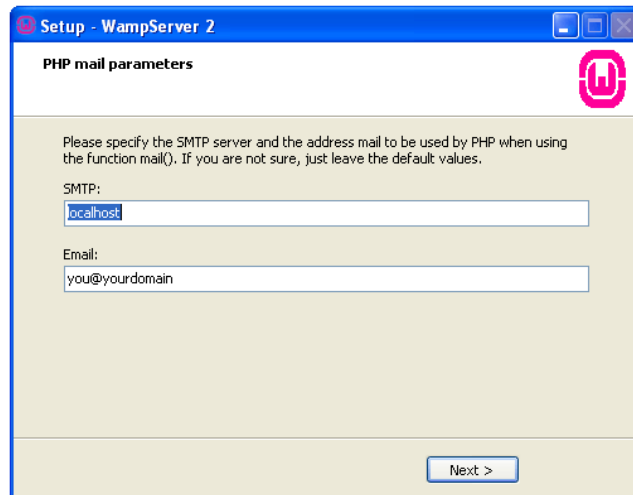
Seleccionamos si queremos o no queremos accesos directos y por ultimo comienza la instalación:



A media instalación nos pedirá que seleccionemos nuestro Navegador predeterminado, basta con hacer doble click sobre el acceso directo del navegador.



La instalación finalizará después de elegir el navegador predeterminado para visualizar los sitios web que se ejecuten sobre el stack WAMP y al final nos dará la opción de configurar el servidor SMTP que usaremos para que nuestras web hospedadas en el servidor que acabamos de instalar envíen correo, esto es opcional.



La instalación habrá acabado al darle a “Siguiente” y después a “Finalizar”. Ahora sabremos que nuestro WAMP está encendido por el icono de la barra de tareas:



Este icono que aparece en la imagen anterior es el pilar fundamental de WAMP, si haces click derecho sobre él accederás a unos menús, si haces click izquierdo, podrás acceder a otros menús diferentes, en todo caso, es el panel de control de WAMP.

### **Instalación de NetBeans en Windows**

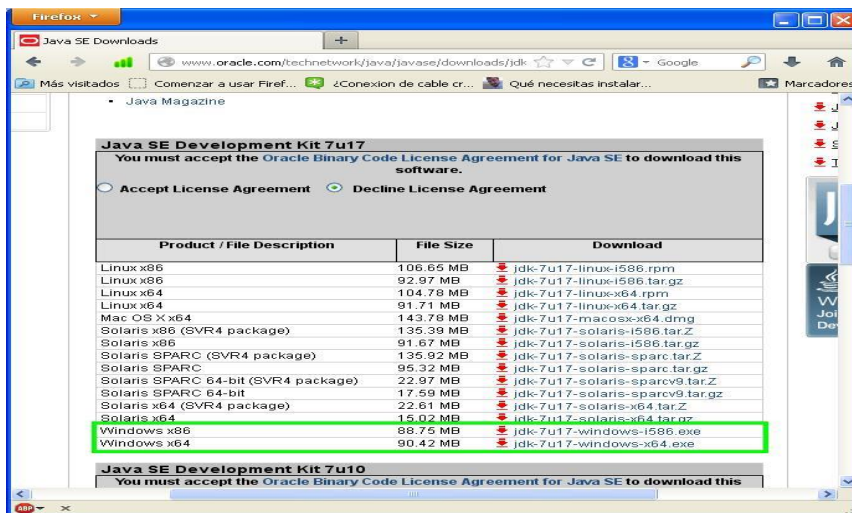
Hoy os enseñaré a instalar NetBeans para programar en C en Windows, durante la creación de la entrada lo instalaré en un Windows XP (en Windows 7 está probado y funciona de maravilla).

### **Descargar e instalar Java JDK 7.**

Para ello lo descargamos del siguiente enlace:

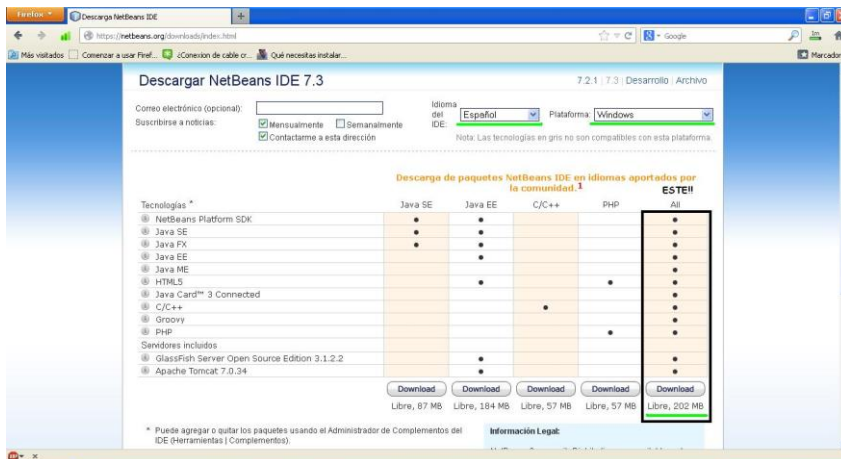
<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html?ssSourceSiteId=otnes>

Nos interesan las señaladas en un cuadro verde (dependiendo de si el sistema operativo es de 32 o 64 bits)



## Descarga e instalar NetBeans.

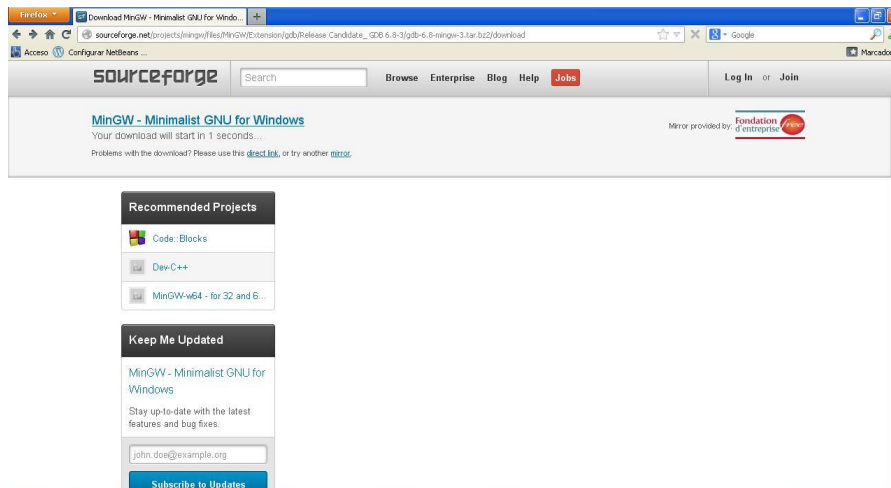
En este paso nos dirigimos al siguiente enlace: <https://netbeans.org/downloads/index.html>



## Descarga el depurador (Debugger)

En el siguiente enlace os descargais el depurador de MinGW, es muy importante que lo descargues:

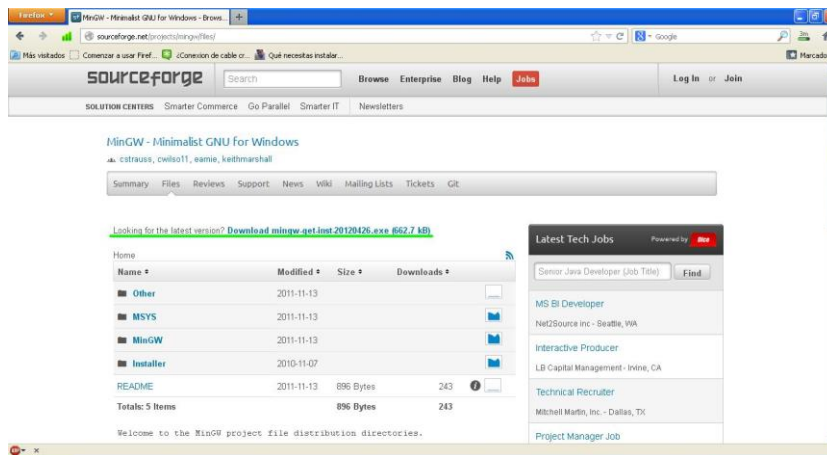
[http://sourceforge.net/project/downloading.php?group\\_id=2435&filename=gdb-6.8-mingw-3.tar.bz2&a=66334146](http://sourceforge.net/project/downloading.php?group_id=2435&filename=gdb-6.8-mingw-3.tar.bz2&a=66334146)



## Descarga e instalación del compilador.

Aquí empieza lo complicado, hay varios compiladores dependiendo de en que vallamos a programar pero nos centraremos en 2 de los mas conocidos para C y C++, estos son Cygwin y MinGW.

