



INSTITUTO TECNOLÓGICO
"CORDILLERA"

CARRERA DE ANALISIS DE SISTEMAS

SISTEMATIZACIÓN DE LOS PROCESOS DE MESA DE AYUDA MEDIANTE
UN SISTEMA INFORMÁTICO PARA LA EMPRESA PARTNERS GROUP DE
LA CIUDAD DE QUITO.

Proyecto de investigación previo a la obtención del título de
Tecnólogo en Análisis de Sistemas

Autor: Vera Segarra Cristhian Oswaldo

Tutor: Lcda. Patricia Garzón.

Quito, Abril 2015



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.

Cristhian Vera Segarra

CC 1720252574

CONTRATO DE CESIÓN SOBRE DERECHOS PROPIEDAD INTELLECTUAL

Comparecen a la celebración del presente contrato de cesión y transferencia de derechos de propiedad intelectual, por una parte, el estudiante **CRISTHIAN OSWALDO VERA SEGARRA**, 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 Y SEGUIMIENTO DE ACTIVIDADES MEDIANTE UNA MESA DE SERVICIOS, APLICANDO METODOLOGÍA ITIL PARA LA EMPRESA PARTNERS GROUP**”, 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 reconvención, 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 06 días del mes de Abril del dos mil quince.

f) _____
C.C. N° 1720252574
CEDENTE

f) _____
Instituto Superior Tecnológico Cordillera
CESIONARIO



AGRADECIMIENTO

Gracias a la Lcda. Patricia Garzón, por el valioso aporte profesional, técnico y humano brindado en el transcurso de este semestre en favor de la culminación de esta retadora carrera que después de mucho empeño y sacrificio voy a culminar. De igual forma agradecer a todo el personal docente, mis queridos maestros que al fin verán los frutos de sus enseñanzas en mí.



DEDICATORIA

A Dios por permitirme alcanzar mis sueños

A mis padres por ser mi eterna inspiración

A mis hermanos por su infinito amor

A mis compañeros por su gran aprecio



Índice General

➤ Portada	
➤ Caratula	
➤ Declaración de Aprobación Tutor y Lector	ii
➤ Declaración de autoría del estudiante	iii
➤ Declaración de cesión de derechos a la institución	iv
➤ Agradecimiento	vi
➤ Dedicatoria	vii
➤ Índice general	viii
➤ Índice de Tablas	xii
➤ Índice de Figuras	xiv
➤ Resumen Ejecutivo	xvi
➤ Abstract	xvii

Contenido

Capítulo I: Antecedentes	1
1.01. Contexto	1
1.02. Justificación.....	2
1.03. Matriz T de análisis de fuerzas.....	12
1.03.01. Análisis de la matriz T	13
Capítulo II: Involucrados	15
2.01. Requerimientos	15



2.01.01. Descripción del sistema actual	15
2.01.02. Visión y alcance	16
2.01.03. Entrevistas	17
2.01.05. Descripción Detallada	20
2.02. Mapeos de Involucrados	23
2.02. Mapeos de Involucrados	25
2.03. Matriz de Análisis de Involucrados	25
2.04. Análisis de la matriz de involucrados	27
Capítulo III: Árbol de Problemas - Objetivos	28
3.01. Árbol de Problemas.....	28
Análisis del árbol de problemas.	29
3.02. Árbol de Objetivos	30
3.03. Diagrama de casos de uso.....	32
3.04. Casos de uso de realización.....	33
3.04.01. Diagrama de casos de uso de realización y especificaciones.....	34
3.05. Diagrama de secuencias del sistema.....	36
3.06. Especificación de los casos de uso.....	38
Capítulo IV: Análisis De Alternativas.....	40
4.01. Matriz de Análisis de Alternativas.....	40
4.02. Matriz de Análisis de Impactos de los Objetivos.....	41
4.03. Estándares para el Diseño de Clases.....	42



4.04. Diagrama de clases.....	44
4.05. Modelo Físico.....	45
4.06. Diagrama de Componentes.....	47
4.7. Diagrama de estrategias.....	48
4.08. Matriz del Marco Lógico.....	50
4.09. Vistas Arquitectónicas.....	50
4.09.01. Vista lógica.....	50
4.09.02. Vista física.....	50
4.09.03. Vista de desarrollo.....	51
4.09.04. Vista de procesos.....	51
Capítulo V: Propuesta	54
5.01. Especificación de estándares de programación.....	54
Programación Orientada a Objetos	54
5.02. Diseño de Interfaces de Usuario.....	56
5.03. Especificación de pruebas de unidad.....	59
5.04. Especificación de pruebas de aceptación.....	60
5.05. Especificación de pruebas de carga.....	60
5.06. Configuración del Ambiente mínima/ideal.....	61
5.06.01. Rendimiento	61
5.06.02. Características Técnicas	61
5.06.04. Requerimientos de Hardware y Software	61



CAPITULO VI.....	62
6.01. Recursos	62
6.02. Presupuesto	63
Presupuesto	63
6.03. Cronograma de Actividades.....	63
Capítulo VII: Conclusiones y Recomendaciones.....	64
7.01. Conclusiones	64
7.02. Recomendaciones.....	64
Anexos (Apéndice).....	65
Bibliografía.....	141

Índice de tablas

Tabla 1 Matriz T del Problema Central.....	12
Tabla 2 Entrevista a Gerente Operativo.	17
Tabla 3 Matriz de requerimientos funcionales.....	18
Tabla 4 Matriz de requerimientos no funcionales.....	18
Tabla 5 Descripción detallada Requerimiento Funcional 001	19
Tabla 6 Descripción detallada Requerimiento Funcional 002	20
Tabla 7 Descripción detallada Requerimiento Funcional 003	21
Tabla 8 Matriz De Requerimientos No Funcionales.....	21
Tabla 9 Matriz de involucrados.....	24
Tabla 10 Especificación del caso de uso de realización. Recepción de incidencia ...	30
Tabla 11 Especificación del caso de uso de realización. Creación de incidencia.....	31
Tabla 12 Especificación del caso de uso de realización. Notificación de incidencia	32
Tabla 13 Especificación del caso de uso de realización. Gestión de incidencia.....	33
Tabla 14 Especificación de caso de uso. Receptar de incidencia	36
Tabla 15 Especificación de caso de uso. Crear incidencia.....	36
Tabla 16 Especificación de caso de uso. Notificación de Incidencia	37
Tabla 17 Especificación de caso de uso. Gestionar incidencia.....	37
Tabla 18 Matriz de análisis de alternativas.	38
Tabla 19 Matriz del Marco Lógico.	46
Tabla 20 Vista de desarrollo	48
Tabla 21 Identificador de la prueba. Inicio de sesión.	54
Tabla 22 Identificador de la prueba. Crear cliente.	55



Tabla 23 Identificador de la prueba. Crear incidencia.	55
Tabla 24 Identificador de la prueba. Modificar.	56
Tabla 25 Identificador de la prueba. Perfiles de usuarios (Administrador).	56
Tabla 26 Identificador de la prueba. Creación de incidencia (Administrador).	57
Tabla 27 Identificador de la prueba. Perfiles de usuario (Técnico)	57
Tabla 28 Identificador de la prueba. Carga de usuarios.	57
Tabla 29 Recursos.	61
Tabla 30 Presupuesto	61

Índice de Figuras

Figura 1: Estándar ITIL.....	6
Figura 2: Mapeo de Involucrados.. ..	23
Figura 3: Árbol de problemas.. ..	26
Figura 4: Árbol de Objetivos.	28
Figura 5: Diagrama de Casos de uso.....	29
Figura 6: Diagrama de Casos de uso y especificación.	30
Figura 7: Casos de uso creación de incidencia.....	31
Figura 8: Casos de uso notificación de incidencias.	32
Figura 9: Casos de uso gestión de incidencia.....	33
Figura 10: Ingreso de Notificaciones.. ..	34
Figura 11: Pantalla de Incidencia.....	34
Figura 12: Aviso de Notificaciones.....	35
Figura 13: Verificación de datos.....	35
Figura 14: Clases UML.....	40
Figura 15: Diagrama de clases.....	42
Figura 16: Modelo Físico.....	43
Figura 17: Capa de acceso de datos.. ..	44
Figura 18: Capa lógica de negocios.	44
Figura 19: Diagrama de estrategias.....	45
Figura 20: Vista lógica.....	47
Figura 21: Vista Física.	47
Figura 22: Diagrama de actividades del proceso de recepción de incidencia.....	48
Figura 23: Diagrama de actividades de notificación de incidencia.....	49
Figura 24: Diagrama de actividades del proceso de gestión de incidencia.....	49



Figura 25: Restricción de Acceso..	52
Figura 26: Búsqueda de cliente..	52
Figura 27: Creación de cliente..	53
Figura 28: Administrar motivos de incidencia.....	53
Figura 29: Registro de llamadas.....	54
Figura 30: Administrar responsable de incidencia.....	54
Figura 31: Pruebas de carga.	58
Figura 32: Pruebas de carga..	58
Figura 33: Cronograma de actividades.	62
Figura 34: Cronograma de actividades.	62



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 labour.

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

En la actualidad se 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 espacio de innovación en ámbitos como la industria, servicios, salud, administración, comercio, educación, entre otros.

El crecimiento constante de las sociedades y sus poblaciones ha ido de la mano con el uso y el mejoramiento constante de las TIC, que influyen directamente en el desarrollo diario de sus actividades y por ende incluye en el desarrollo de nuevas aplicaciones tecnológicas.

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

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

Como muestra de la incorporación de las TI, tenemos la constante digitalización de información y la velocidad con la que está ocurriendo, impactando la forma en la que se materializan las estrategias del negocio.

El ámbito económico de los servicios está 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.

Mijaíl Bakunin activista y filósofo soviético, sostiene, que la fuerza destructora de lo viejo es la fuerza creadora de lo nuevo: "la pasión por la destrucción es una pasión creadora" lo cual resulta muy cierto a la hora de hablar sobre la tecnología ya que cada vez es mayor la dimensión de las actividades que las personas deben realizar 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, lo cual se traduce en menos oportunidades de trabajo para las personas, quienes serán suplidas por sistemas informáticos que automaticen en procesos determinados las actividades y tareas de cada individuo, convirtiendo a la tecnología en destructiva en el sentido de disminuir plazas de trabajo.

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 un área de soporte al negocio, descuidando incluso muchas veces el uso de criterios racionales que permiten medir su rentabilidad, eficacia y la calidad del servicio ofrecidos a toda la organización, por lo cual es recomendable implementar metodologías para el correcto funcionamiento y fiel cumplimiento de los procesos.

Entre las metodologías más usadas internacionalmente tenemos ITIL, COBIT, CMMI, PMBOK, entre otros.

Estos métodos no implican una nueva manera de pensar y actuar, antes bien representan la manera correcta de ejecutar una acción además de proporcionar un marco en el cual se plantean procesos existentes y actividades en un contexto estructurado.

Para ello se deben distinguir procesos de 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 o específica y contribuye al cumplimiento de la misión general para cada tarea.

Considerando el punto de vista del negocio, el propósito de la gestión de la infraestructura de IT es optimizar la contribución y soporte de esta infraestructura para alcanzar sus metas de negocio. En aspectos de gestión de procesos de IT, las normas y marcos de referencia existentes dicen claramente “Qué Hacer”, mientras que la base de conocimientos ITIL , desarrolla también en detalle el “Cómo Hacerlo”.

ITIL, Information Technology Infrastructure Library es un set de documentos donde se describen los procesos requeridos para la gestión eficiente y efectiva de los Servicios de tecnologías de Información dentro de una organización. Son un conjunto de mejores prácticas y estándares en procesos para hacer más eficiente el diseño y administración de las infraestructuras de datos dentro de la organización. Es un “marco de trabajo” (framework) para la Administración de Procesos de TI.

Esta metodología se basa en la calidad de servicio y el desarrollo eficaz y eficiente de los procesos que cubren las actividades más importantes de las organizaciones. Garantizando así los niveles de servicio establecidos entre la organización y sus clientes.

El objetivo de ITIL es diseminar las mejores prácticas en la Gestión de Servicios de Tecnologías de Información. Esta metodología está especialmente desarrollada para reducir los costos de provisión y soporte de los servicios IT, al mismo tiempo de

garantizar los requerimientos de la información en cuanto a seguridad, mantienen e incrementan sus niveles de fiabilidad, consistencia y calidad.

ITIL brinda una descripción detallada de un número de prácticas importantes en TI, a través de una amplia lista de verificación, tareas, procedimientos y responsabilidades que pueden adaptarse a cualquier organización. ITIL describe una aproximación sistemática y profesional a la Gestión de Servicios TI. Haciendo énfasis en la importancia clave de cumplir con los requerimientos del negocio respetando los costos acordados.

ITIL representa una predisposición incondicional para orientarse al cliente y al servicio, es un prerrequisito. En muchas compañías esto requiere un cambio cultural predominante.

En resumen, con la ayuda de ITIL, se crea una terminología clara en el sector de la Gestión de Servicios de TI. Son un conjunto de mejores prácticas y estándares en procesos para hacer más eficiente el diseño y administración de infraestructuras de datos. Las normas ISO son demasiado rígidas para los negocios, pues lo que se ajusta bien a una empresa no lo hace a otra. En cambio, la incorporación de mejores prácticas es una forma sencilla de mejorar y estandarizar la calidad de los procesos corporativos. Las guías generales de mejores prácticas les sirven a todas las compañías, esta es una de muchas de las ventajas del uso de las metodologías ITIL. El planteo principal se basa en la calidad de servicio y el desarrollo eficaz y eficiente de los procesos.

La idea subyacente es que, sin importar el rubro, la tecnología es cada vez más crítica para el negocio de cualquier empresa. Esto quiere decir que si la tecnología no es administrada eficientemente, el negocio no funciona, lo que se vuelve más cierto al ser más dependiente de la infraestructura tecnológica. En este sentido, los

estándares ITIL exigen un replanteamiento del área tecnológica y la definición de los elementos y procesos "críticos" dentro de la empresa.

Esta metodología está especialmente desarrollada para reducir los costos de provisión y soporte de los servicios de TI, al mismo tiempo que se garantizan los requerimientos de la información en cuanto a seguridad manteniendo e incrementando sus niveles de fiabilidad, consistencia y calidad.

Las normas ISO son demasiado rígidas para los negocios, ya que lo que se ajusta bien a una empresa no lo hace a otra. En cambio, la incorporación de mejores prácticas (ITIL) es una forma sencilla de mejorar y estandarizar la calidad de los procesos corporativos. Las guías generales de mejores prácticas les sirven a todas las compañías.

La filosofía ITIL adopta la gestión de procesos y considera que, para lograr los objetivos claves de la Administración de Servicios estos procesos deberían ser usados por las personas y las herramientas efectiva, eficiente y económicamente en el desarrollo de la alta calidad y la innovación de los servicios de TI alineados con los procesos de negocio.

Los estándares ITIL exigen un replanteamiento del área tecnológica y la definición de los elementos y procesos "críticos" dentro de la empresa.