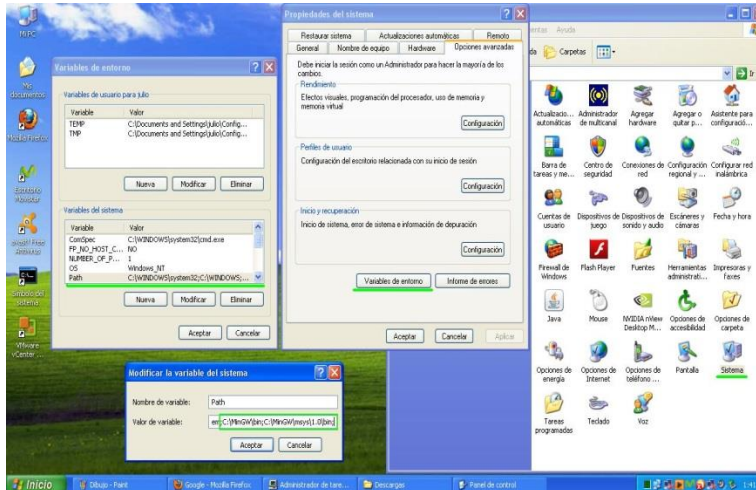
Nosotros utilizaremos MinGW ya que a mi me parece mas sencillo de instalar. Para descargarlo vamos al siguiente enlace: <http://sourceforge.net/projects/mingw/files/>



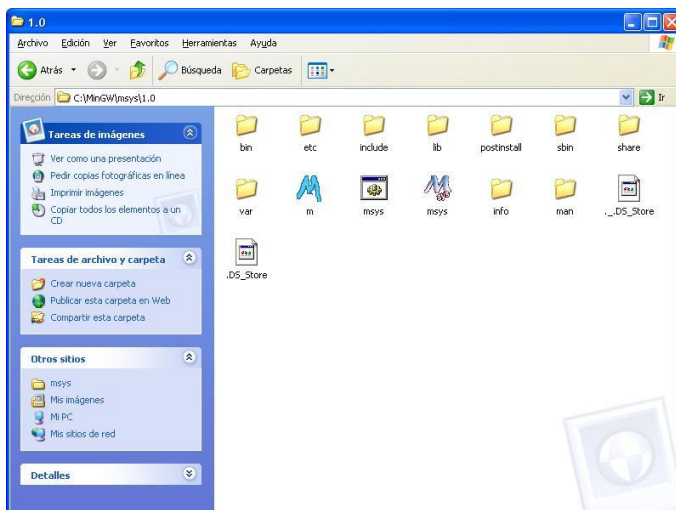
Una vez todo descargado e instalado empezaremos con lo complejo.

## Modificar la variable del equipo y acoplar el Debugger.

Necesitamos modificar la variable del sistema llamada "Path" poniendo al final de la variable lo siguiente: ;C:\MinGW\bin;C:\MinGW\msys\1.0\bin; En la siguiente foto se ve como se accede a la variable desde Windows XP, para Windows 7".



Para el debugger descomprimos el archivo "gdb-6.8-mingw-3.tar", dentro de este encontramos el siguiente archivo "gdb-6.8-mingw-3.tar" y lo descomprimos, seleccionamos todo el contenido de la carpeta anterior y combinamos los archivos con la siguiente dirección "C:\MinGW\msys\1.0" tal y como se muestra en la captura.

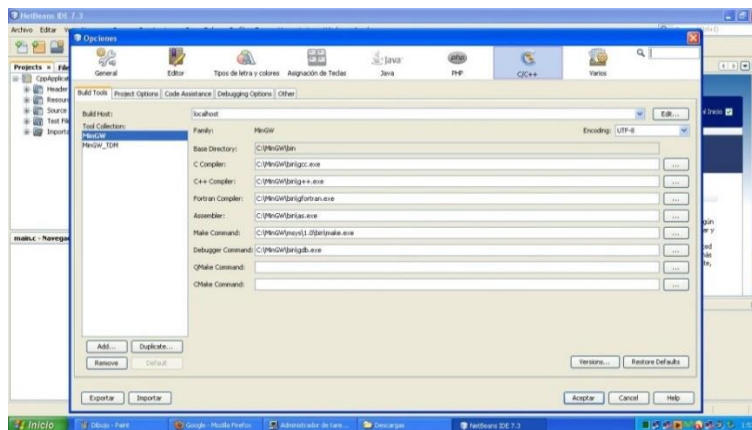


**Comprobar que todo funcione.**

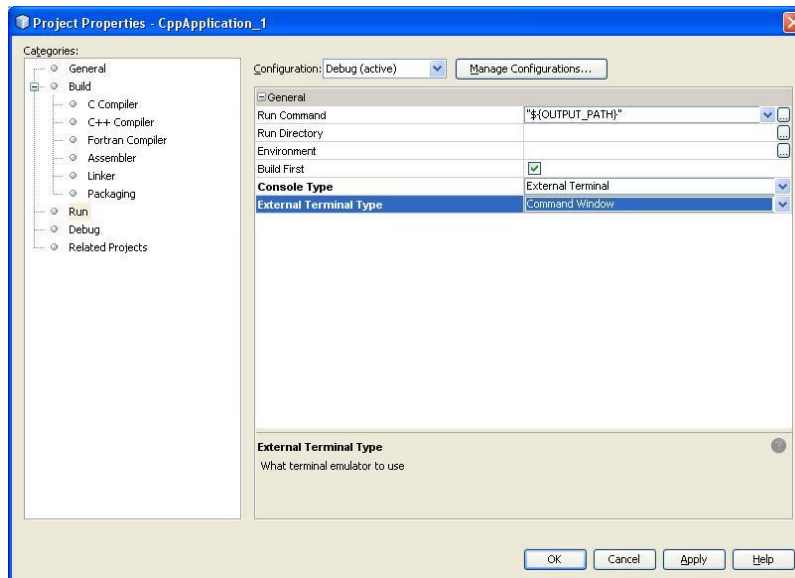
Antes de nada advierto que este paso puede variar dependiendo del equipo y de como halláis completado los pasos, yo pondré los míos. Bueno, una vez completado todos los



pasos iniciamos NetBeans, al iniciarlo nos vamos a la barra superior y en “Herramientas” hacemos clic en “Opciones”, en la nueva ventana buscamos la pestaña de C/C++ y comprobamos que este activo, si no es así activarlo desde la misma ventana sin problemas, una vez activado debe aparecer la ventana como se muestra en la foto.



Para el Debugger tendremos que meter el descargado insertando la dirección manualmente, donde pone “Debugger Command insertar la siguiente dirección:



## A.04 Manual de usuario

Ingreso al Sistema: el usuario realiza un ingreso de datos al login para ingresar al sistema.

El campo usuario es obligatorio

El campo contraseña es obligatorio

Presione para iniciar sesión

Restricción de Acceso. El usuario al momento de ingresar a la pantalla principal no podrá visualizar los módulos sin antes registrarse. Ya que el sistema le muestra una pantalla de error el mismo permitirá realizar sus funciones determinadas y con sus debidas restricciones de usuario.

Búsqueda de cliente. Clic sobre el botón

Campos obligatorios

Busqueda de cliente. El usuario al momento de ingresar podrá realizar la búsqueda de un cliente para crear la incidencia.

Creación de un cliente. Clic sobre el botón

Ingrese los campos del formulario

**Crear Cliente TradeSystem**

Codigo Cliente *	<input type="text"/>	R.Social	<input type="text"/>
Ruc	<input type="text"/>	Direccion Titular	<input type="text"/>
Telefono Titular	<input type="text"/>	Telefono2 Titular	<input type="text"/>
Grupo Clientes	<input type="text"/>	Mail	<input type="text"/>
Nombre Contacto	<input type="text"/>	Apellido Contacto	<input type="text"/>
Email Contacto	<input type="text"/>		

Seleccione Usuario





















Usuario: -- Seleccione --

**Crear** **Cancelar**

Creación de cliente. El usuario al momento de ingresar podrá realizar la creación de un nuevo cliente, los campos con \* de color rojo son obligatorios.

Motivos de incidencia. Clic sobre el botón

**Administrar Motivos de Incidencia**

Negocio	Tipificación	Motivo	Tiempo	Estado	
REPSOLGAS	Soporte Técnico	Telefonía		Activo	 
REPSOLGAS	Soporte Técnico	Equipos informáticos	48	Activo	 
REPSOLGAS	Pedidos	Equipos informáticos	48	Activo	 
REPSOLGAS	Pedidos	Materiales	48	Activo	 
REPSOLGAS	Pedidos	Partes y Piezas	48	Activo	 
REPSOLGAS	Pedidos	Componentes de Telefonía	48	Activo	 
REPSOLGAS	Mantenimiento Preventivo	Frecuencia Programada	48	Activo	 
REPSOLGAS	Mantenimiento Preventivo	No programado	48	Activo	 
REPSOLGAS	Servicios Técnicos	Gestión de Garantía	48	Activo	 
REPSOLGAS	Servicios Técnicos	Inspecciones	48	Activo	 

← 1 2 3 →

Administrar motivos de incidencia. El usuario al momento de ingresar podrá realizar modificación y creación de motivos de incidencia.