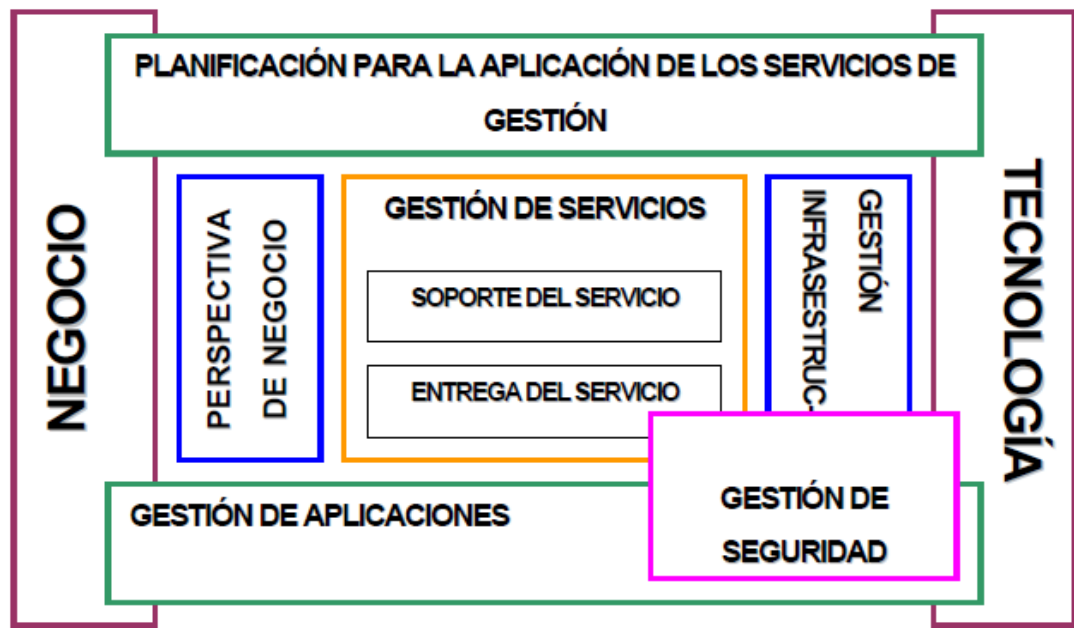


Figura 1: Estándar ITIL. Gráfico de procesos de soporte y entrega de sistemas ITIL

Las principales áreas de ITIL se dividen en diez procesos de soporte y entrega que debe incluir una estrategia de sistemas que ITIL describe muy claramente para cada uno, que son:

Soporte de Servicio:

Administración de incidentes (incident management)

Administración de problemas (problem management)

Administración del cambio (change management)

Administración de la configuración (configuration management)

Administración de las liberaciones (release management)

Entrega de Servicio:

Administración del nivel de servicio (service level management)

Administración financiera (financial management)

Administración de la capacidad (capacity management)

Administración de la disponibilidad (availability management)

Administración de la continuidad (continuity management)

Ventajas y desventajas del uso de metodología ITIL

Ventajas:

Al incluir la propuesta de ITIL se da una mejor utilización de los recursos de la organización, definiendo claramente hacia dónde estos recursos deben ser dirigidos. De esta manera la empresa será más competitiva, porque estará en mejor posición para hacer cambios en su infraestructura de TI. Adicionalmente, ITIL optimiza la disponibilidad, confiabilidad y seguridad de toda la plataforma, especialmente de los servicios "de misión crítica", facilitando también el aprendizaje de experiencias previas, lo que elimina el trabajo redundante.

Por otra parte, los procesos y plazos de un proyecto se ven mejorados, porque estas metodologías involucran la definición de procedimientos estándares, ayudando a brindar así servicios que satisfagan las demandas del negocio, clientes y usuarios. Finalmente, los estándares ITIL ofrecen indicadores de desempeño demostrables, lo que, por ejemplo, facilita la justificación de incrementos de costo en calidad de servicio.

Los principales beneficios obtenidos por la implantación de la metodología ITIL son:

a) Para el negocio:

Incremento en la productividad del negocio: Mayor disponibilidad y fiabilidad de las tecnologías de Información.

Mejora continua en la calidad de la prestación del servicio de las Tecnologías de información, ya que, tiene en cuenta tanto las necesidades de la compañía como sus objetivos.

La reducción del riesgo de no cumplir los objetivos de negocio gracias a la capacidad de recuperación y a la consistencia de los servicios.

Mayor flexibilidad y en consecuencia un mejor alcance de las acciones de la organización frente a cambios del entorno y el mercado. Posicionándose así en un soporte fiable para el negocio.

Soporte para los procesos de negocios y las tareas de toma de decisiones de TI.

Mediante la puesta en marcha de servicios basados en principios metodológicos y de calidad acordes con los requerimientos presentes y futuros de la compañía. Mejora en la satisfacción de los clientes, ya que se les asegura una mejora en la calidad del servicio entregado. Además el servicio puede ser representativamente medido, evaluado y gestionado.

Definición de funciones, roles y responsabilidades en el sector de los servicios.

La posibilidad de auditar el cumplimiento de las mejores prácticas.

Mejora en la satisfacción de los empleados y reducción de fluctuaciones de nivel de personal.

Incremento cualitativo en la salud, la seguridad, la disponibilidad y el rendimiento de los servicios de ITIL.

b) Económicos:

Diseño de la infraestructura y servicios de las Tecnologías de Información a costos argumentados.

Reducción de los costos operativos de desarrollo, procedimientos e instrucciones de trabajo, al disponer, de un marco de trabajo definido.

Desventajas:

Su introducción puede llevar tiempo y bastante esfuerzo, y supone un cambio de cultura en la organización. Una introducción demasiado ambiciosa puede llevar a la frustración porque nunca se alcanzan los objetivos.

Si la estructura de procesos se convierte en un objetivo en sí misma, la calidad del servicio se puede ver afectada de forma adversa. En ese caso, los procedimientos se transforman en obstáculos burocráticos que tratan de evitarse en lo posible.

No hay progreso por la falta de comprensión sobre lo que deben dar los procesos, cuáles son los indicadores de desempeño, y cómo se controlan los procesos.

Una implementación exitosa implica el compromiso del personal de todos los niveles de la organización.

Dejar el desarrollo de las estructuras de proceso a un departamento de especialistas puede aislar al departamento de la organización y puede fijar una dirección no aceptada por los otros departamentos.

Si hay poca inversión en las herramientas de soporte, los procesos no harán justicia y el servicio no mejorará. Se pueden necesitar más recursos y más personal si la organización se encuentra sobrecargada con las actividades de rutina de la

Administración de Servicio TI.

Gestión de incidencia

Incidencia es cualquier evento que no forma parte de la operación estándar de un servicio y que causa, o puede causar una interrupción, o una reducción de la calidad del mismo.

➤ ¿Por qué es necesaria una herramienta de gestión de incidencias?

Todos los departamentos de TI atienden fallos en hardware o software, y otras peticiones de servicio como faltas de empleados, peticiones de información, cambios de clave, etcétera. Si esta labor de apoyo diario no se sistematiza se depende mucho de la capacidad de cada técnico y no se reutiliza todo el conocimiento empleado en resolver incidencias pasadas.

La Gestión de Incidencias puede ser la clave del éxito o fracaso de un proyecto.

Incidencias son aquellos hechos inesperados y anómalos que se presentan durante la realización de las actividades y tareas del proyecto, y que producen desviaciones en la planificación. Ejemplos de incidencias que se pueden presentar en un proyecto son los retrasos en la entrega de un software, fallos en la infraestructura de desarrollo, enfermedad de alguien del equipo de proyecto, etc.

El uso de una herramienta de Gestión de Incidencias tiene tres objetivos básicos:

- Minimizar los periodos de fuera de servicio.
- Registrar la información relevante de todas las incidencias.
- Incorporar las mejores prácticas del mercado de forma sistemática. La gestión de incidencias es uno de los procesos más importantes definidos por ITIL. Su objetivo

es restablecer el funcionamiento normal del servicio lo más rápidamente posible, y con el menor impacto sobre la actividad del negocio.

Los beneficios de una gestión eficaz de incidencias son:

- Reducción del impacto de las incidencias sobre la organización.
- Uso más eficiente de los recursos de personal.
- Usuarios más satisfechos.
- Mayor visibilidad del trabajo realizado.

➤ ¿Cómo ayuda la Gestión de Incidencias en la resolución de incidencias?

La Gestión de Incidencias es un instrumento para aquellas organizaciones que quieran incorporar las mejores prácticas en la gestión de incidencias. En la práctica esto puede:

- Registrar la incidencia: quién informa del problema, síntomas, equipo involucrado, etc.
- Clasificar la incidencia y asignar el trabajo a realizar a un grupo de soporte o a un técnico.
- Investigar la causa de la incidencia y compararla con otras incidencias parecidas.

Documentar la solución, anexar ficheros con información relacionada y cerrar la incidencia.

- Elaborar informes, que ayuden a conocer qué está sucediendo y a mejorar el proceso.

1.03 Matriz T de análisis de fuerzas

La siguiente matriz indica 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 una escala del 1 al 5.

Tabla 1

Matriz T del Problema Central

SITUACIÓN EMPEORADA	SITUACIÓN ACTUAL				SITUACIÓN MEJORADA
Clientes 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 en el manejo de incidencias.
Registro digital de la información.	4	3	3	4	Reducidas oportunidades de contar con reporte de incidencias.
Capacitaciones	2	3	2	3	Ausencia de personal

Nota.-

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.

El problemas que afronta la empresa es el manejo de las incidencias que se realizan de forma manual, así como también reportes, por tal motivo el sistema permitirá



optimizar procesos y actividades, además estabilizar y mejorar el ambiente de trabajo e impulsar 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 realizará en menor tiempo, disminuyendo la cantidad de errores.

Capítulo II: Involucrados

2.01. Requerimientos

2.01.01. Descripción del sistema actual

Actualmente la empresa “PARTNERS GROUP”, de aquí en adelante como “empresa” maneja el control de incidencias de forma manual, toda la información es almacenada en una hoja de cálculo de Microsoft Excel, generando tickets manuales que se envían a los clientes y al Help Desk lo cual no permite llevar un control del tiempo de solución, lo que trae como consecuencias un inadecuado seguimiento y control de incidencias.

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

Después de ser ingresada la notificación, ésta es analizada por el jefe de almacén quien establece tiempo de solución y dependiendo a la dificultad o tipo de incidencia se asigna un técnico calificado dependiendo de la marca del producto adquirido por el cliente.

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 atendido: Servicios profesionales, instalaciones, asistencia técnica, remplazo de partes y piezas, etc. Y es aquí donde radica el problema, ya que el técnico no reporta el incidente o problema quedando el proceso inconcluso.

2.01.02. Visión y alcance

El control de incidencias con la implementación de una mesa de ayuda, tiene como objetivo resolver, de la manera más rápida y eficaz posible, cualquier incidente que

cause una interrupción de cualquier tipo de servicio técnico relacionado a los productos comercializados por la empresa.

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

Sin embargo, es evidente, que existe una fuerte relación 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 para 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.

2.01.03. Entrevistas

Tabla 2

Entrevista a Gerente Operativo.

DISEÑO ENTREVISTA		
identificador: 001		
Preguntas	Objetivos	Análisis posterior
¿Cuál es la problemática que buscan solucionar con la implementación de una mesa de ayuda?	Determinar los problemas que se pretende solucionar	Obtener tiempos de respuesta por incidencia Usuarios con rol de Supervisión puedan visualizar la gestión de los gestores de la mesa de ayuda en tiempo real. Dashboard estadístico de Incidencias resueltas Se requiere que las personas que tendrán acceso a la información sean:
¿Quiénes tendrán acceso a la mesa de ayuda?	Obtener el listado de los usuarios que manejarán la mesa de ayuda	Gerentes (áreas de: Calidad, Operaciones, Sistemas) Clientes Supervisores Help Desk
¿Cómo le gustaría llevar control de la forma como se maneja el control de incidencias?	Conocer como desea llevar el control de incidencias	Tiempos de respuestas y solución por incidencia exactos Solución de quejas de forma ágil y oportuna Generar tickets por incidencia que se generaran estableciendo prioridades Gestionar niveles de proceso en el que se encuentra la incidencia (sean estas: iniciado, en proceso y cerrado o finalizado)
¿Cuál es la mayor dificultad en el proceso actual y que cosa piensan que puede ser cambiada para mejor?	Evaluar la situación diaria del procedimiento más complicado y demorado en el manejo de información	No se da seguimiento a las Incidencias que se presentan No se cumple con el SLA (acuerdo de nivel de servicio) establecido por la empresa Se debe establecer una metodología para el fiel cumplimiento de las actividades solicitadas (se propone ITIL o COBIT)
¿Qué recomienda que se deba mejorar en el proceso?	Determinar los puntos más críticos que actualmente manejan en la institución	Poder controlar el cumplimiento de las incidencias. Llevar de forma organizada la gestión que se está realizando. Dar seguimiento a las quejas de los clientes. Establecer una metodología para el cumplimiento de procedimientos.

2.01.04 Matriz de Requerimientos

Tabla 3

Matriz de requerimientos funcionales

Identificador	Descripción	Fuente	Prioridad	Tipo	Estado	Usuarios Involucrados
REQUERIMIENTOS FUNCIONALES						
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
RF004	Historial de incidencias	Operativo	Alta	Sistema	En revisión	Administración, Operaciones
RF005	Seguimiento de clientes	Servicio	Alta	Sistema	En revisión	Atención al cliente

Tabla 4 *Matriz de requerimientos no funcionales*

Identificador	Descripción	Fuente	Prioridad	Tipo	Estado	Usuarios Involucrados
REQUERIMIENTOS NO FUNCIONALES						
RF001	La interfaz debe ser de fácil manejo para los usuarios.	Usuario	Media	Sistema	En revisión	Centro de servicios, administración, empleados
RF002	Manejo de reportes	Usuario	Alta	Sistema	En revisión	Clientes, Administración
RF003	Exportar los reportes cualquier tipo de archivo	Usuario	Media	Sistema	En revisión	Clientes, Administración
RF004	Informes dinámicos	Usuario	Media	Sistema	En revisión	Clientes, Administración

2.01.05 Descripción Detallada

Tabla 5

Descripción detallada Requerimiento Funcional 001

Creación de incidencias		Estado	Análisis	
Creado por	Cristhian Vera		Actualizado por	Cristhian Vera
Fecha Creación	24/11/2014	Fecha de Actualización	24/11/2014	
Identificador	RF001			
Tipo de Requerimiento	Crítico		Tipo de Requerimiento	Funcional
Datos de Entrada	➤ Datos del cliente ➤ Tipificación, responsable ➤ Motivo y submotivo ➤ Fecha de gestión, estado de gestión ➤ Prioridad ➤ Observaciones			
Descripción	Una vez ingresada la información de la incidencia se genera un nuevo registro			
Datos de salida	Incidencia ingresada con éxito			
Resultados Esperados	Incidencia almacenada correctamente.			
Origen	Técnico			
Dirigido a	Control de servicios			
Prioridad	8			
Requerimientos Asociados	Ninguno			
ESPECIFICACIÓN				
Precondiciones	Para ejecutar el requerimiento el cliente debe estar creado			
Pos-condiciones	Se podrá realizar una búsqueda de la nueva incidencia.			
Criterios de Aceptación	Permite que el usuario pueda establecer sus búsquedas.			