Prioridades de incidencia. Clic sobre el botón

**Administrar Prioridades de Incidencia**

Nombre	Estado	Color
<input type="text"/>	<input type="text"/>	
Alto	Activo	
Bajo	Activo	
Medio	Activo	
Urgente	Activo	

Administrar responsable de incidencia. El usuario al momento de ingresar podrá realizar modificación y crecion de responsables de incidencia.

Responsables de incidencia. Clic sobre el botón

**Administrar Responsables**

Nombre	Email	Estado	Usuario
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Andres Barreno	owk666@gmail.com	Activo	andres.barreno
Christian Villareal	owk666@gmail.com	Activo	christian.villareal
Gabriel Mejia	owk666@gmail.com	Activo	gabriel.mejia
Giovanni Granda	owk666@gmail.com	Activo	giovanni.granda
Kevin Hidalgo	owk666@gmail.com	Activo	kevin.hidalgo
Mauricio Arcentales	owk666@gmail.com	Activo	mauricio.arcentales
Roberto Vozmediano	owk666@gmail.com	Activo	roberto.vozmediano
Roy Chavez	owk666@gmail.com	Activo	roy.chavez
Santiago Benitez	owk666@gmail.com	Activo	santiago.benitez
Sonia Quipo	owk666@gmail.com	Activo	sonia.quipo

## A.05 Manual técnico

### CODIGO FUENTE REPORTE

```
public function actionIndex() {

    $model = new ReporteIncidencias;

    $arrayReportIncidencias = array();
    $arrayReportIncidenciasAbiertas = array();
    $arrayReportIncidenciasEnProceso = array();

    $arrayReportIncidenciasGestionadas = array(); // agreado por Jhonaran
    Quilca
    $arrayReportIncidenciasFacturadas = array(); // agreado por Jhonaran Quilca
    $arrayReportIncidenciasCobradas = array(); // agreado por Jhonaran Quilca

    $arrayReportIncidenciasResueltas = array();
    $arrayReportIncidenciasResueltasATiempo = array();
    $arrayReportIncidenciasResueltasEficiencia = array();
    $arrayReportIncidenciasResueltasATiempoEficiencia = array();
    $visualizarReporte = false;

    $this->performAjaxValidation($model);

    if (isset($_POST['ReporteIncidencias'])) {
        $model->attributes = $_POST['ReporteIncidencias'];
        $viaIngresoArray = explode(",", $model->via_ingreso);
        $incidencias = null;
        $incidenciasResueltas = null;
        $incidenciasAbiertas = null;
        $incidenciasEnProceso = null;

        $incidenciasGestionadas = null; //
        $incidenciasFacturadas = null; //
        $incidenciasCobradas = null; //

        $incidenciasResueltasATiempo = null;
        foreach ($viaIngresoArray as $viaIngresoId) {

            switch ($model->tipo_reporte) {
                case "TIPIFICACION":
                    $incidenciasAux = Incidencia::model()-
>getIncidenciasPorTipificacion($model->anio, $model->categoria, $viaIngresoId,
                    $model->ciudad);
```

```
$incidenciasAbiertasAux = Incidencia::model()-
>getIncidenciasPorTipificacion($model->anio, $model->categoria, $viaIngresoId,
$model->ciudad, "ABIERTO");
    $incidenciasEnProcesoAux = Incidencia::model()-
>getIncidenciasPorTipificacion($model->anio, $model->categoria, $viaIngresoId,
$model->ciudad, "EN_PROCESO");
    $incidenciasGestionadasAux = Incidencia::model()-
>getIncidenciasPorTipificacion($model->anio, $model->categoria, $viaIngresoId,
$model->ciudad, array("GESTIONADO"));
    $incidenciasFacturadasAux = Incidencia::model()-
>getIncidenciasPorTipificacion($model->anio, $model->categoria, $viaIngresoId,
$model->ciudad, array("FACTURADO"));
    $incidenciasCobradasAux = Incidencia::model()-
>getIncidenciasPorTipificacion($model->anio, $model->categoria, $viaIngresoId,
$model->ciudad, array("COBRADO"));
    //
    $incidenciasResueltasAux = Incidencia::model()-
>getIncidenciasPorTipificacion($model->anio, $model->categoria, $viaIngresoId,
$model->ciudad, array("GESTIONADO", "CERRADO"));
    $incidenciasResueltasATiempoAux = Incidencia::model()-
>getIncidenciasPorTipificacion($model->anio, $model->categoria, $viaIngresoId,
$model->ciudad, array("GESTIONADO", "CERRADO"), true);
    break;
    case "RESPONSABLE":
        $incidenciasAux = Incidencia::model()-
>getIncidenciasPorResponsable($model->anio, $model->tipificacion,
$viaIngresoId, $model->ciudad);
        $incidenciasAbiertasAux = Incidencia::model()-
>getIncidenciasPorResponsable($model->anio, $model->tipificacion,
$viaIngresoId, $model->ciudad, "ABIERTO");
        $incidenciasEnProcesoAux = Incidencia::model()-
>getIncidenciasPorResponsable($model->anio, $model->tipificacion,
$viaIngresoId, $model->ciudad, "EN_PROCESO");
        $incidenciasGestionadasAux = Incidencia::model()-
>getIncidenciasPorResponsable($model->anio, $model->categoria, $viaIngresoId,
$model->ciudad, array("GESTIONADO"));
        $incidenciasFacturadasAux = Incidencia::model()-
>getIncidenciasPorResponsable($model->anio, $model->categoria, $viaIngresoId,
$model->ciudad, array("FACTURADO"));
        $incidenciasCobradasAux = Incidencia::model()-
>getIncidenciasPorResponsable($model->anio, $model->categoria, $viaIngresoId,
$model->ciudad, array("COBRADO"));
        //
        $incidenciasResueltasAux = Incidencia::model()-
>getIncidenciasPorResponsable($model->anio, $model->tipificacion,
$viaIngresoId, $model->ciudad, array("GESTIONADO", "CERRADO"));
```

```
$incidenciasResueltasATiempoAux = Incidencia::model()-
>getIncidenciasPorResponsable($model->anio, $model->tipificacion,
$viaIngresoId, $model->ciudad, array("GESTIONADO", "CERRADO"), true);
break;
case "MOTIVO":
    $incidenciasAux = Incidencia::model()-
    >getIncidenciasPorMotivo($model->anio, $model->categoria, $model-
    >tipificacion, $model->producto, $viaIngresoId, $model->ciudad);
    $incidenciasAbiertasAux = Incidencia::model()-
    >getIncidenciasPorMotivo($model->anio, $model->categoria, $model-
    >tipificacion, $model->producto, $viaIngresoId, $model->ciudad, "ABIERTO");
    $incidenciasEnProcesoAux = Incidencia::model()-
    >getIncidenciasPorMotivo($model->anio, $model->categoria, $model-
    >tipificacion, $model->producto, $viaIngresoId, $model->ciudad,
    "EN_PROCESO");
    $incidenciasGestionadasAux = Incidencia::model()-
    >getIncidenciasPorMotivo($model->anio, $model->categoria, $model-
    >tipificacion, $model->producto, $viaIngresoId, $model->ciudad,
    array("GESTIONADO"));
    $incidenciasFacturadasAux = Incidencia::model()-
    >getIncidenciasPorMotivo($model->anio, $model->categoria, $model-
    >tipificacion, $model->producto, $viaIngresoId, $model->ciudad,
    array("FACTURADO"));
    $incidenciasCobradasAux = Incidencia::model()-
    >getIncidenciasPorMotivo($model->anio, $model->categoria, $model-
    >tipificacion, $model->producto, $viaIngresoId, $model->ciudad,
    array("COBRADO"));
    $incidenciasResueltasAux = Incidencia::model()-
    >getIncidenciasPorMotivo($model->anio, $model->categoria, $model-
    >tipificacion, $model->producto, $viaIngresoId, $model->ciudad,
    array("GESTIONADO", "CERRADO"));
    $incidenciasResueltasATiempoAux = Incidencia::model()-
    >getIncidenciasPorMotivo($model->anio, $model->categoria, $model-
    >tipificacion, $model->producto, $viaIngresoId, $model->ciudad,
    array("GESTIONADO", "CERRADO"), true);
    break;
case "SUBMOTIVO":
    $incidenciasAux = Incidencia::model()-
    >getIncidenciasPorSubmotivo($model->anio, $model->categoria, $model-
    >tipificacion, $model->motivo, $model->producto, $viaIngresoId, $model-
    >ciudad);
    $incidenciasAbiertasAux = Incidencia::model()-
    >getIncidenciasPorSubmotivo($model->anio, $model->categoria, $model-
    >tipificacion, $model->motivo, $model->producto, $viaIngresoId, $model-
    >ciudad, "ABIERTO");
    $incidenciasEnProcesoAux = Incidencia::model()-
    >getIncidenciasPorSubmotivo($model->anio, $model->categoria, $model-
```

```
>tipificacion, $model->motivo, $model->producto, $viaIngresoId, $model-  
>ciudad, "EN_PROCESO");  
    $incidenciasGestionadasAux = Incidencia::model()-  
>getIncidenciasPorSubmotivo($model->anio, $model->categoria, $model-  
>tipificacion, $model->motivo, $model->producto, $viaIngresoId, $model-  
>ciudad, array("GESTIONADO"));  
    $incidenciasFacturadasAux = Incidencia::model()-  
>getIncidenciasPorSubmotivo($model->anio, $model->categoria, $model-  
>tipificacion, $model->motivo, $model->producto, $viaIngresoId, $model-  
>ciudad, array("FACTURADO"));  
    $incidenciasCobradasAux = Incidencia::model()-  
>getIncidenciasPorSubmotivo($model->anio, $model->categoria, $model-  
>tipificacion, $model->motivo, $model->producto, $viaIngresoId, $model-  
>ciudad, array("COBRADO"));  
    //  
    $incidenciasResueltasAux = Incidencia::model()-  
>getIncidenciasPorSubmotivo($model->anio, $model->categoria, $model-  
>tipificacion, $model->motivo, $model->producto, $viaIngresoId, $model-  
>ciudad, array("GESTIONADO", "CERRADO"));  
    $incidenciasResueltasATiempoAux = Incidencia::model()-  
>getIncidenciasPorSubmotivo($model->anio, $model->categoria, $model-  
>tipificacion, $model->motivo, $model->producto, $viaIngresoId, $model-  
>ciudad, array("GESTIONADO", "CERRADO"), true);  
    break;  
    default:  
    break;  
}  
  
//Incidencias  
if (!$incidencias) {  
    $incidencias = $incidenciasAux;  
} else {  
    $this->sumaViasIngreso($incidencias, $incidenciasAux);  
}  
//Incidencias Abiertas  
if (!$incidenciasAbiertas) {  
    $incidenciasAbiertas = $incidenciasAbiertasAux;  
} else {  
    $this->sumaViasIngreso($incidenciasAbiertas,  
$incidenciasAbiertasAux);  
}  
//Incidencias En Proceso  
if (!$incidenciasEnProceso) {  
    $incidenciasEnProceso = $incidenciasEnProcesoAux;  
} else {  
    $this->sumaViasIngreso($incidenciasAbiertas,  
$incidenciasAbiertasAux);  
}
```



```
//Incidencias Gestionadas
if (!$IncidenciasGestionadas) {
    $IncidenciasGestionadas =
($IncidenciasGestionadasAux?$IncidenciasGestionadasAux:"");
} else {
    $this->sumaViasIngreso($IncidenciasAbiertas,
$IncidenciasAbiertasAux);
}
//Incidencias Facturadas
if (!$IncidenciasFacturadas) {
    $IncidenciasFacturadas =
($IncidenciasGestionadasAux?$IncidenciasGestionadasAux:"");
} else {
    $this->sumaViasIngreso($IncidenciasAbiertas,
$IncidenciasAbiertasAux);
}
//Incidencias Cobradas
if (!$IncidenciasCobradas) {
    $IncidenciasCobradas = $IncidenciasCobradasAux;
} else {
    $this->sumaViasIngreso($IncidenciasAbiertas,
$IncidenciasAbiertasAux);
}
//Incidencias Resueltas
if (!$IncidenciasResueltas) {
    $IncidenciasResueltas = $IncidenciasResueltasAux;
} else {
    $this->sumaViasIngreso($IncidenciasResueltas,
$IncidenciasResueltasAux);
}
//Incidencias Resueltas a Tiempo
if (!$IncidenciasResueltasATiempo) {
    $IncidenciasResueltasATiempo = $IncidenciasResueltasATiempoAux;
} else {
    $this->sumaViasIngreso($IncidenciasResueltasATiempo,
$IncidenciasResueltasATiempoAux);
}
}
$arrayReportIncidencias = $this->buildData($arrayReportIncidencias,
$Incidencias);
$arrayReportIncidenciasAbiertas = $this-
>buildData($arrayReportIncidenciasAbiertas, $IncidenciasAbiertas);
$arrayReportIncidenciasEnProceso = $this-
>buildData($arrayReportIncidenciasEnProceso, $IncidenciasEnProceso);
$arrayReportIncidenciasGestionadas = $this-
>buildData($arrayReportIncidenciasGestionadas, $IncidenciasGestionadas);
$arrayReportIncidenciasFacturadas = $this-
>buildData($arrayReportIncidenciasFacturadas, $IncidenciasFacturadas);
```

```
$arrayReportIncidentesCobradas = $this-
>buildData($arrayReportIncidentesCobradas, $incidentesCobradas);

$arrayReportIncidentesResueltas = $this-
>buildData($arrayReportIncidentesResueltas, $incidentesResueltas);
$arrayReportIncidentesResueltasATiempo = $this-
>buildData($arrayReportIncidentesResueltasATiempo,
$incidentesResueltasATiempo);
$arrayReportIncidentesResueltasEficiencia = $this-
>efectividad($arrayReportIncidentes, $arrayReportIncidentesResueltas);
$arrayReportIncidentesResueltasATiempoEficiencia = $this-
>efectividad($arrayReportIncidentes,
$arrayReportIncidentesResueltasATiempo);
$visualizarReporte = true;
}

$this->render("index", array(
    "arrayReportIncidentes" => $arrayReportIncidentes,
    "arrayReportIncidentesAbiertas" => $arrayReportIncidentesAbiertas,
    "arrayReportIncidentesEnProceso" => $arrayReportIncidentesEnProceso,

    "arrayReportIncidentesGestionadas" =>
$arrayReportIncidentesGestionadas,
    "arrayReportIncidentesFacturadas" => $arrayReportIncidentesFacturadas,
    "arrayReportIncidentesCobradas" => $arrayReportIncidentesCobradas,
    "arrayReportIncidentesResueltas" => $arrayReportIncidentesResueltas,
    "arrayReportIncidentesResueltasATiempo" =>
$arrayReportIncidentesResueltasATiempo,
    "arrayReportIncidentesResueltasEficiencia" =>
$arrayReportIncidentesResueltasEficiencia,
    "arrayReportIncidentesResueltasATiempoEficiencia" =>
$arrayReportIncidentesResueltasATiempoEficiencia,
    "visualizarReporte" => $visualizarReporte,
    "model" => $model
));
}
```