Nota.-

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	
Creado por	Cristhian Vera		Actualizado por	Cristhian Vera
Fecha Creación	24/11/2014	Fecha de Actualización	24/11/2014	
Identificador	RF002			
Tipo de Requerimiento	Crítico		Tipo de Requerimiento	Funcional
Datos de Entrada	Datos del cliente Fecha de gestión Motivo y submotivo Prioridad Estado de gestión Observaciones			
Descripción	Una vez ingresada la información de la incidencia se enviara un técnico.			
Datos de salida	El técnico será el encargado de cambiar el estado si la incidencia ha sido resuelta,			
Resultados Esperados	Incidencia gestionada correctamente.			
Origen	Técnico			
Dirigido a	Control de servicios			
Prioridad	8			
Requerimientos Asociados	Ninguno			
ESPECIFICACIÓN				
Precondiciones	Para ejecutar el requerimiento la incidencia debe estar creada.			
Pos-condiciones	Se podrá cambiar el estado de la incidencia o asignar un nuevo responsable.			
Criterios de Aceptación	Permite que el usuario pueda establecer sus búsquedas.			

*Nota.-**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	Cristhian Vera		Actualizado por	Cristhian Vera
Fecha Creación	24/11/2014	Fecha de Actualización	24/11/2014	
Identificador	RF003			
Tipo de Requerimiento	Crítico		Tipo de Requerimiento	Funcional
Datos de Entrada	➤ Datos del cliente ➤ Historial de incidencias			
Descripción	Una vez ingresada la información de la incidencia se enviara un técnico			
Datos de salida	➤ Usuario ➤ Cliente ➤ Motivo ➤ Fecha ➤ Tiempo ➤ Responsable			
Resultados Esperados	Incidencia notificada con éxito.			
Origen	Técnico			
Dirigido a	Control de servicios			
Prioridad	8			
Requerimientos Asociados	Ninguno			
ESPECIFICACIÓN				
Precondiciones	Para ejecutar el requerimiento la incidencia debe estar creada.			
Pos-condiciones	Se podrá cambiar el estado de la incidencia			
Criterios de Aceptación	Permite que el usuario pueda establecer sus búsquedas.			

Nota.-

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
REQUERIMIENTOS NO FUNCIONALES						
RNF001	La interfaz debe ser de fácil manejo para los usuarios.	Técnico	Media	Usuario	En revisión	Administrador Centro de servicios

2.02. Mapeos de Involucrados

Los Empleados revisan, analizan, consultan, realizan los informes de forma manual, indican el procedimiento 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; mientras que la Entrega del Servicio (Service Delivery) se centra en la planificación y mejora a largo plazo de los mismos.

2.02 Mapeos de Involucrados



Figura 2: *Mapeo de Involucrados.* Analiza los actores involucrados y sus intereses.

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
Técnico	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.
Empleado	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.
Supervisor	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 informática en el control de incidencias.

2.04 Análisis de la matriz de involucrados

En la matriz se analizan 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 las incidencias.

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 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.

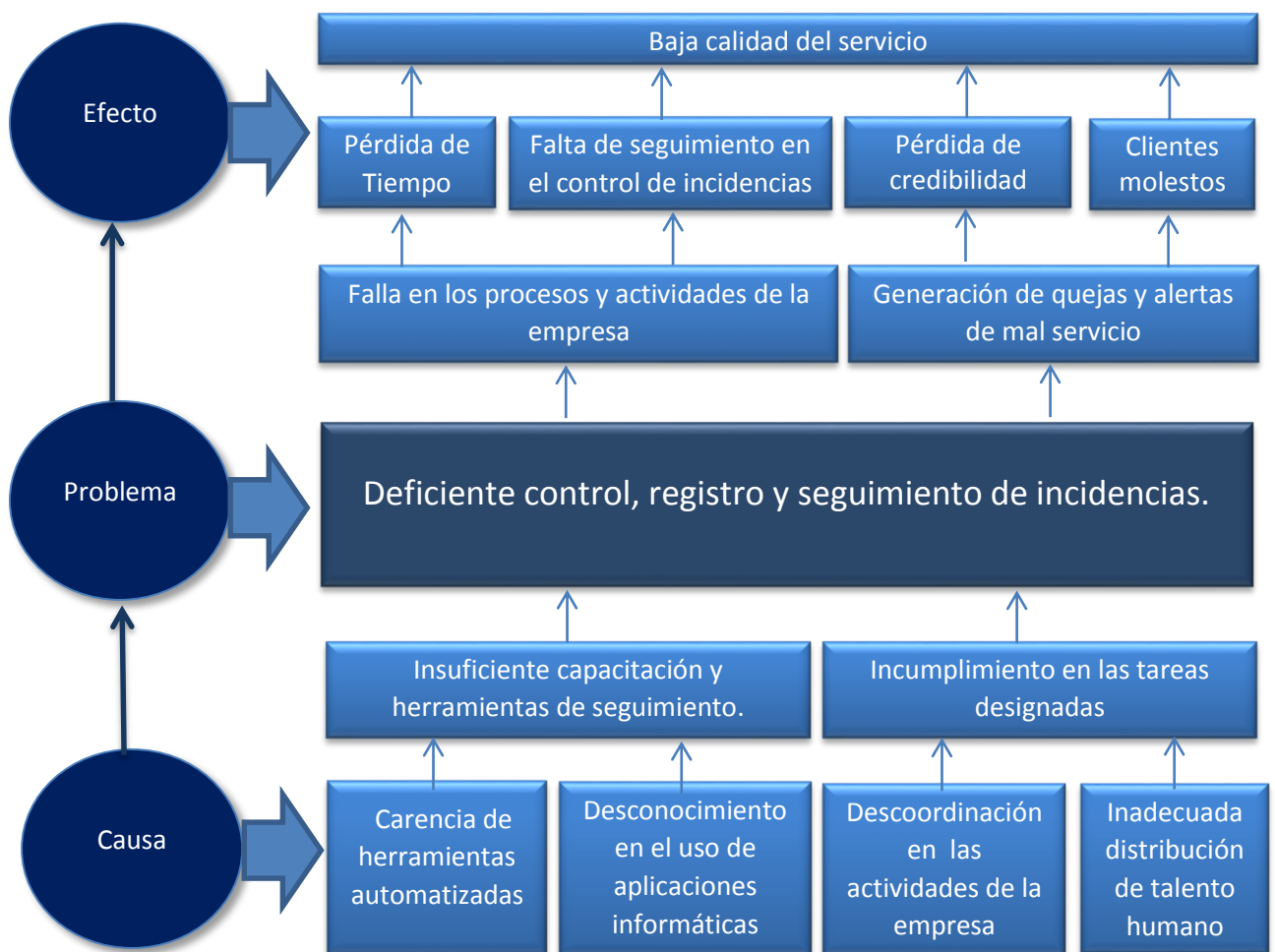


Figura 3: Árbol de problemas. Se describe los aspectos necesarios que nos permiten entender de mejor manera la problemática a resolver, listando las causas y sus efectos.

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 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.

3.02. 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.

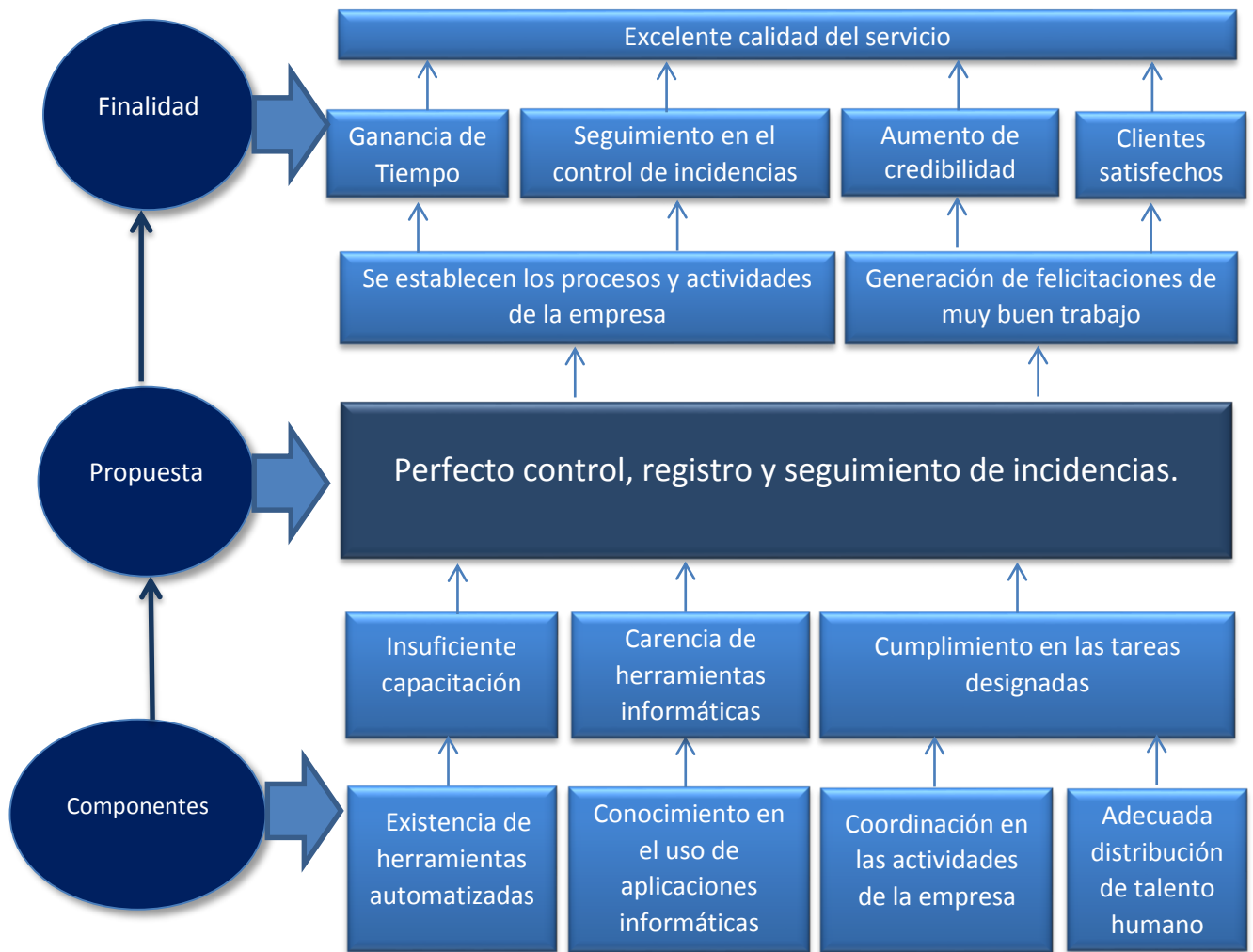


Figura 4: Árbol de Objetivos. Se describe la solución a todos los inconvenientes identificados en el árbol de problemas, convirtiendo a los componentes en medios para encarar el problema de desarrollo identificado

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.

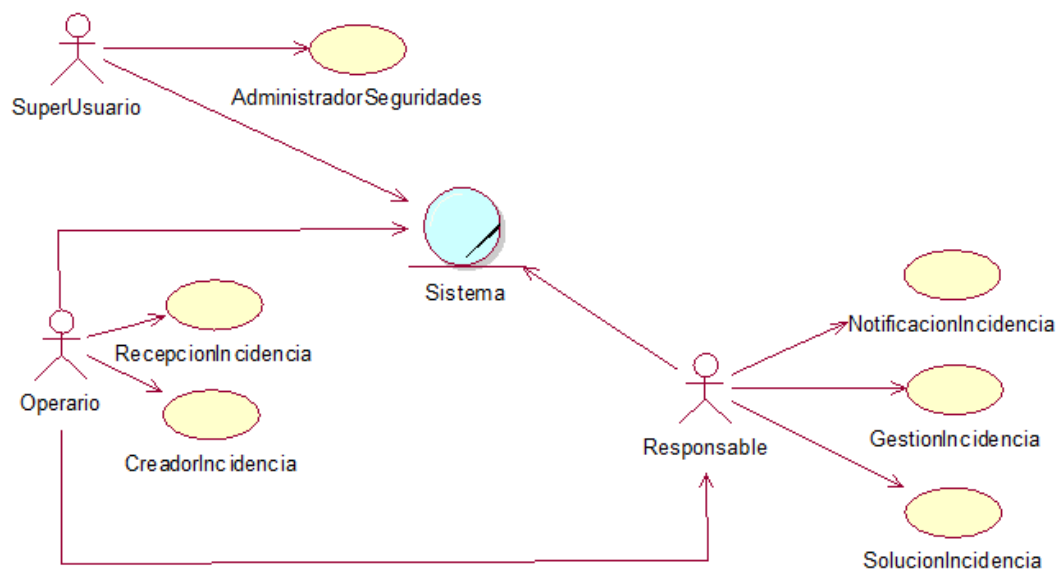


Figura 5: *Diagrama de Casos de uso.* Diagrama de comportamiento UML mejorado.

3.04. Casos de uso de realización

3.04.01. Diagrama de casos de uso de realización y especificaciones

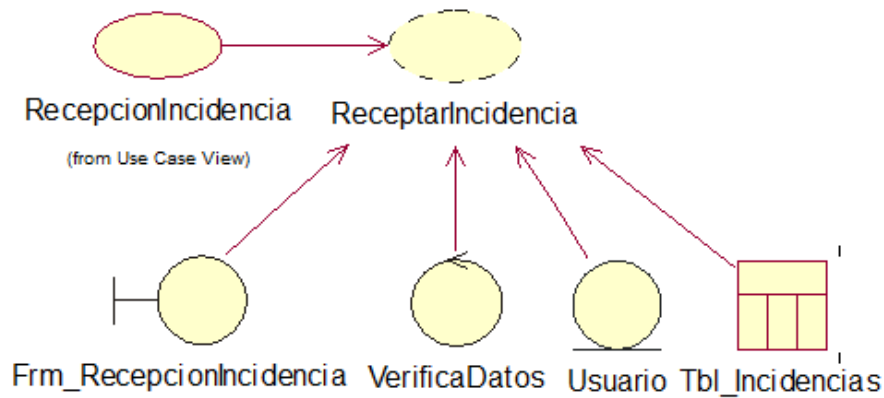


Figura 6: Diagrama de Casos de uso y especificación.