## **CODIGO PENDIENTES**

```
/**
 * Retorna el campo de parametrización requerido para el proceso de llamada
 * @param type $actor
 * @param type $id
 * @author Ivan Naranjo <inaranjo@tradesystem.com.ec>
 */
public function actionGetActorNumber($actor, $id) {
    echo CJSON::encode(Incidencia::getActorNumber($actor, $id));
}
```

```
/**
 * Retorna el nombre del usuario que esta gestionando pendiente
 * @param type $id
 * @author Ivan Naranjo <inaranjo@tradesystem.com.ec>
 */
public function actionUsuarioEnGestion($id = NULL) {
    echo CJSON::encode(CrugaStoredUser::model()->findByPk($id)->username);
}

public function actionCargarArchivo($id = NULL) {
    Yii::import("xupload.models.XUploadForm");
    $archivo = new XUploadForm;
    $this->renderPartial('_cargarArchivo', array(
        'archivo' => $archivo,
        'id' => $id,
    ), FALSE, TRUE);
}

public function actionListarArchivos($id = null) {

    $this->renderPartial('_archivosPorIncidencia', array(
        'id' => $id,
    ), false, true);
}

public function actionUpload($id = NULL) {
    Yii::import("xupload.models.XUploadForm");
    if (!file_exists('upload/incidencias/' . $id)) {
        mkdir('upload/incidencias/' . $id, 0777, true);
    }
    $path = realpath(Yii::app()->getBasePath() . '/../upload/incidencias/' . $id) .
"/";
    $publicPath = Yii::app()->getBaseUrl() . "/upload/incidencias/" . $id;

    header('Vary: Accept');
    if (isset($_SERVER['HTTP_ACCEPT']) &&
(strpos($_SERVER['HTTP_ACCEPT'], 'application/json') !== false)) {
        header('Content-type: application/json');
    } else {
        header('Content-type: text/plain');
    }

    if (isset($_GET['_method'])) {
        if ($_GET['_method'] == "delete") {
            if ($_GET["file"][0] !== '.') {
                $file = $path . $_GET["file"];
                if (is_file($file)) {
                    unlink($file);
                }
            }
        }
    }
}
```

```
    }
  }
  echo json_encode(true);
}
} else {
  $model = new XUploadForm;
  $model->file = CUploadedFile::getInstance($model, 'file');
  if ($model->file !== null) {
    $model->mime_type = $model->file->getType();
    $model->size = $model->file->getSize();
    $model->name = $model->file->getName();
    $filename = $model->file;
    if ($model->validate()) {
      $model->file->saveAs($path . $filename);
      chmod($path . $filename, 0777);
      if (Yii::app()->user->hasState('images')) {
        $userImages = Yii::app()->user->getState('images');
      } else {
        $userImages = array();
      }
      $userImages[] = array(
        "path" => $path . $filename,
        "thumb" => $path . $filename,
        "filename" => $filename,
        'size' => $model->size,
        'mime' => $model->mime_type,
        'name' => $model->name,
      );
      Yii::app()->user->setState('images', $userImages);
      echo json_encode(array(array(
        "name" => $model->name,
        "type" => $model->mime_type,
        "size" => $model->size,
        "url" => $publicPath . $filename,
        "delete_url" => $this->createUrl("upload", array(
          "_method" => "delete",
          "file" => $filename,
          'id' => $id
        )),
        "delete_type" => "POST"
      )));
    } else {
      echo json_encode(array(
        array("error" => $model->getErrors('file'),
      )));
      Yii::log("XUploadAction: " . CVarDumper::dumpAsString($model->getErrors()), CLogger::LEVEL_ERROR, "xupload.actions.XUploadAction");
    }
  }
}
```

```
    }
    } else {
        throw new CHttpException(500, "Could not upload file");
    }
}
}

public function actionIndex($notify = NULL) {
    $usuario = Yii::app()->user->id;
    $estado = null;
    //Incidencias
    $incidencia = new Incidencia('search');
    $incidencia->unsetAttributes(); // clear any default values
    if (isset($_GET['Incidencia'])) {
        $incidencia->attributes = $_GET['Incidencia'];
        if (isset($_GET['Incidencia']['cliente_search'])) {
            $incidencia->cliente_search = $_GET['Incidencia']['cliente_search'];
        }
        if (isset($_GET['Incidencia']['cliente_duragas_search'])) {
            $incidencia->cliente_duragas_search =
$_GET['Incidencia']['cliente_duragas_search'];
        }
    }

    /*
    * el siguiente codigo es para filtrar cuando se envíe in id de incidencia
    */
    if (isset($_GET['id'])) {
        $incidencia->id = $_GET['id'];
        $estado=$incidencia->find('id=' . $incidencia->id)->estado_gestion;
    }
    /*
    * fin
    */

    $incidenciasControlProcesosAbiertas = $incidencia->model()->count(array(
        'condition' => 'categoria = :categoria AND via_ingreso_id = :via_ingreso_id
AND estado_gestion =:estado_incidencia',
        'params' => array(':categoria' => 'REPSOLGAS', ':via_ingreso_id' =>
Constants::CONTROL_PROCESOS, ':estado_incidencia' => 'ABIERTO',)
    ));
    $incidenciasControlProcesosEnProceso = $incidencia->model()->count(array(
        'condition' => 'categoria = :categoria AND via_ingreso_id = :via_ingreso_id
AND estado_gestion =:estado_incidencia',
        'params' => array(':categoria' => 'REPSOLGAS', ':via_ingreso_id' =>
Constants::CONTROL_PROCESOS, ':estado_incidencia' => 'EN_PROCESO',)
    ));
    $incidenciasControlProcesosGestionado = $incidencia->model()-
>count(array(
```

```
'condition' => 'categoria = :categoria AND via_ingreso_id = :via_ingreso_id
AND estado_gestion =:estado_incidencia',
'params' => array(':categoria' => 'REPSOLGAS', ':via_ingreso_id' =>
Constants::CONTROL_PROCESOS, ':estado_incidencia' => 'GESTIONADO',)
));
$incidenciasControlProcesosFacturado = $incidencia->model()->count(array(
'condition' => 'categoria = :categoria AND via_ingreso_id = :via_ingreso_id
AND estado_gestion =:estado_incidencia',
'params' => array(':categoria' => 'REPSOLGAS', ':via_ingreso_id' =>
Constants::CONTROL_PROCESOS, ':estado_incidencia' => 'FACTURADO',)
));
$incidenciasControlProcesosCobrado = $incidencia->model()->count(array(
'condition' => 'categoria = :categoria AND via_ingreso_id = :via_ingreso_id
AND estado_gestion =:estado_incidencia',
'params' => array(':categoria' => 'REPSOLGAS', ':via_ingreso_id' =>
Constants::CONTROL_PROCESOS, ':estado_incidencia' => 'COBRADO',)
));
//Tiempo alerta incidencia
$alertaIncidencia = ConfiguracionSistema::model()->findByPk(1)-
>tiempo_alerta_incidencia;
$this->render('index', array(
'incidencia' => $incidencia,
'incidenciasControlProcesosAbiertas' =>
$incidenciasControlProcesosAbiertas,
'incidenciasControlProcesosEnProceso' =>
$incidenciasControlProcesosEnProceso,
'incidenciasControlProcesosGestionado' =>
$incidenciasControlProcesosGestionado,
'incidenciasControlProcesosFacturado' =>
$incidenciasControlProcesosFacturado,
'incidenciasControlProcesosCobrado' =>
$incidenciasControlProcesosCobrado,
'alertaIncidencia' => $alertaIncidencia,
'estado'=>$estado,

'notify' => $notify,
));
}
/**
 * Accion para mostrar la obsorcacion que tiene una incidencia
 */
public function actionObservacionIncidencia($id = null) {
    $model = Incidencia::model()->findByPk($id);

    $this->renderPartial('_observacionIncidencia', array(
        'model' => $model,
    ), false, true);
}
```