Tabla 10

Especificación del caso de uso de realización. Recepción de incidencia

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

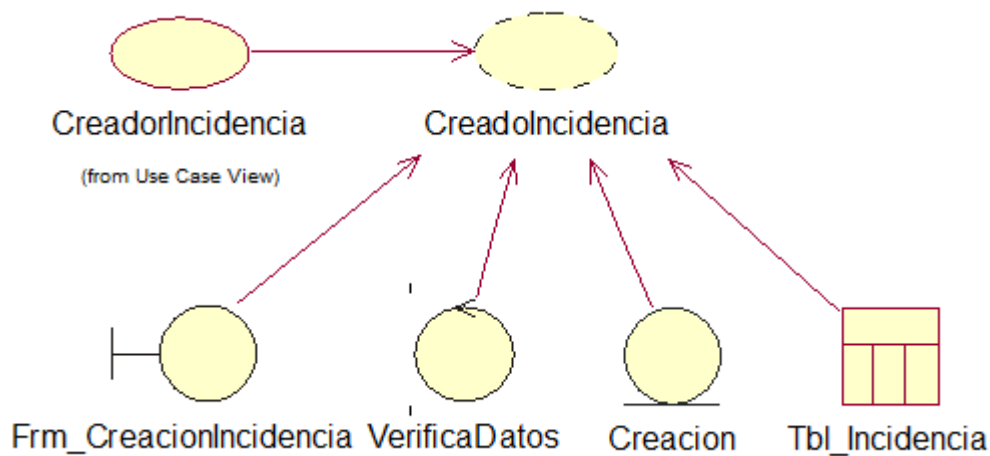


Figura 7: Casos de uso creación de incidencia

Tabla 11

Especificación del caso de uso de realización. Creación de incidencia

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

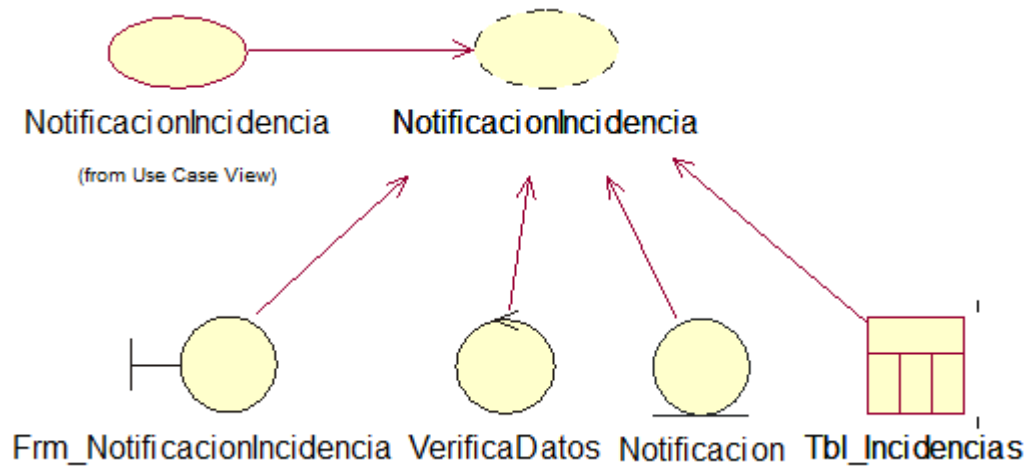


Figura 8: *Casos de uso notificación de incidencias.*

Tabla 12

Especificación del caso de uso de realización. Notificación de incidencia

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

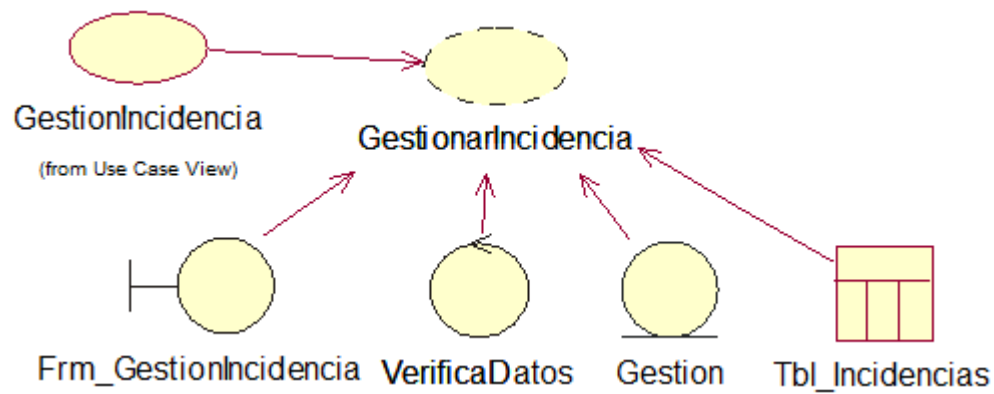


Figura 9: Casos de uso gestión de incidencia.

Tabla 13

Especificación del caso de uso de realización. Gestión de incidencia

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

3.04. Diagrama de secuencias del sistema

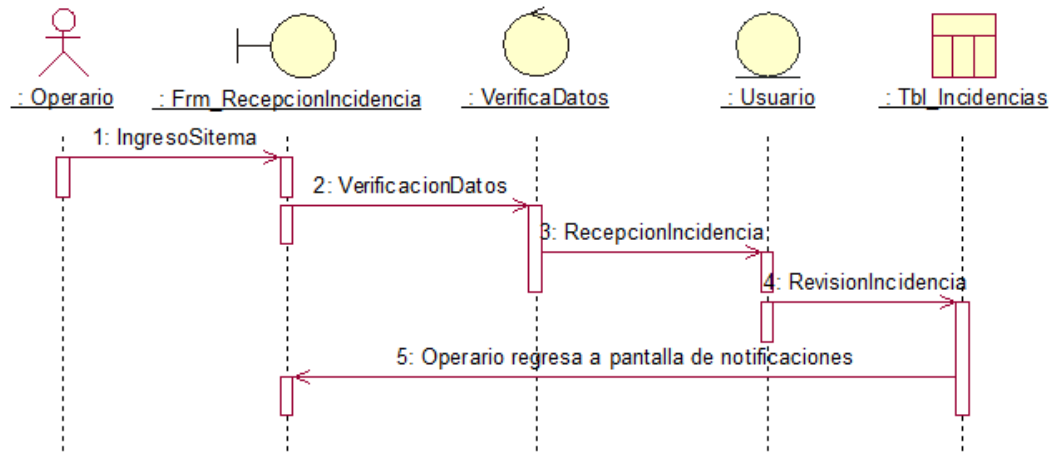


Figura 10: *Ingreso de Notificaciones.* Se puede ver el proceso de ingreso de incidencia.

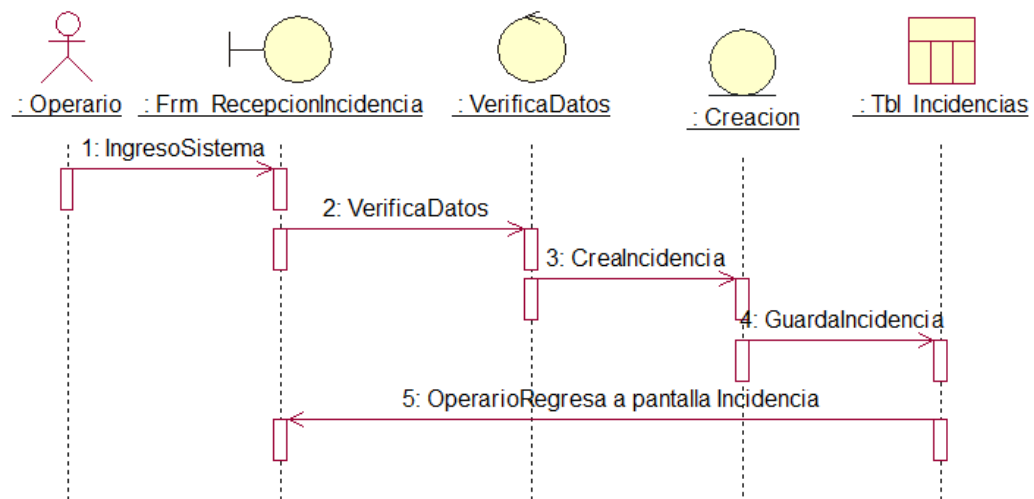


Figura 11: *Pantalla de Incidencia.* Se puede ver el proceso completo para regresar a nueva gestión (pantalla de incidencia)

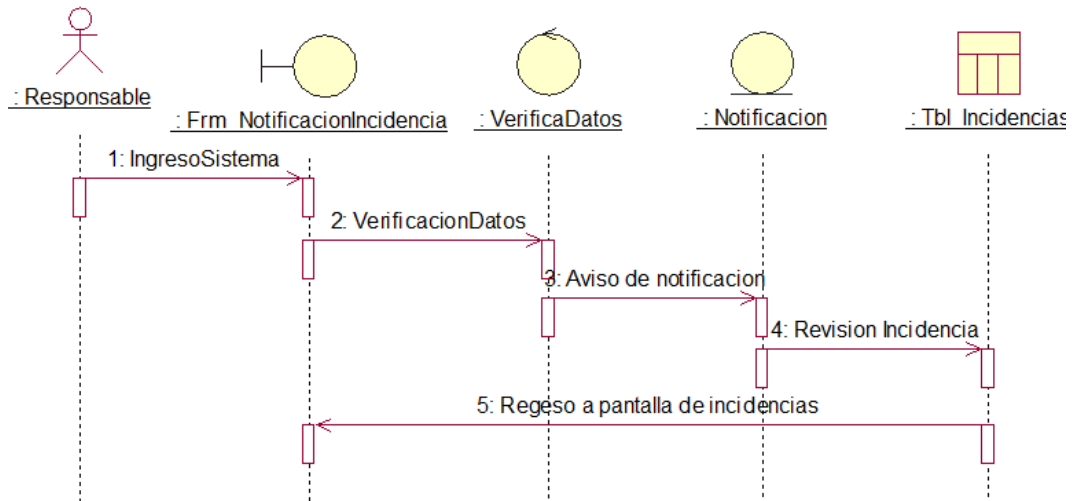


Figura 12: *Aviso de Notificaciones.* Se puede ver proceso de notificaciones.

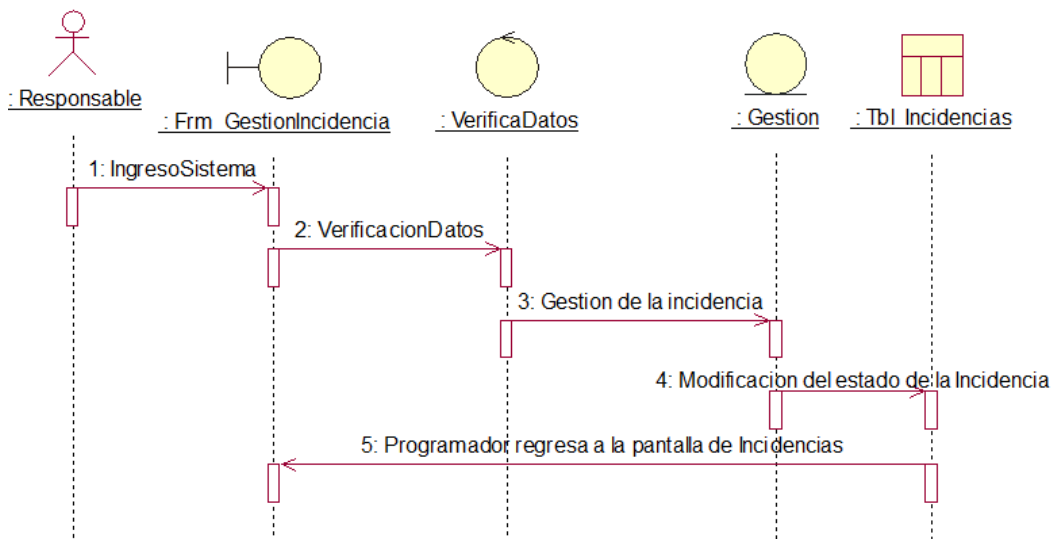


Figura 13: *Verificación de datos.* Se puede observar el proceso de verificación.

3.05.Especificación de los casos de uso

Tabla 14

Especificación de caso de uso. Receptar de incidencia

CASOS DE USO	Receptar Incidencia
IDENTIFICADOR	CU001

CURSO TÍPICO DE EVENTOS

USUARIO

Ingresa al sistema y verifica datos de recepción de incidencias

SISTEMA

Tipificación de incidencia
Motivo y posible submotivo
Fecha de gestión
Prioridad
Estado de gestión
Concepto
Observaciones

CURSOS ALTERNATIVOS

El operario debe haber recibido la incidencia

Tabla 15

Especificación de caso de uso. Crear incidencia

CASOS DE USO	Crear Incidencia
IDENTIFICADOR	CU002

CURSO TÍPICO DE EVENTOS

USUARIO

Verifica datos de creación de Incidencia

SISTEMA

Tipificación de incidencia
Motivo y posible submotivo
Fecha de gestión
Prioridad
Estado de gestión
Concepto
Observaciones
Guardar incidencia

CURSOS ALTERNATIVOS

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

Tabla 16

Especificación de caso de uso. Notificación de Incidencia

CASOS DE USO	Notificación de Incidencia
IDENTIFICADOR	CU003

CURSO TIPICO DE EVENTOS

USUARIO SISTEMA

El Responsable revisara la incidencia generada. Incidencia
Responsable de resolver incidencia

CURSOS ALTERNATIVOS

Si no recibe notificación deberá revisar la incidencia generada.

Tabla 17

Especificación de caso de uso. Gestionar incidencia

CASOS DE USO	Gestionar Incidencia
IDENTIFICADOR	CU004

CURSO TIPICO DE EVENTOS

Usuario Sistema

Se dará solución a la Incidencia

El Responsable deberá proceder a gestionar la incidencia

CURSOS ALTERNATIVOS

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

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	Categorí a
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
TOTAL	16	14	16	10	11	67	

Nota.-

0-5 =Bajo
6-10 =Medio Baja
11-15 = Media Alta
16- 20 = Alta

4.02. Matriz de Análisis de Impactos de los Objetivos.

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

Tabla 19

Matriz de análisis de alternativas.

AUMENTO 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>
	-Los beneficios son mayores que los costos.	-Responde a las expectativas de los beneficiarios	-Fortalece la participación de los empleados.	
	- Cuenta con financiamiento por parte de la empresa.	-Es una prioridad sentida por los beneficiarios	-Fortalece la Organización local	60 puntos
	-Protege el uso de los recursos.	-Beneficia a grupos de mayor carencia y vulnerabilidad	-Los beneficiarios están interesados en el proyecto	22 a 32 Baja
	-Es aceptable y conveniente para los beneficiarios	-Los beneficios son deseados por los beneficiarios	-La Empresa respalda el financiamiento del proyecto a futuro	33 a 44 Media baja
	-Existe tecnología adecuada para su realización			45 a 66 Media Alta
	-Se cuenta con soporte político-institucional.			67 a 88 Alta
	-Mejora el entorno social con los clientes.			
			16 puntos	
	28 Puntos	16 puntos		

4.03. 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:

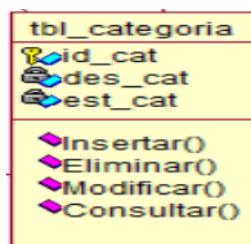


Figura 14: Clases UML. Se detalla las clases UML.

- ✓ **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** (+, ): Indica que el atributo será visible tanto dentro como fuera de la clase.
- ✓ **private** (-, ): Indica que el atributo sólo será accesible desde dentro de la clase.
- ✓ **protected** (#, ): Indica que el atributo no será accesible desde fuera de la clase.

Métodos:

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

- ✓ **public** (+, ): Indica que el método será visible tanto dentro como fuera de la clase.
- ✓ **private** (-, ): Indica que el método sólo será accesible desde dentro de la clase.
- ✓ **protected** (#, ): Indica que el método no será accesible desde fuera de la clase.

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).

4.04. Diagrama de clases

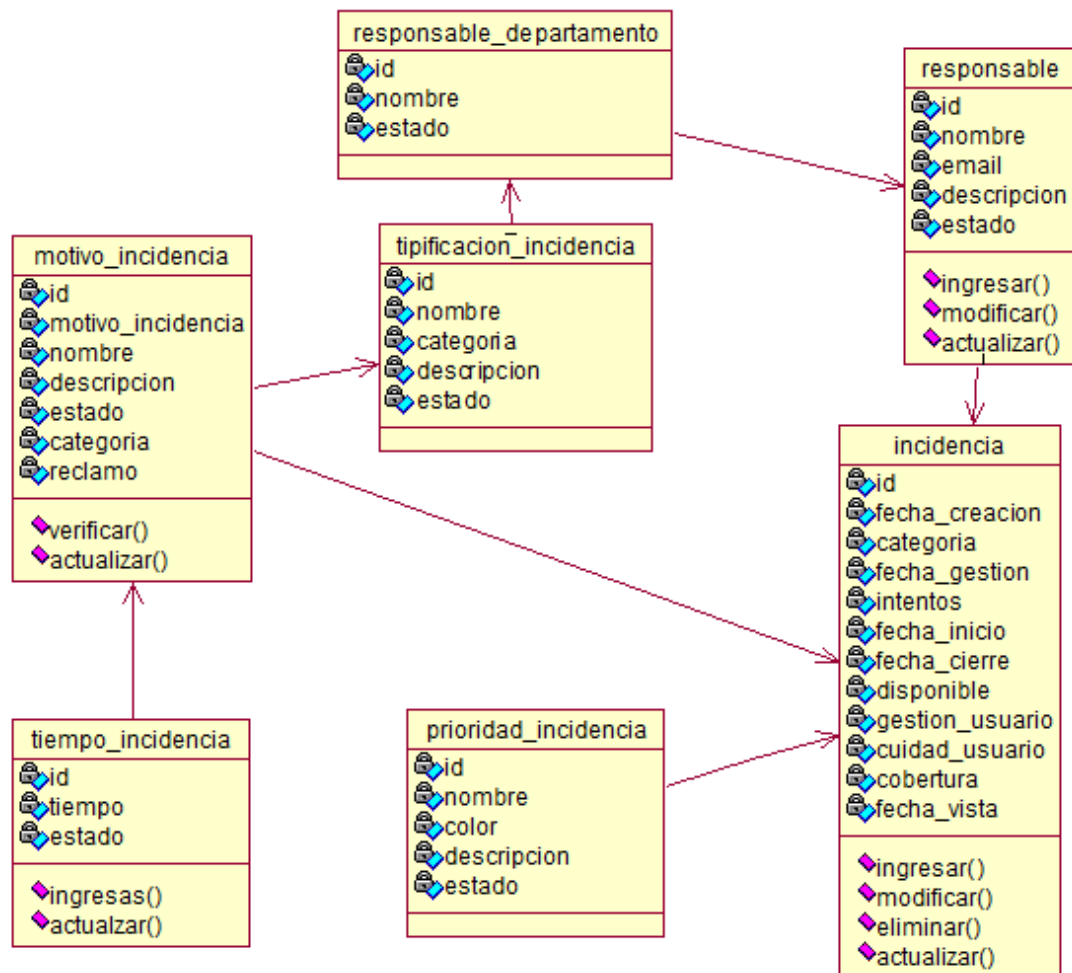


Figura 15: Diagrama de clases. Se muestran el diagrama de clases con sus respectivas validaciones.

4.06. Diagrama de Componentes

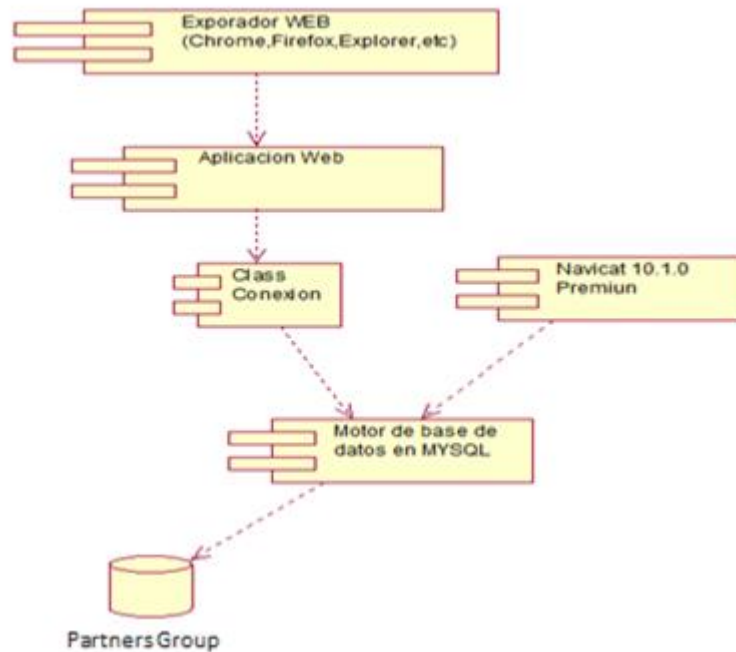


Figura 17: Capa de acceso de datos. Se muestran los medios por los cuales acceder.

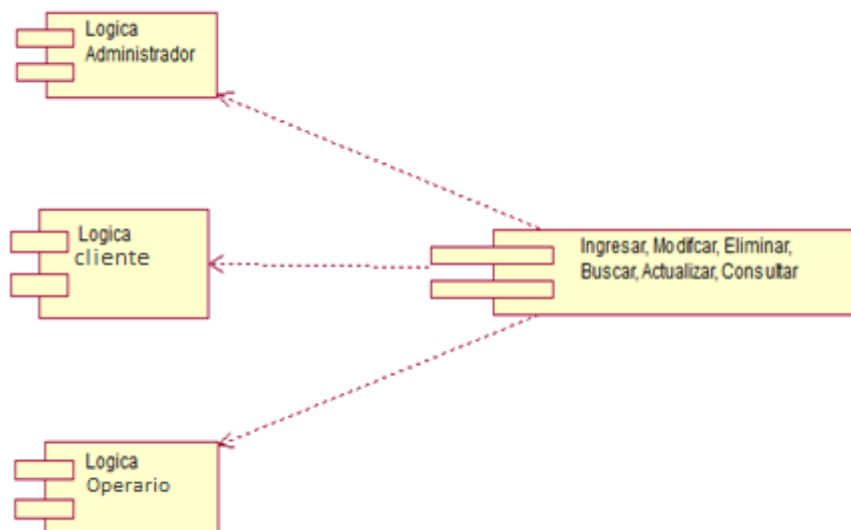


Figura 18: Capa lógica de negocios.

4.7. Diagrama de estrategias

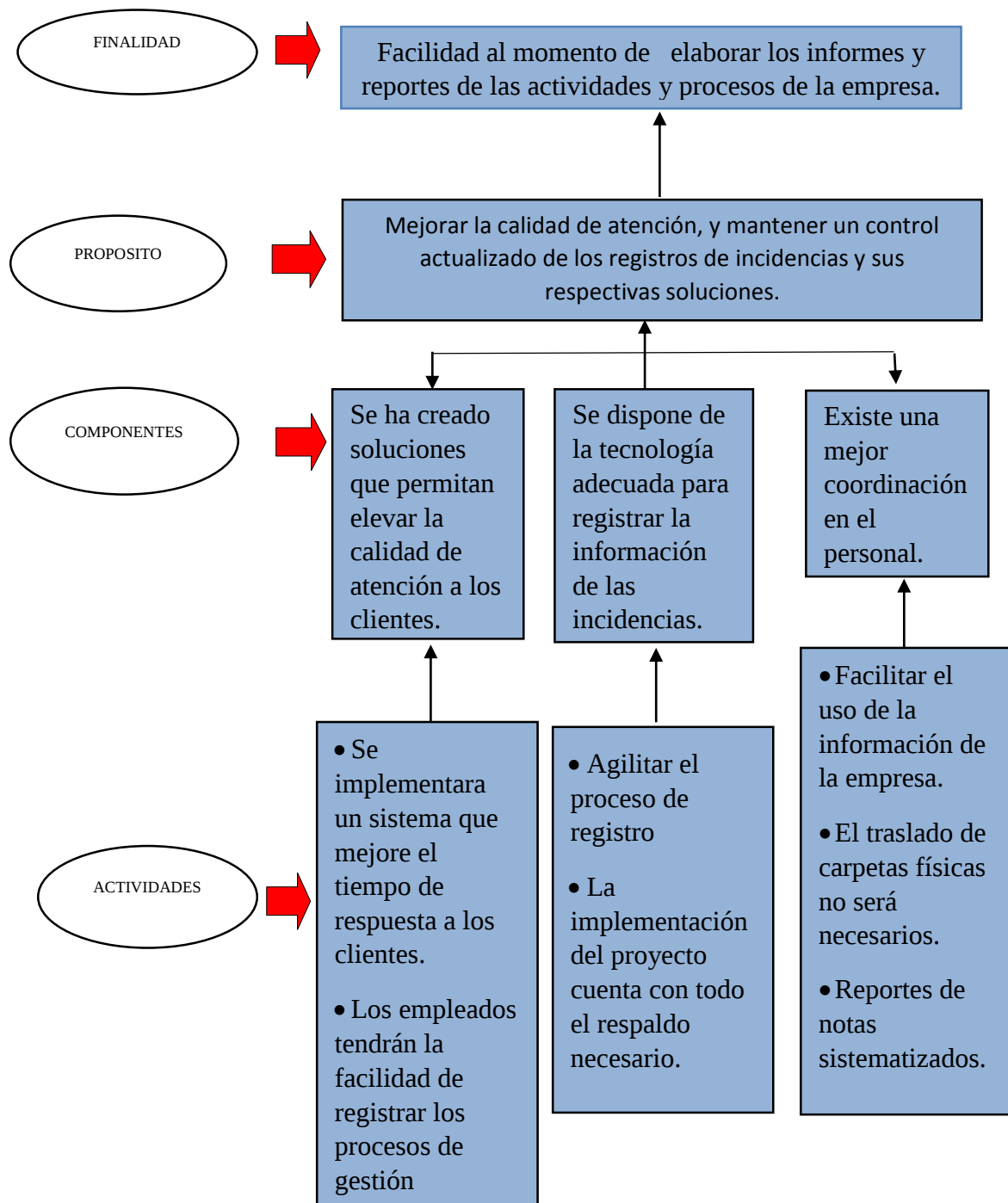


Figura 19: Diagrama de estrategias. Se muestra y se detalla la estrategia a seguir a fin de conseguir los objetivos planteados.

4.08. Matriz del Marco Lógico

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

Tabla 20

Matriz del Marco Lógico.

Resumen narrativo de objetos	Indicadores	Medios de verificación	Supuestos
Fin: Mejorar el manejo y proceso de incidencias	Obtener reportes ilustrativos y dinámicos que permitan visualizar el avance.	Fácil obtención de datos proporcionados desde el módulo del sistema.	Capacitación al personal de la empresa con la nueva aplicación.
Propósito: Disminuir el tiempo de respuesta en la solución de las incidencias.	Prevenir pérdidas.	Optimizar recursos y mejorar los procesos.	Mayor bienestar de los usuarios
Componentes: 1. Sistema amigable y atractivo para el usuario.	Realizar ensayos de calidad a la aplicación con distintos usuarios para verificar la complejidad.	Realizar modelos que muestre las mejoras que se está logrando	.
Actividad: 1. Analizar cómo funciona el proceso. 2. Levantar requerimientos. 3. Realizar BDD y desarrollar la aplicación.	Tener claras las reglas del negocio para realizar un buen funcionamiento de la aplicación.	Documentar todo los cambios del sistema.	El tiempo es demasiado corto para añadir módulos que van de la mano con el control de incidencias como son encuestas de satisfacción, seguimiento a entrega de documentos financieros (facturas), etc.

4.09 Vistas Arquitectónicas

La arquitectura seleccionada para el desarrollo del sistema informático de la Empresa "PARTNERS GROUP" ha sido la arquitectura de tres niveles o arquitectura tres capas.

4.09.01. Vista lógica

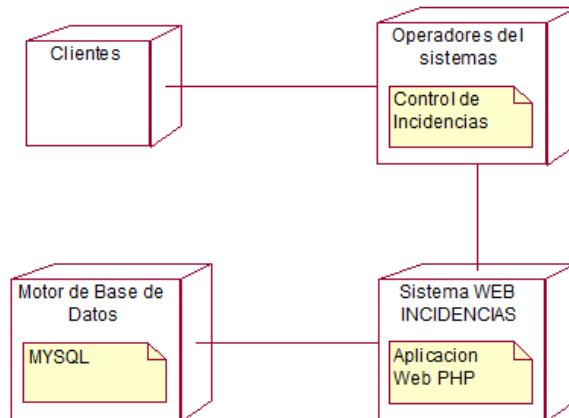


Figura 20: Vista lógica.

4.09.02. Vista física

En esta Figura se detalla la arquitectura en tres capas del sistema informático.

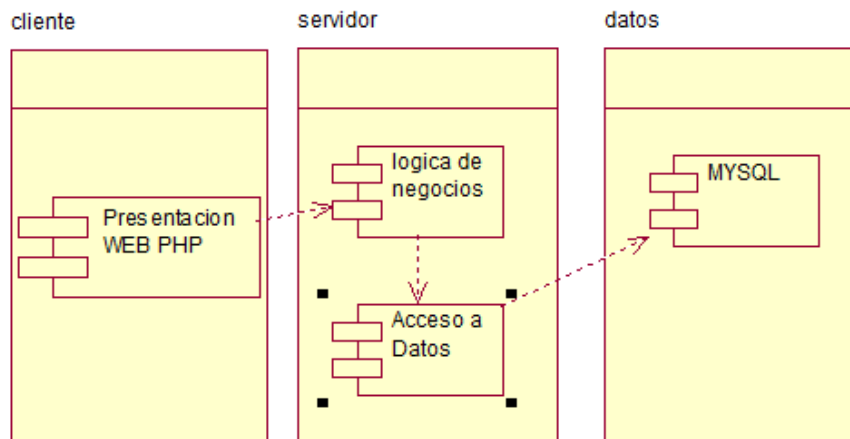


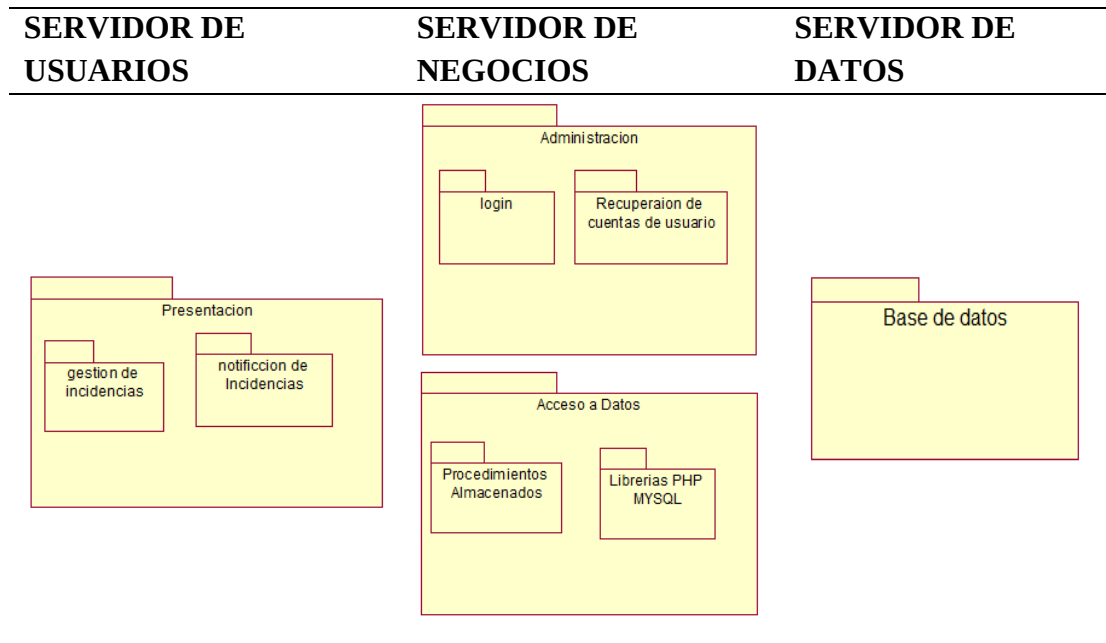
Figura 21: Vista Física.