## CODIGO DE INCIDENCIA

```
/**
 * //obtiene el tiempo en horas de la incidencia
 * @param type $id
 * @param type $categoria
 * @author Ivan Naranjo <inaranjo@tradesystem.com.ec>
 */

public function actionTiempoIncidencia($id) {
    $resp = TiempoIncidencia::model()->findAll(array("condition" => "estado =
'ACTIVO' AND id =:id ", 'params' => array(':id' => $id)));
    header("Content-type: application/json");
    echo CJSON::encode($resp);
}

/**
 * //obtiene los responsables de la tipificación
 * @param type $id
 * @author Ivan Naranjo <inaranjo@tradesystem.com.ec>
 */

public function actionResponsable($id) {
    $resp = Responsable::model()->with('tipificacionIncidencias')-
>findAll(array("condition" => "t.estado = 'ACTIVO' AND
tipificacionIncidencias_tipificacionIncidencias.tipificacion_incidencia_id =:id ", 'params'
=> array(':id' => $id), 'order' => 't.nombre'));
    header("Content-type: application/json");
    echo CJSON::encode($resp);
}

/**
 * //obtiene los motivosIncidencia segun negocio(categoria), tipificación de
incidencia(id) y viasIngreso
```

```
* @param type $id
* @param type $categoria
* @author Ivan Naranjo <inaranjo@tradesystem.com.ec>
*/

public function actionMotivos($id, $categoria, $viaIngreso) { //agregar parametro de
viasIngreso

    $resp = MotivoIncidencia::model()->with('viaIngresos')->findAll(array("condition"
=> "t.estado = 'ACTIVO' AND categoria=:categoria AND t.motivo_incidencia_id IS
NULL AND tipificacion_incidencia_id=:id AND
viaIngresos_viaIngresos.via_ingreso_id=:via ", 'params' => array(':id' => $id, ':categoria'
=> $categoria, ':via' => $viaIngreso), 'order' => 't.nombre'));

    header("Content-type: application/json");

    echo CJSON::encode($resp);

}

/**
 * //obtiene los Submotivos segun negocio(categoria), tipificación de incidencia(id) y
viaIngreso
 * @param type $id
 * @param type $categoria
 * @author Ivan Naranjo <inaranjo@tradesystem.com.ec>
 */

public function actionSubMotivos($id, $categoria, $padre, $viaIngreso) {

    $resp = MotivoIncidencia::model()->with('viaIngresos')->findAll(array("condition"
=> "t.estado = 'ACTIVO' AND categoria=:categoria AND t.motivo_incidencia_id=:padre
AND tipificacion_incidencia_id=:id AND viaIngresos_viaIngresos.via_ingreso_id=:via",
'params' => array(':id' => $id, ':categoria' => $categoria, ':padre' => $padre, ':via' =>
$viaIngreso), 'order' => 't.nombre'));

    header("Content-type: application/json");

    //    var_dump($resp);
    //    die();

    echo CJSON::encode($resp);

}

/**
```



\* //mediante ajax verifica que el estado sea abierto, y redirecciona a pendientes creando un Notify