4.09.03. Vista de desarrollo

Diagrama de paquetes del sistema informático

Tabla 21

Vista de desarrollo



4.09.04. Vista de procesos

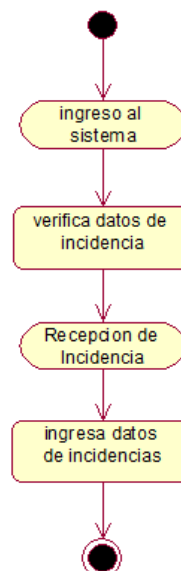


Figura 22: Diagrama de actividades del proceso de recepción de incidencia. Muestra las actividades que se realizarán dentro del sistema para la ejecución de cada proceso.

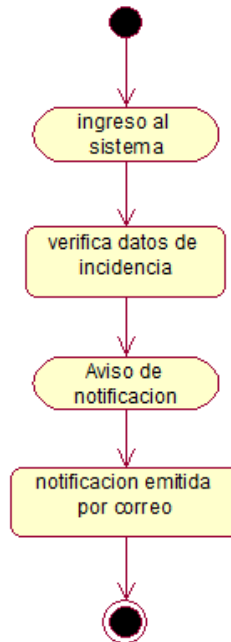


Figura 23: Diagrama de actividades de notificación de incidencia. Muestra las actividades que se realizarán dentro del sistema para la ejecución de cada proceso.

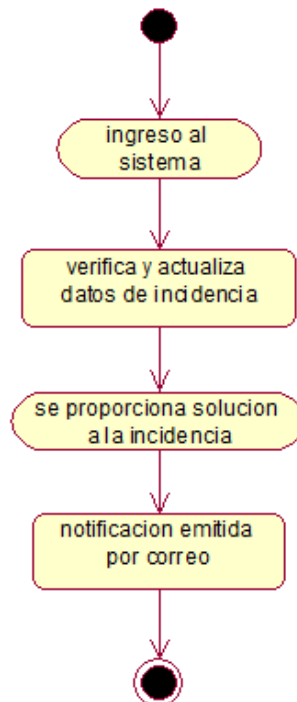


Figura 24: Diagrama de actividades del proceso de gestión de incidencia. Muestra las actividades que se realizarán dentro del sistema para la ejecución de cada proceso.

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.

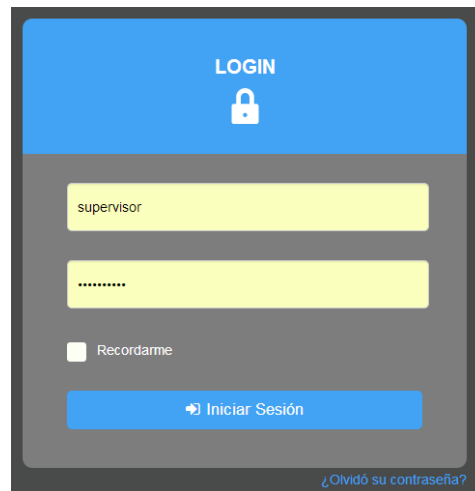


Figura 25: Restricción de Acceso. El usuario al momento de ingresar a la pantalla principal no podrá visualizar los módulos sin antes registrarse. Ya que el sistema le muestra una pantalla de error el mismo permitirá realizar sus funciones determinadas y con sus debidas restricciones de usuario.



	Rating	Nombre	Telefono	Industria	Tipo	Usuario Responsable	
☐	★ ★ ★ ★ ★	Cristhian Vera	022694065			admin	 

Figura 26: Búsqueda de cliente. El usuario al momento de ingresar podrá realizar la búsqueda de un cliente para crear la incidencia la búsqueda se podrá realizar por varios filtros.

Crear Cuenta

Los campos con * son obligatorios.

Nombre *

Ruc / Cédula

Celular

Email

Telefono

✓ Crear

✗ Cancelar

Figura 27: *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.

+ Crear

Administrar Incidencias

Q

Mostrar incidencias de estado

Numero Ticket	Cuenta	Resolución Estimada	Estado	Motivo	Submotivo	Vía de Ingreso	Prioridad	
0000001	Cristhian Vera	hace 4 semanas	Estado 1	Motivo 1	Submotivo 1	Via 1	Prioridad 1	
0000002	Cristhian Vera	hace 4 semanas	Estado 1	Motivo 1	Submotivo 1	Via 1	Prioridad 1	
0000003	Cristhian Vera	hace 4 semanas	Estado 1	Motivo 1	Submotivo 1	Via 1	Prioridad 1	

Viendo 1-3 de 3 resultados.

Figura 28: *Administrar motivos de incidencia.* El usuario al momento de ingresar podrá realizar modificación y creación de motivos de incidencia.

Registrar Llamada

Los campos con * son obligatorios.

Motivo

Submotivo *

Observaciones

Figura 29: Registró de llamadas. En esta vista se ingresaran un motivo y en caso de requerir una derivación (submotivo) más las observaciones del cliente.

Administrador de Usuarios ▾ Campos Personalizados ▾ Roles y Asignaciones ▾ Sistema ▾

Administrar Usuarios

Viendo 1-5 de 5 resultados.

Username	Correo	Estado de la cuenta	Ultimo acceso
user2	cvera@partners.com.ec	Cuenta Activada	18 Feb, 2015 07:02:44 pm
cvera	cvera@hotmail.com	Cuenta Activada	18 Feb, 2015 07:02:16 pm
supervisor	supervisor@gmail.com	Cuenta Activada	20 Mar, 2015 05:03:33 am
user	user1@hotmail.com	Cuenta Activada	18 Feb, 2015 07:02:23 pm
admin	admin@tucorreo.com	Cuenta Activada	19 Mar, 2015 10:03:58 pm

Figura 30: Administrar responsable de incidencia. El usuario al momento de ingresar podrá realizar modificación y creación de responsables de incidencia.

5.03. Especificación de pruebas de unidad

Tabla 22

Identificador de la prueba. Inicio de sesión.

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

Tabla 23

Identificador de la prueba. Crear cliente.

IDENTIFICADOR DE LA PRUEBA	PUC002
MÉTODO A PROBAR	Método "Crear Cliente"
Objetivo de la Prueba	Crear nuevo cliente correctamente
Datos de Entrada	
✓ Teléfono Titular , Teléfono 2 Titular	
✓ Email	
✓ Nombre Contacto, Apellido Contacto	
✓ Email Contacto	
Resultado Esperado	
El sistema verifique que los campos sean ingresados correctamente	
Si se encuentran vacíos, emitirá un mensaje de que los campos son obligatorios.	
Si los datos son incorrectos, emitirá mensajes de alerta.	
Comentarios	
El proceso funciona al 100% al momento de ejecutarlo	

Tabla 24

Identificador de la prueba. Crear incidencia.

IDENTIFICADOR DE LA PRUEBA	PUC003
MÉTODO A PROBAR	Método "Crear Incidencia"
Objetivo de la Prueba	Crear nueva incidencia correctamente
Datos de Entrada	
✓ Teléfono Titular	
✓ Email	
✓ Nombre Contacto, Apellido Contacto	
✓ Detalle	
✓ Motivo	
✓ Submotivo	
Resultado Esperado	
El sistema verifique que los campos sean ingresados correctamente	
Si se encuentran vacíos, emitirá un mensaje de que los campos son obligatorios.	
Notificar a los encargados	
Comentarios	
El proceso funciona al 100% al momento de ejecutarlo	

Tabla 25

Identificador de la prueba. Modificar.

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

5.04. Especificación de pruebas de aceptación

Tabla 26

Identificador de la prueba. Perfiles de usuarios (Administrador).

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

Tabla 27

Identificador de la prueba. Creación de incidencia (Administrador).

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

Tabla 28

Identificador de la prueba. Perfiles de usuario (Técnico)

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

5.05. Especificación de pruebas de carga

Tabla 29

Identificador de la prueba. Carga de usuarios.

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

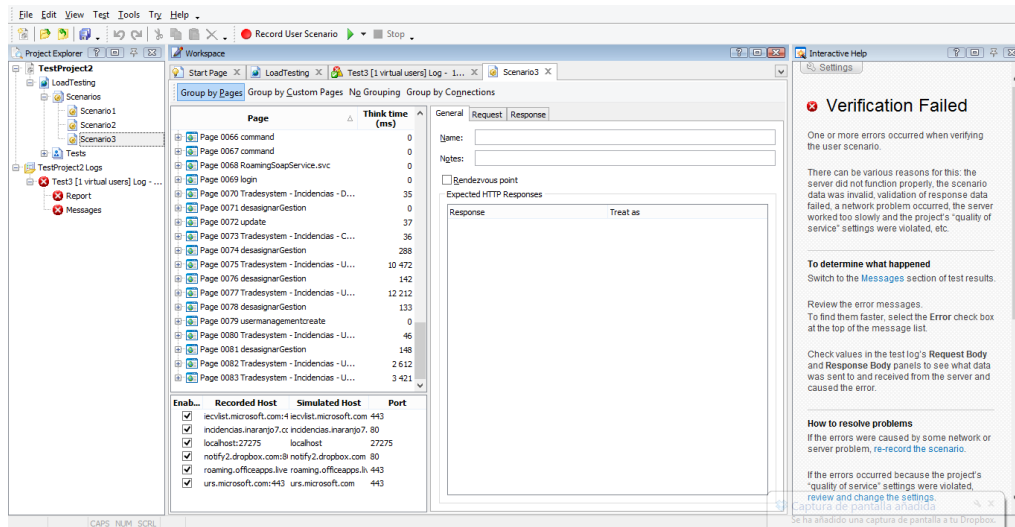


Figura 31: Pruebas de carga. En esta figura los formularios utilizados en las pruebas de carga.

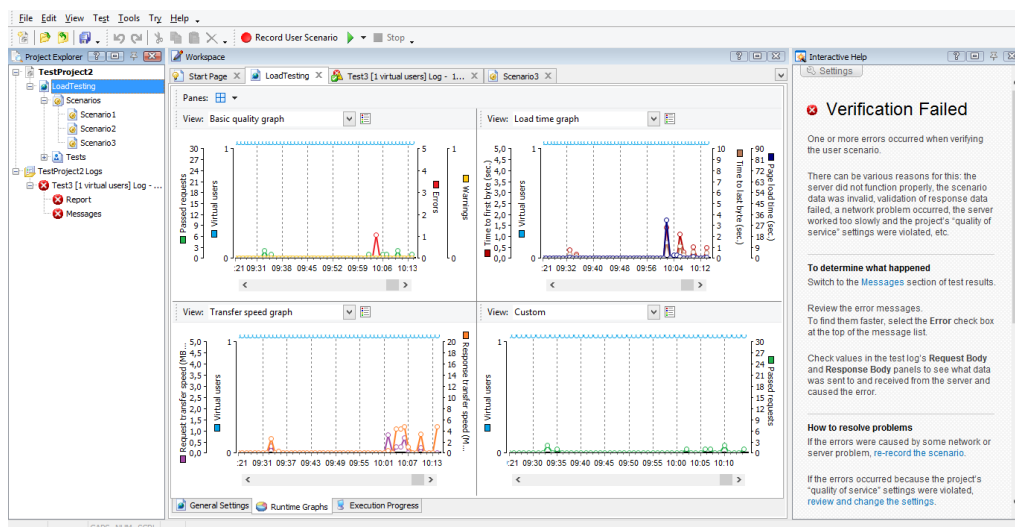


Figura 32: Pruebas de carga. En esta figura muestra el resultado de las pruebas de carga.

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, catálogo de pedidos, catálogo 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 2Duo 2.4 GHz .

RAM 2 GB mínimo

Disco duro 10 Gb espacio libre

EQUIPO CLIENTE

Procesador Core Duo 1.8 GHz

RAM 1 Gb mínimo

Disco duro 1 Gb espacio libre

5.06.04.02 Requerimientos de Software

EQUIPO SERVIDOR

Sistema operativo Windows Vista o superior.

Manejadores de base de datos MySQL 5.0 -

Browser Google Chrome, Internet Explorer, Mozilla Firefox.

EQUIPO CLIENTE

Sistema operativo Windows Vista o superior.

Browser Google Chrome, Internet Explorer, Mozilla Firefox.

CAPITULO VI

6.01. Recursos

Tabla 30

Recursos.

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

6.02 Presupuesto

Tabla 31

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
320	Transporte	\$ 0,25	\$ 80,00
2	Anillados	\$ 2,50	\$ 5,00
2	Empastados	\$ 3,50	\$ 7,00
6	Internet	\$ 21,00	\$ 126,00
1	Seminarios	\$ 720,00	\$ 720,00
1	Tutorías	\$ 200,00	\$ 200,00
160	Horas program.	\$ 20,00	\$ 3200,00
TOTAL			\$ 4.547,00

En esta tabla detallamos el presupuesto para el proyecto.

6.3 Cronograma de Actividades

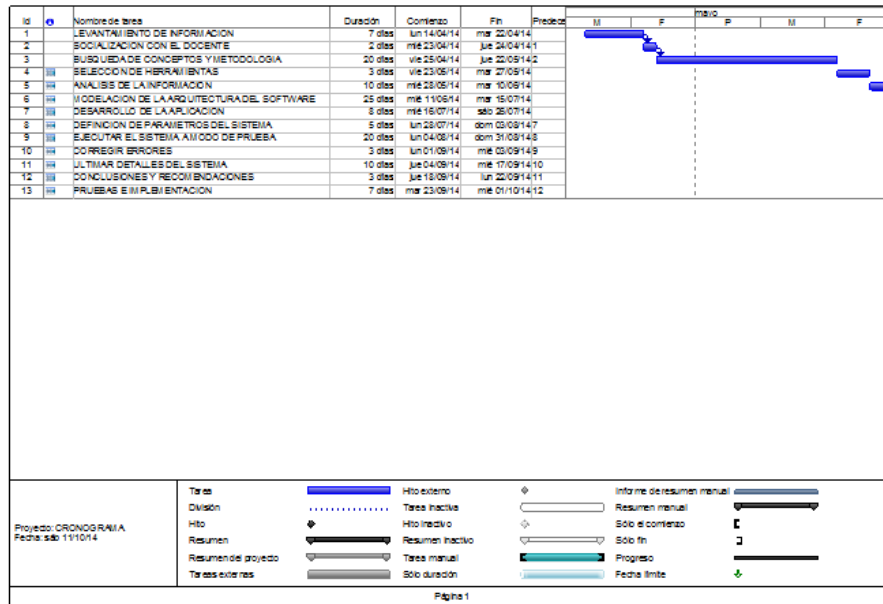


Figura 33: Cronograma de actividades. En esta figura se detalla la descripción y fechas de las actividades

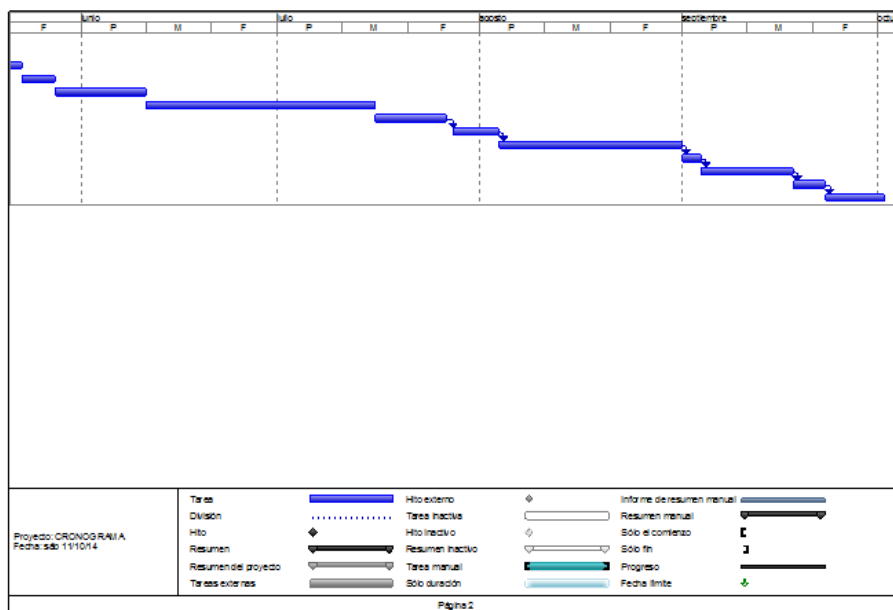


Figura 34: Cronograma de actividades. En esta figura se detalla fechas de las actividades y diagrama de Gant

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 contarán 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

A.01 ENLACES

a. Información acerca de Arquitectura 3 capas.

<http://www.slideshare.net/Decimo/arquitectura-3-capas> (capa de negocio).

b. Procedimiento para la realización de pruebas de unidad (Guidelines for preparing articles for Cuban journal of engineering)

<http://www.informatica->

[juridica.com/trabajos/Procedimiento_para_la_realizacion_de_pruebas_de_unidad.asp](http://www.informatica-juridica.com/trabajos/Procedimiento_para_la_realizacion_de_pruebas_de_unidad.asp)

c. Gestión de Incidencias

http://itilv3.osiatis.es/operacion_servicios_TI/gestion_incidencias.php

d. Procesos implicados en la correcta Gestión de Incidencias

http://itilv3.osiatis.es/operacion_servicios_TI/gestion_incidencias/proceso.php

e. KMKey Help Desk

http://www.kmkey.com/productos/software_help_desk

f. Trabajar con el software de gestión de incidencias HelpDesk

<http://www.artologik.com/es/HelpDesk/Acerca-del-programa/Sobre-HelpDesk/Trabajar-con-HelpDesk.aspx>

g. Ventajas de HelpDesk

<http://www.artologik.com/es/HelpDesk/Sobre-HelpDesk.aspx>

A.02 MANUALES

A.02.01. Manual de Instalación

Índice de Figuras

Fig 1: <i>Setup</i>	66
Figura 2: <i>Contrato</i>	66
Figura 3: <i>Seleccionar carpeta donde se guardara</i>	67
Figura 4: <i>Instalación</i>	67
Figura 5: <i>Escoger Navegador</i>	68
Figura 6: <i>Configuración</i>	68
Figura 7: <i>Barra de inicio</i>	68
Figura 8: <i>Java JDK</i>	69
Figura 9: <i>Descargar NetBeans</i>	70
Figura 10: <i>Verificación</i>	71
Figura 11: <i>Debugger Command</i>	71

Instalación de WAMPSERVER

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

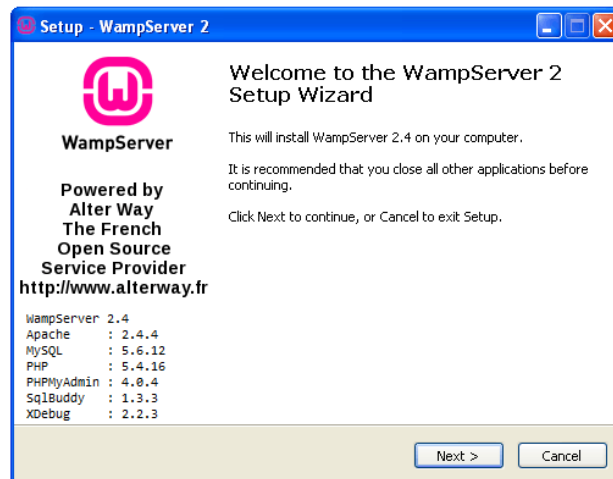


Figura 1: *Setup.*

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

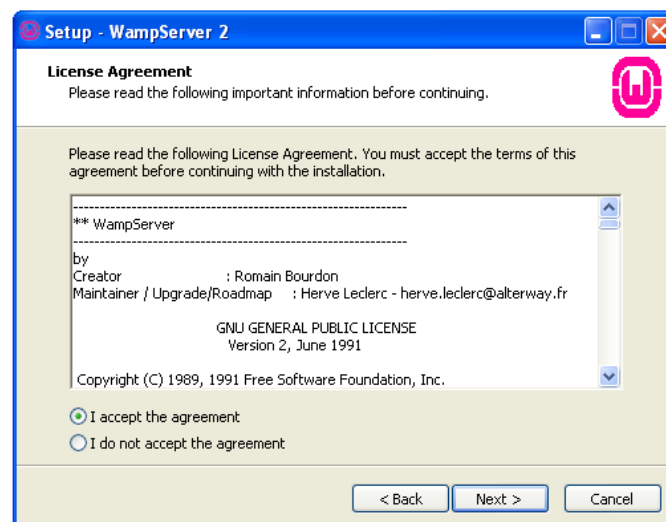


Figura 2: *Contrato.*

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

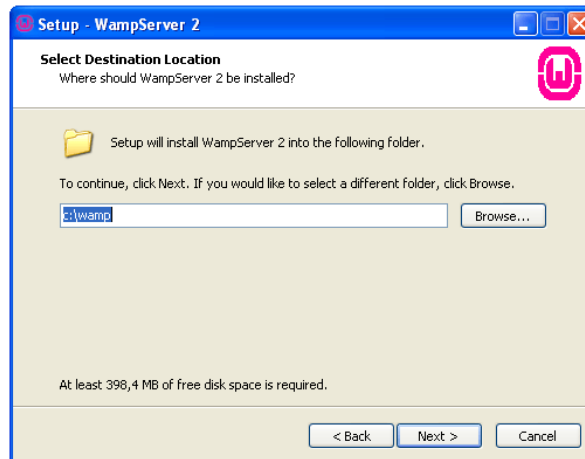


Figura 3: *Seleccionar carpeta donde se guardara*

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

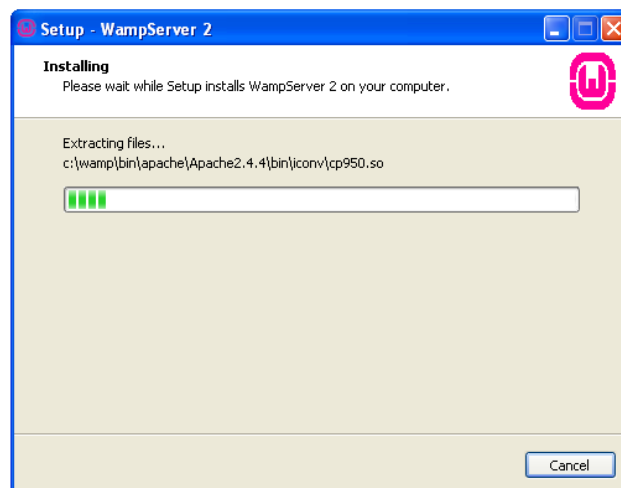


Figura 4: *Instalación.*

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

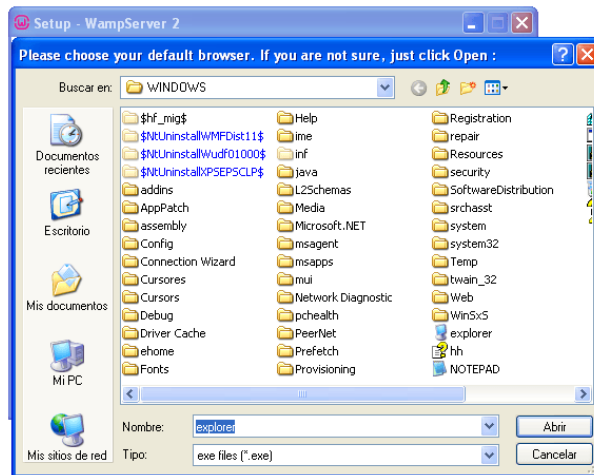


Figura 5: Escoger Navegador

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.

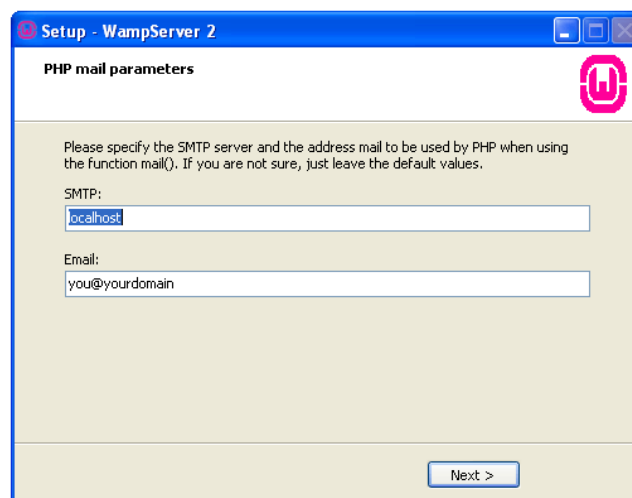


Figura 6: Configuración

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:



Figura 7: Barra de inicio

Este icono que aparece en la imagen anterior es el pilar fundamental de WAMP, si hacer click derecho sobre el 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 esta 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)

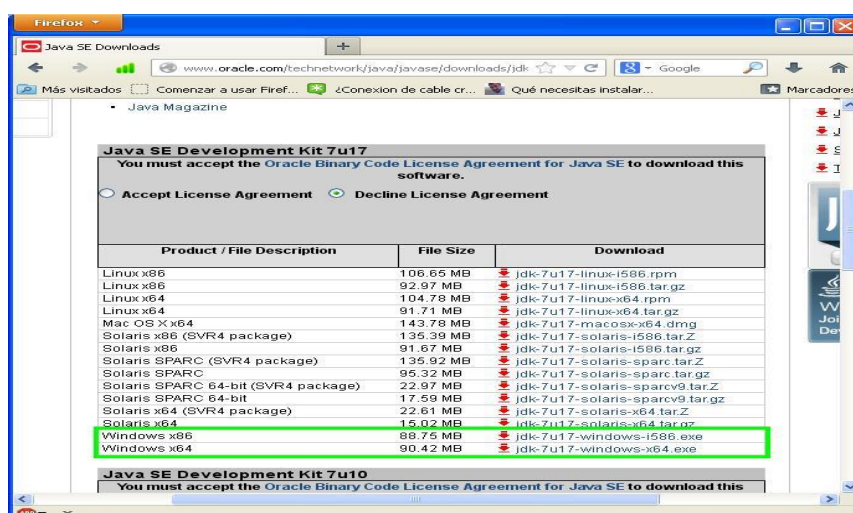


Figura 8: Java JDK

Descarga e instalar NetBeans.

En este paso nos dirigimos al siguiente enlace:

<https://netbeans.org/downloads/index.html>

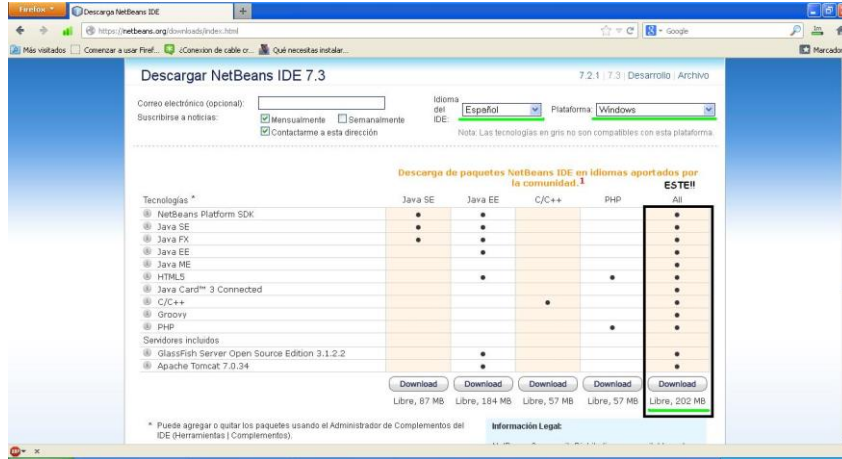


Figura 9: *Descargar NetBeans*

Comprobar que todo funcione.

Antes de nada advierto que este paso puede variar dependiendo del equipo y de cómo se haya completado los pasos,. 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.