\* @param type \$id

\* @return type

\* @author Ivan Naranjo <inaranjo@tradesystem.com.ec>

\*/

```
public function actionTecnico($id) {
```

```
    if (Yii::app()->request->isAjaxRequest) {
```

```
        $model = $this->loadModel($id);
```

```
        echo CJSON::encode($model);
```

```
        return;
```

```
    }
```

```
    $model = $this->loadModel($id);
```

```
    $model->estado_gestion = "EN_PROCESO";
```

```
    $model->valor = (int) $model->valor;
```

```
    if ($model->validate()) {
```

```
        if ($model->save()) {
```

```
            //ENVIO DE EMAIL
```

```
            $tiempo = Util::semaforo($model->fecha_gestion ? $model->fecha_gestion :  
null);
```

```
            Mailer::enviarEmail($model->responsable->email, 'INCIDENCIA  
GESTIONADA TRADESYSTEM # ' . $model->id, '<table>
```

```
        <tr>
```

```
            <td style="padding: 10px;"></td>
```

```
        </tr>
```

```
        <tr>
```

```
            <td colspan="2"><b>Estimado ' . $model->responsable->nombre  
' ,</b></td>
```

```
        </tr>
```

```
<br/>

<tr>

    <td colspan="2"><b>El requerimiento ha sido cumplido
satisfactoriamente, a continuaci&oacute;n los detalles de la misma:</b></td>

</tr>

<br/>

<tr>

    <td colspan="2"><b>#: ' . $model->id . '</b></td>

</tr>

<tr>

    <td colspan="2">Cliente :<b>' . $model-
>clienteRepsolgasCliente->nombre_cliente . '.' . '</b></td>

</tr>

<tr>

    <td colspan="2">Tipificaci&oacute;n de la incidencia :<b>' .
$model->tipificacionIncidencia->nombre . '.' . '</b></td>

</tr>

<tr>

    <td colspan="2">Motivo de la incidencia :<b>' . $model-
>motivoIncidencia->nombre . '.' . '</b></td>

</tr>

<tr>

    <td colspan="2">Tiempo para resolver :<b>' . $tiempo . '.' .
'</b></td>

</tr>

<tr>

    <td colspan="2">Prioridad :<b>' . $model->prioridadIncidencia-
>nombre . '.' . '</b></td>

</tr>

<tr>

    <td colspan="2">Observaciones :<b>' . $model->observaciones .
'.' . '</b></td>

</tr>

<br/>
```

```

        <tr>

            <td colspan="2"><b>Para acceder a la incidencia puedes dar
click en el siguiente link :

            <br/>

            <a href="" . Yii::app()->createAbsoluteUrl("pendientes") .
'/default/index/id/' . $model->id . "'>' . Yii::app()->createAbsoluteUrl("pendientes") .
'/default/index/id/' . $model->id . '</a></b></td>

        </tr>

        <tr>

            <td colspan="2" style="padding: 10px;"></td>

        </tr>

        <tr>

            <td colspan="2"><b>Gracias por tu amable colaboraci&oacute;n.'
. '</b></td>

        </tr>

        <tr>

            <td colspan="2" style="padding: 10px;"></td>

        </tr>

    </table>'

    );

    $this->redirect(array('../pendientes/default/index', 'notify' => $id));

    }

    } else {

    }

}
    
```

**a. Manual de mantenimiento**

Estructura de la base.

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
```

```
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,  
FOREIGN_KEY_CHECKS=0;
```

```
SET @OLD_SQL_MODE=@@SQL_MODE,  
SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';
```

```
CREATE SCHEMA IF NOT EXISTS `tradesystem` DEFAULT CHARACTER SET latin1 ;  
USE `tradesystem` ;
```

```
-- Table `tradesystem`.`cruge_authitem`  
-----
```

```
CREATE TABLE IF NOT EXISTS `tradesystem`.`cruge_authitem` (  
  `name` VARCHAR(64) NOT NULL,  
  `type` INT(11) NOT NULL,  
  `description` TEXT NULL DEFAULT NULL,  
  `bizrule` TEXT NULL DEFAULT NULL,  
  `data` TEXT NULL DEFAULT NULL,  
  PRIMARY KEY (`name`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = latin1;
```

```
-- Table `tradesystem`.`cruge_user`  
-----
```

```
CREATE TABLE IF NOT EXISTS `tradesystem`.`cruge_user` (  
  `iduser` INT(11) NOT NULL AUTO_INCREMENT,  
  `regdate` BIGINT(30) NULL DEFAULT NULL,  
  `actdate` BIGINT(30) NULL DEFAULT NULL,  
  `logondate` BIGINT(30) NULL DEFAULT NULL,
```

```
`username` VARCHAR(64) NULL DEFAULT NULL,  
`email` VARCHAR(45) NULL DEFAULT NULL,  
`password` VARCHAR(64) NULL DEFAULT NULL COMMENT 'Hashed password',  
`authkey` VARCHAR(100) NULL DEFAULT NULL COMMENT 'llave de autentificacion',  
`state` INT(11) NULL DEFAULT '0',  
`totalsessioncounter` INT(11) NULL DEFAULT '0',  
`currentsessioncounter` INT(11) NULL DEFAULT '0',  
PRIMARY KEY (`iduser`))  
ENGINE = InnoDB  
AUTO_INCREMENT = 36  
DEFAULT CHARACTER SET = latin1;
```

```
-- Table `tradesystem`.`cruge_authassignment`  
-----
```

```
CREATE TABLE IF NOT EXISTS `tradesystem`.`cruge_authassignment` (  
  `userid` INT(11) NOT NULL,  
  `bizrule` TEXT NULL DEFAULT NULL,  
  `data` TEXT NULL DEFAULT NULL,  
  `itemname` VARCHAR(64) NOT NULL,  
  PRIMARY KEY (`userid`, `itemname`),  
  CONSTRAINT `fk_cruge_authassignment_cruge_authitem1`  
    FOREIGN KEY (`itemname`)  
      REFERENCES `tradesystem`.`cruge_authitem` (`name`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_cruge_authassignment_user`  
    FOREIGN KEY (`userid`)  
      REFERENCES `tradesystem`.`cruge_user` (`iduser`)  
    ON DELETE CASCADE  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB
```

DEFAULT CHARACTER SET = latin1;

```
CREATE INDEX `fk_crug_assignment_crug_authitem1` ON
`tradesystem`.`crug_assignment` (`itemname` ASC);
```

```
CREATE INDEX `fk_crug_assignment_user` ON `tradesystem`.`crug_assignment`
(`userid` ASC);
```

-----

-- Table `tradesystem`.`crug\_authitemchild`

-----

```
CREATE TABLE IF NOT EXISTS `tradesystem`.`crug_authitemchild` (
```

```
  `parent` VARCHAR(64) NOT NULL,
```

```
  `child` VARCHAR(64) NOT NULL,
```

```
  PRIMARY KEY (`parent`, `child`),
```

```
  CONSTRAINT `crug_authitemchild_ibfk_1`
```

```
    FOREIGN KEY (`parent`)
```

```
      REFERENCES `tradesystem`.`crug_authitem` (`name`)
```

```
    ON DELETE CASCADE
```

```
    ON UPDATE CASCADE,
```

```
  CONSTRAINT `crug_authitemchild_ibfk_2`
```

```
    FOREIGN KEY (`child`)
```

```
      REFERENCES `tradesystem`.`crug_authitem` (`name`)
```

```
    ON DELETE CASCADE
```

```
    ON UPDATE CASCADE)
```

```
ENGINE = InnoDB
```

```
DEFAULT CHARACTER SET = latin1;
```

```
CREATE INDEX `child` ON `tradesystem`.`crug_authitemchild` (`child` ASC);
```

-----

-- Table `tradesystem`.`crug\_field`

```
-----  
CREATE TABLE IF NOT EXISTS `tradesystem`.`cruge_field` (  
  `idfield` INT(11) NOT NULL AUTO_INCREMENT,  
  `fieldname` VARCHAR(20) NOT NULL,  
  `longname` VARCHAR(50) NULL DEFAULT NULL,  
  `position` INT(11) NULL DEFAULT '0',  
  `required` INT(11) NULL DEFAULT '0',  
  `fieldtype` INT(11) NULL DEFAULT '0',  
  `fieldsize` INT(11) NULL DEFAULT '20',  
  `maxlength` INT(11) NULL DEFAULT '45',  
  `showinreports` INT(11) NULL DEFAULT '0',  
  `useregexp` VARCHAR(512) NULL DEFAULT NULL,  
  `useregexpmsg` VARCHAR(512) NULL DEFAULT NULL,  
  `predetvalue` MEDIUMBLOB NULL DEFAULT NULL,  
  PRIMARY KEY (`idfield`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = latin1;
```

```
-----  
-- Table `tradesystem`.`cruge_fieldvalue`  
-----
```

```
CREATE TABLE IF NOT EXISTS `tradesystem`.`cruge_fieldvalue` (  
  `idfieldvalue` INT(11) NOT NULL AUTO_INCREMENT,  
  `iduser` INT(11) NOT NULL,  
  `idfield` INT(11) NOT NULL,  
  `value` BLOB NULL DEFAULT NULL,  
  PRIMARY KEY (`idfieldvalue`),  
  CONSTRAINT `fk_crug_fieldvalue_crug_field1`  
    FOREIGN KEY (`idfield`)  
    REFERENCES `tradesystem`.`cruge_field` (`idfield`)  
    ON DELETE CASCADE  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_crug_fieldvalue_crug_user1`
```



```
FOREIGN KEY (`iduser`)
REFERENCES `tradesystem`.`cruge_user` (`iduser`)
ON DELETE CASCADE
ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;

CREATE INDEX `fk_cruge_fieldvalue_cruge_user1` ON `tradesystem`.`cruge_fieldvalue`
(`iduser` ASC);

CREATE INDEX `fk_cruge_fieldvalue_cruge_field1` ON `tradesystem`.`cruge_fieldvalue`
(`idfield` ASC);

-----
-- Table `tradesystem`.`cruge_session`
-----

CREATE TABLE IF NOT EXISTS `tradesystem`.`cruge_session` (
  `idsession` INT(11) NOT NULL AUTO_INCREMENT,
  `iduser` INT(11) NOT NULL,
  `created` BIGINT(30) NULL DEFAULT NULL,
  `expire` BIGINT(30) NULL DEFAULT NULL,
  `status` INT(11) NULL DEFAULT '0',
  `ipaddress` VARCHAR(45) NULL DEFAULT NULL,
  `usagecount` INT(11) NULL DEFAULT '0',
  `lastusage` BIGINT(30) NULL DEFAULT NULL,
  `logoutdate` BIGINT(30) NULL DEFAULT NULL,
  `ipaddressout` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`idsession`))
ENGINE = InnoDB
AUTO_INCREMENT = 11
DEFAULT CHARACTER SET = latin1;

CREATE INDEX `crugesession_iduser` ON `tradesystem`.`cruge_session` (`iduser` ASC);
```



-----  
-- Table `tradesystem`.`cruge\_system`  
-----

```
CREATE TABLE IF NOT EXISTS `tradesystem`.`cruge_system` (  
  `idsystem` INT(11) NOT NULL AUTO_INCREMENT,  
  `name` VARCHAR(45) NULL DEFAULT NULL,  
  `largename` VARCHAR(45) NULL DEFAULT NULL,  
  `sessionmaxdurationmins` INT(11) NULL DEFAULT '30',  
  `sessionmaxsameipconnections` INT(11) NULL DEFAULT '10',  
  `sessionreusesessions` INT(11) NULL DEFAULT '1' COMMENT '1yes 0no',  
  `sessionmaxsessionsperday` INT(11) NULL DEFAULT '-1',  
  `sessionmaxsessionsperuser` INT(11) NULL DEFAULT '-1',  
  `systemnonewsessions` INT(11) NULL DEFAULT '0' COMMENT '1yes 0no',  
  `systemdown` INT(11) NULL DEFAULT '0',  
  `registerusingcaptcha` INT(11) NULL DEFAULT '0',  
  `registerusingterms` INT(11) NULL DEFAULT '0',  
  `terms` BLOB NULL DEFAULT NULL,  
  `registerusingactivation` INT(11) NULL DEFAULT '1',  
  `defaultroleforregistration` VARCHAR(64) NULL DEFAULT NULL,  
  `registerusingtermslabel` VARCHAR(100) NULL DEFAULT NULL,  
  `registrationonlogin` INT(11) NULL DEFAULT '1',  
  PRIMARY KEY (`idsystem`))  
ENGINE = InnoDB  
AUTO_INCREMENT = 2  
DEFAULT CHARACTER SET = latin1;
```

-----  
-- Table `tradesystem`.`incidencia\_estado`  
-----

```
CREATE TABLE IF NOT EXISTS `tradesystem`.`incidencia_estado` (  
  `id` INT(11) NOT NULL AUTO_INCREMENT,  
  `nombre` VARCHAR(64) NOT NULL,
```

```
`porcentaje` INT(11) NOT NULL,  
`peso` INT(11) NULL DEFAULT NULL,  
`estado` ENUM('ACTIVO','INACTIVO') NOT NULL,  
PRIMARY KEY (`id`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = latin1;
```

```
-----  
-- Table `tradesystem`.`incidencia_prioridad`  
-----
```

```
CREATE TABLE IF NOT EXISTS `tradesystem`.`incidencia_prioridad` (  
  `id` INT(11) NOT NULL AUTO_INCREMENT,  
  `nombre` VARCHAR(64) NOT NULL,  
  `color` VARCHAR(10) NOT NULL,  
  `peso` INT(11) NULL DEFAULT NULL,  
  `estado` ENUM('ACTIVO','INACTIVO') NOT NULL,  
  PRIMARY KEY (`id`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = latin1;
```

```
-----  
-- Table `tradesystem`.`incidencia_categoria`  
-----
```

```
CREATE TABLE IF NOT EXISTS `tradesystem`.`incidencia_categoria` (  
  `id` INT(11) NOT NULL AUTO_INCREMENT,  
  `nombre` VARCHAR(64) NOT NULL,  
  `estado` ENUM('ACTIVO','INACTIVO') NOT NULL,  
  PRIMARY KEY (`id`))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = latin1;
```

```
-----  
-- Table `tradesystem`.`incidencia_motivo`  
-----
```

```
CREATE TABLE IF NOT EXISTS `tradesystem`.`incidencia_motivo` (  
  `id` INT(11) NOT NULL AUTO_INCREMENT,  
  `nombre` VARCHAR(64) NOT NULL,  
  `estado` ENUM('ACTIVO','INACTIVO') NOT NULL,  
  `incidencia_categoria_id` INT(11) NOT NULL,  
  PRIMARY KEY (`id`),  
  CONSTRAINT `fk_incidencia_motivo_incidencia_categoria1`  
    FOREIGN KEY (`incidencia_categoria_id`)  
      REFERENCES `tradesystem`.`incidencia_categoria` (`id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = latin1;
```

```
CREATE INDEX `fk_incidencia_motivo_incidencia_categoria1_idx` ON  
`tradesystem`.`incidencia_motivo` (`incidencia_categoria_id` ASC);
```

```
-----  
-- Table `tradesystem`.`incidencia_submotivo`  
-----
```

```
CREATE TABLE IF NOT EXISTS `tradesystem`.`incidencia_submotivo` (  
  `id` INT(11) NOT NULL AUTO_INCREMENT,  
  `nombre` VARCHAR(64) NOT NULL,  
  `estado` ENUM('ACTIVO','INACTIVO') NOT NULL,  
  `incidencia_categoria_id` INT(11) NOT NULL,  
  `incidencia_motivo_id` INT(11) NOT NULL,  
  PRIMARY KEY (`id`),  
  CONSTRAINT `fk_incidencia_submotivo_incidencia_motivo1`
```

```
FOREIGN KEY (`incidencia_motivo_id`)
REFERENCES `tradesystem`.`incidencia_motivo` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;

CREATE INDEX `fk_incidencia_submotivo_incidencia_motivo1_idx` ON
`tradesystem`.`incidencia_submotivo` (`incidencia_motivo_id` ASC);

-----
-- Table `tradesystem`.`incidencia_via_ingreso`
-----

CREATE TABLE IF NOT EXISTS `tradesystem`.`incidencia_via_ingreso` (
  `id` INT(11) NOT NULL AUTO_INCREMENT,
  `nombre` VARCHAR(64) NOT NULL,
  `estado` ENUM('ACTIVO','INACTIVO') NOT NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;

-----
-- Table `tradesystem`.`incidencia`
-----

CREATE TABLE IF NOT EXISTS `tradesystem`.`incidencia` (
  `id` INT(11) NOT NULL AUTO_INCREMENT,
  `nombre` VARCHAR(64) NOT NULL,
  `descripcion` TEXT NULL DEFAULT NULL,
  `estado` ENUM('ACTIVO','INACTIVO') NOT NULL,
  `usuario_creacion_id` INT(11) NOT NULL,
  `usuario_actualizacion_id` INT(11) NULL DEFAULT NULL,
```

```
`fecha_est_resolucion` DATE NULL DEFAULT NULL,  
`hora_est_resolucion` TIME NULL DEFAULT NULL,  
`fecha_creacion` DATETIME NOT NULL,  
`fecha_actualizacion` DATETIME NULL DEFAULT NULL,  
`permisos` ENUM('OWNER','ALL') NOT NULL,  
`owner_id` INT(11) NOT NULL,  
`incidencia_estado_id` INT(11) NOT NULL,  
`incidencia_submotivo_id` INT(11) NOT NULL,  
`incidencia_via_ingreso_id` INT(11) NULL DEFAULT NULL,  
`incidencia_prioridad_id` INT(11) NULL DEFAULT NULL,  
`cuenta_id` INT(11) NULL DEFAULT NULL,  
`contacto_id` INT(11) NULL DEFAULT NULL,  
PRIMARY KEY (`id`),  
CONSTRAINT `fk_incidencia_incidencia_estado1`  
  FOREIGN KEY (`incidencia_estado_id`)  
  REFERENCES `tradesystem`.`incidencia_estado` (`id`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION,  
CONSTRAINT `fk_incidencia_incidencia_prioridad1`  
  FOREIGN KEY (`incidencia_prioridad_id`)  
  REFERENCES `tradesystem`.`incidencia_prioridad` (`id`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION,  
CONSTRAINT `fk_incidencia_incidencia_submotivo1`  
  FOREIGN KEY (`incidencia_submotivo_id`)  
  REFERENCES `tradesystem`.`incidencia_submotivo` (`id`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION,  
CONSTRAINT `fk_incidencia_incidencia_via_ingreso1`  
  FOREIGN KEY (`incidencia_via_ingreso_id`)  
  REFERENCES `tradesystem`.`incidencia_via_ingreso` (`id`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)
```



ENGINE = InnoDB

DEFAULT CHARACTER SET = latin1;

```
CREATE INDEX `fk_incidencia_incidencia_estado1_idx` ON `tradesystem`.`incidencia`  
(`incidencia_estado_id` ASC);
```

```
CREATE INDEX `fk_incidencia_incidencia_via_ingreso1_idx` ON `tradesystem`.`incidencia`  
(`incidencia_via_ingreso_id` ASC);
```

```
CREATE INDEX `fk_incidencia_incidencia_prioridad1_idx` ON `tradesystem`.`incidencia`  
(`incidencia_prioridad_id` ASC);
```

```
CREATE INDEX `fk_incidencia_incidencia_submotivo1_idx` ON `tradesystem`.`incidencia`  
(`incidencia_submotivo_id` ASC);
```

```
SET SQL_MODE=@OLD_SQL_MODE;
```

```
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
```

```
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

## A.06 Diccionario de datos:

cruge_authassignment										
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
userid	INT(11)	✓	✓							
bizrule	TEXT								NULL	
data	TEXT								NULL	
itemname	VARCHAR(64)	✓	✓							

cruge_authitem										
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
name	VARCHAR(64)	✓	✓							
type	INT(11)		✓							
description	TEXT								NULL	
bizrule	TEXT								NULL	
data	TEXT								NULL	

cruge_authitemchild										
---------------------	--	--	--	--	--	--	--	--	--	--

Column name	Data Type	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
parent	VARCHAR(64)	✓	✓							
child	VARCHAR(64)	✓	✓							
cruge_field										
Column name	Data Type	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
idfield	INT(11)	✓	✓					✓		
fieldname	VARCHAR(20)		✓							
longname	VARCHAR(50)								NULL	
position	INT(11)								'0'	
required	INT(11)								'0'	
fieldtype	INT(11)								'0'	
fieldsize	INT(11)								'20'	
maxlength	INT(11)								'45'	
showinreports	INT(11)								'0'	
useregexp	VARCHAR(512)								NULL	
useregexpmsg	VARCHAR(512)								NULL	



predetvalue	MEDIUMBLOB								NULL	
<b>cruge_fieldvalue</b>										
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
idfieldvalue	INT(11)	✓	✓					✓		
iduser	INT(11)		✓							
idfield	INT(11)		✓							
value	BLOB								NULL	
<b>cruge_session</b>										
Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
idsession	INT(11)	✓	✓					✓		
iduser	INT(11)		✓							
created	BIGINT(30)								NULL	
expire	BIGINT(30)								NULL	
status	INT(11)								'0'	
ipaddress	VARCHAR(45)								NULL	
usagecount	INT(11)								'0'	

lastusage	BIGINT(30)								NULL	
logoutdate	BIGINT(30)								NULL	
ipaddressout	VARCHAR(45)								NULL	

cruge_system										
Column name	Data Type	P K	N N	U Q	BI N	U N	Z F	A I	Defau lt	Comme nt
idsystem	INT(11)	✓	✓					✓		
name	VARCHAR(45)								NULL	
largename	VARCHAR(45)								NULL	
sessionmaxdurationmins	INT(11)								'30'	
sessionmaxsameipconnections	INT(11)								'10'	
sessionreusesessions	INT(11)								'1'	1yes 0no
sessionmaxsessionsperday	INT(11)								'-1'	
sessionmaxsessionsperuser	INT(11)								'-1'	
systemnonewsessions	INT(11)								'0'	1yes 0no
systemdown	INT(11)								'0'	

registerusingcaptcha	INT(11)									'0'	
registerusingterms	INT(11)									'0'	
terms	BLOB									NULL	
registerusingactivation	INT(11)									'1'	
defaultroleforregistration	VARCHAR(64)									NULL	
registerusingtermslabel	VARCHAR(100)									NULL	
registrationonlogin	INT(11)									'1'	

**cruge\_user**

Column name	DataType	P K	N N	U Q	BI N	U N	Z F	A I	Default	Comment
iduser	INT(11)	✓	✓					✓		
regdate	BIGINT(30)								NULL	
actdate	BIGINT(30)								NULL	
logondate	BIGINT(30)								NULL	
username	VARCHAR(64)								NULL	
email	VARCHAR(45)								NULL	

password	VARCHAR(64)								NULL	Hashed password
authkey	VARCHAR(100)								NULL	llave de autenticación
state	INT(11)								'0'	
totalsessioncounter	INT(11)								'0'	
currentsessioncounter	INT(11)								'0'	

### incidencia

Column name	DataType	P K	N N	U Q	B I N	U N	Z F	A I	Defa ult	Comme nt
id	INT(11)	✓	✓					✓		
nombre	VARCHAR(64)		✓							
descripcion	TEXT								NUL L	
estado	ENUM('ACTIVO','INACTIVO')		✓							
usuario_creacion_id	INT(11)		✓							
usuario_actualizacion_id	INT(11)								NUL L	
fecha_est_resolucion	DATE								NUL	

n										L	
hora_est_resolucion	TIME									NUL L	
fecha_creacion	DATETIME		✓								
fecha_actualizacion	DATETIME									NUL L	
permisos	ENUM('OWNER','ALL')		✓								
owner_id	INT(11)		✓								
incidencia_estado_id	INT(11)		✓								
incidencia_submotivo_id	INT(11)		✓								
incidencia_via_ingreso_id	INT(11)									NUL L	
incidencia_prioridad_id	INT(11)									NUL L	
cuenta_id	INT(11)									NUL L	
contacto_id	INT(11)									NUL L	
<b>incidencia_categoria</b>											

Column name	Data Type	P K	N N	U Q	BI N	U N	Z F	A I	Default	Comment
id	INT(11)	✓	✓					✓		
nombre	VARCHAR(64)		✓							
estado	ENUM('ACTIVO','INACTIVO')		✓							

**incidencia\_estado**

Column name	Data Type	P K	N N	U Q	BI N	U N	Z F	A I	Default	Comment
id	INT(11)	✓	✓					✓		
nombre	VARCHAR(64)		✓							
porcentaje	INT(11)		✓							
peso	INT(11)								NULL	
estado	ENUM('ACTIVO','INACTIVO')		✓							

**incidencia\_motivo**

Column name	Data Type	P K	N N	U Q	BI N	U N	Z F	A I	Default	Comment
id	INT(11)	✓	✓					✓		

nombre	VARCHAR(64)		✓								
estado	ENUM('ACTIVO','INACTIVO')		✓								
incidencia_categoria_id	INT(11)		✓								

**incidencia\_prioridad**

Column name	DataType	P K	N N	U Q	BI N	U N	Z F	A I	Default	Comment
id	INT(11)	✓	✓					✓		
nombre	VARCHAR(64)		✓							
color	VARCHAR(10)		✓							
peso	INT(11)								NULL	
estado	ENUM('ACTIVO','INACTIVO')		✓							

**incidencia\_submotivo**

Column name	DataType	P K	N N	U Q	BI N	U N	Z F	A I	Default	Comment
id	INT(11)	✓	✓					✓		
nombre	VARCHAR(64)		✓							

estado	ENUM('ACTIVO','INACTIVO')	✓									
incidencia_categoria_id	INT(11)	✓									
incidencia_motivo_id	INT(11)	✓									
<b>incidencia_via_ingreso</b>											
Column name	DataType	P K	N N	U Q	BI N	U N	Z F	A I	Default	Comment	
id	INT(11)	✓	✓					✓			
nombre	VARCHAR(64)		✓								
estado	ENUM('ACTIVO','INACTIVO')		✓								





## BIBLIOGRAFIA

SLIDER (2014). Información acerca de Arquitectura 3 capas. [En Línea] [Consultado el 01 de unio del 2014]: Disponible en: <http://www.slideshare.net/Decimo/arquitectura-3-capas> (capa de negocio).

KMKey (2011) Help Desk [En Línea] [Consultado el 01 de unio del 2014]: Disponible en: [http://www.kmkey.com/productos/software\\_help\\_desk](http://www.kmkey.com/productos/software_help_desk)