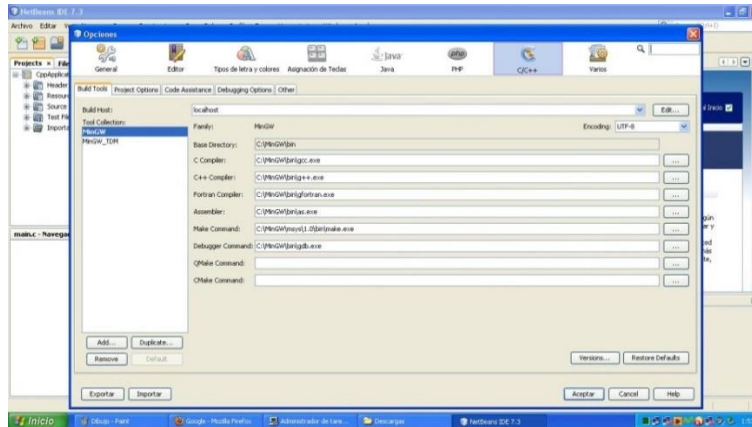


Figura 10: Verificación

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

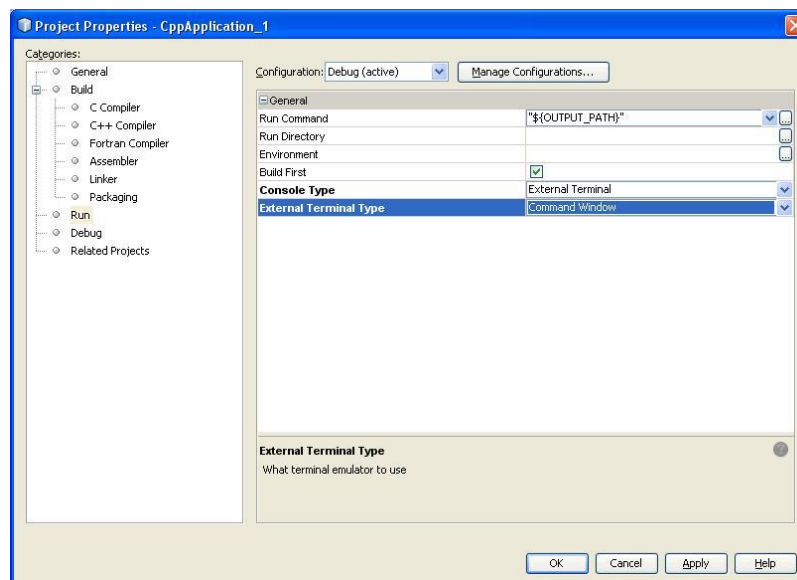


Figura 11: Debugger Command

A.02.02. Manual de usuario

Índice

Figura 1: <i>Login</i>	73
Figura 2: <i>Dashboard</i>	74
Figura 3: <i>Tipo de usuario</i>	74
Figura 4: <i>Menú</i>	75
Figura 5: <i>Administración de usuarios</i>	75
Figura 6: <i>Editar usuario</i>	76
Figura 7: <i>Registrar llamada</i>	76
Figura 8: <i>Administrador de Clientes</i>	77
Figura 9: <i>Información del cliente</i>	77
Figura 10: <i>Administrador de incidencias</i>	78
Figura 11: <i>Modulo de actividades</i>	78
Figura 12: <i>Bandeja de Incidencias</i>	79
Figura 13: <i>Ingreso de incidencias</i>	78

Ingreso al Sistema: el usuario realiza un ingreso de datos al login para ingresar al sistema todos los campos son obligatorios y tenemos la opción de recordarme para que en nuestro siguiente ingreso no sea necesario ingresar las credenciales nuevamente.

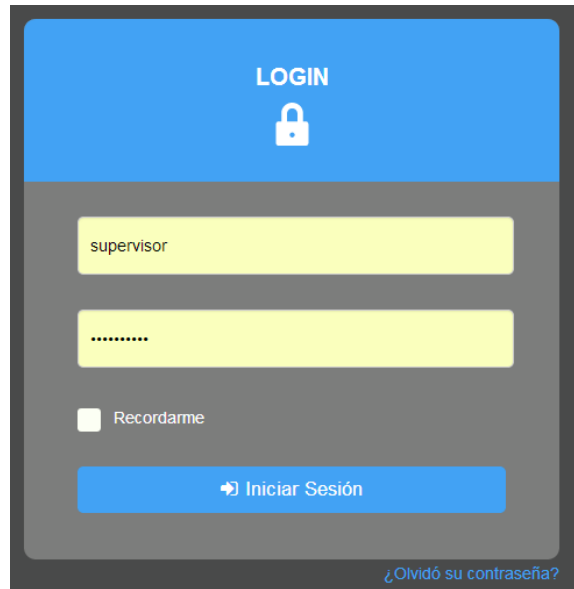
The image shows a login form with a blue header containing the word "LOGIN" and a lock icon. Below the header, there are two yellow input fields: the first contains the text "supervisor" and the second contains a series of dots representing a password. Under the password field is a checkbox labeled "Recordarme". At the bottom of the form is a blue button with a right-pointing arrow and the text "Iniciar Sesión". In the bottom right corner of the form area, there is a link that says "¿Olvidó su contraseña?".

Figura 12: *Login*

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.

Dashboard: Una vez dentro se nos desplegara información acerca de todas las actividades realizadas es una vista de la información estadística:

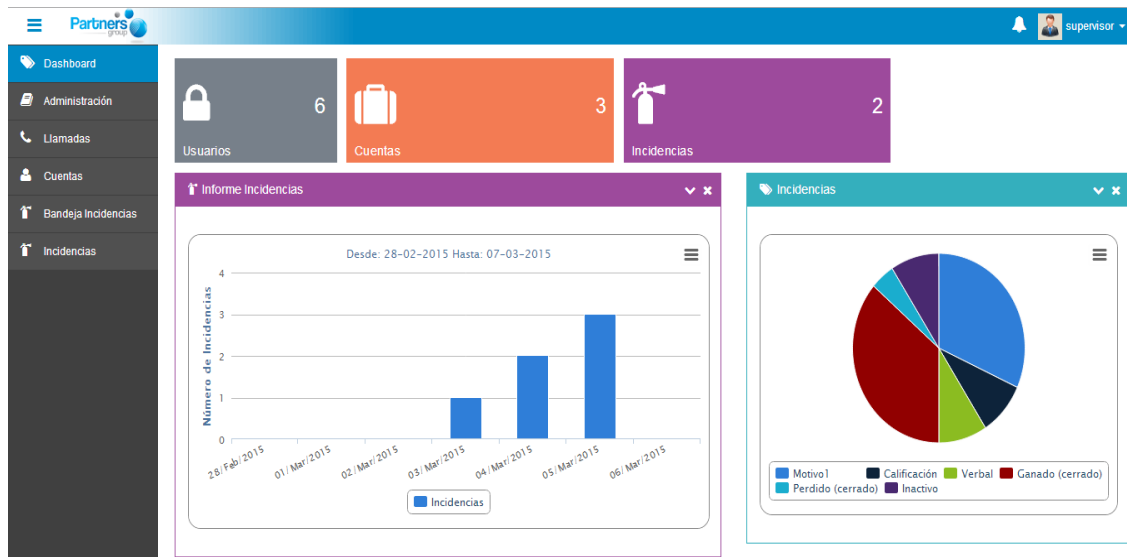


Figura 2: Dashboard

Además esta pantalla muestra lo siguiente:

Tipo de usuario:

Dependerá de los roles que se creen, en nuestro caso se usa SUPERVISOR ADMIN Y TECNICO (ojo esto es modificable dependerá del administrador)

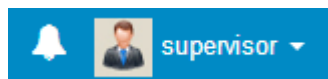


Figura 3: Tipo de usuario

Como se puede ver en la imagen anterior junto al tipo de usuario hay una campana la cual no alertara cada vez que exista una notificación (incidencias)

Menú:

En la parte lateral izquierda se encuentra nuestro menú de navegación el cual nos permitirá revisar cada una de las acciones que podremos realizar las mismas que serán analizadas más adelante

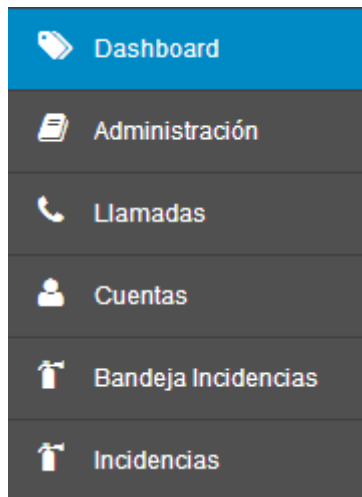


Figura 4: Menú

Administración: esta opción está habilitada solamente para los usuarios con permisos de administrador o supervisores

Administrar Usuarios

Viendo 1-6 de 6 resultados.

Username	Correo	Estado de la cuenta	Ultimo acceso	
user3	user3@hotmail.com	Cuenta Activada		
user2	cvera@partners.com.ec	Cuenta Activada	18 Feb, 2015 07:02:44 pm	
cvera	cvera@hotmail.com	Cuenta Activada	18 Feb, 2015 07:02:16 pm	
supervisor	supervisor@gmail.com	Cuenta Activada	30 Mar, 2015 04:03:32 am	
user	user1@hotmail.com	Cuenta Activada	18 Feb, 2015 07:02:23 pm	
admin	admin@tucorreo.com	Cuenta Activada	21 Mar, 2015 04:03:17 pm	

Figura 5: Administración de usuarios

En esta podemos visualizar todos los usuarios existentes y podemos aplicar las configuraciones y roles adecuados para cada uno de ellos como vemos en la siguiente imagen en la que hay 3 roles disponibles, en este caso estamos visualizando el de administrador quien tiene asignados todos los roles por ser el superusuario.

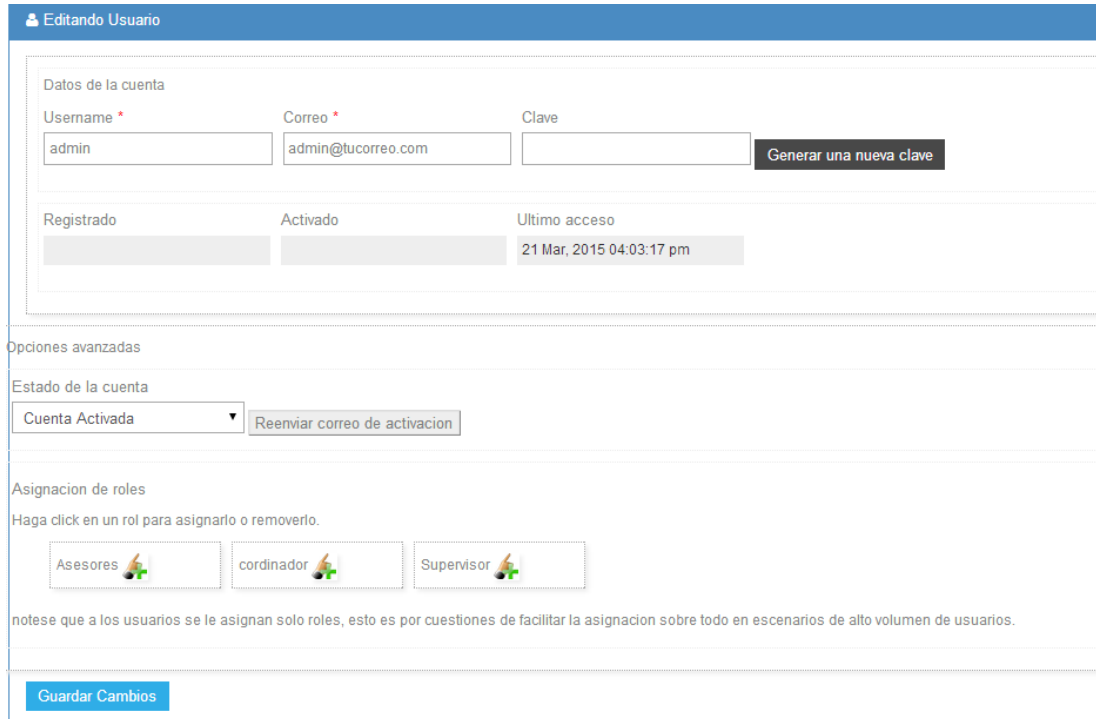


Figura 6: *Editar usuario*

Llamadas:

Existen ocasiones en la que los clientes no desean reportar incidencias y tan solo desean información es por eso que este sistema incluye un pequeño formulario para ingresar llamadas de este tipo.

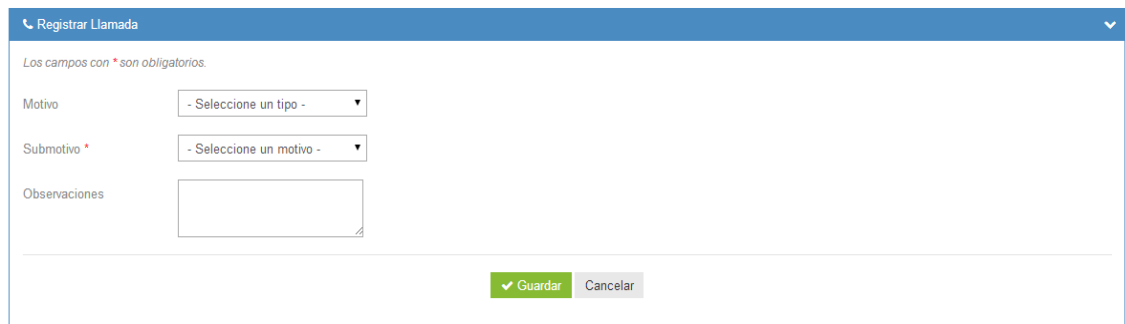


Figura 7: *Registrar llamada*

Cuentas:

En esta sección encontramos la lista de todos nuestros clientes lo cual nos permitirá de forma más fácil administrar nuestras incidencias.

+ Crear

Exportar

Administrar Cuentas

Q

☐	Rating	Nombre	Telefono	Industria	Tipo	Usuario Responsable	
☐	★ ★ ★ ★ ★	Cristhian Vera	022694065			admin	 
☐	★ ★ ★ ★ ★	cuebt 2				supervisor	 
☐	★ ★ ★ ★ ★	zdfxy578hi				supervisor	 

Viendo 1-3 de 3 resultados.

Figura 8: *Administrador de Clientes*

Estas cuentas pueden ser actualizadas cada vez que sea necesario además al ingresar a cada una ella encontraremos lo siguiente:

 Cristhian Vera

★ ★ ★ ★ ★

Información General

Nombre

Cristhian Vera

Ruc / Cedúla

1720457825

Celular

0995304118

Email

cvera@partners.com.ec

Telefono

022694065

Descripcion

ninguna

Creado por admin hace 5 meses

Actualizado por última vez por supervisor hace 5 horas

Actualizar

Notas

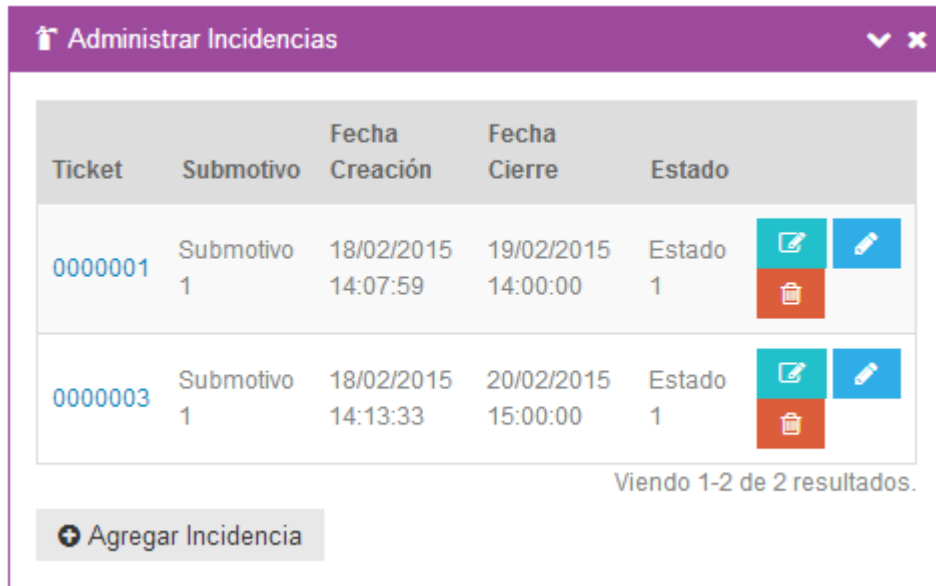
muchas pruebas

+ Add files

Guardar

Figura 9: *Información del cliente*

Información relacionada con este cliente además de que podemos agregar notas u observaciones según sea el caso además de una tabla en la que se visualiza un historial de las incidencias creadas por este cliente como se ve a continuación:



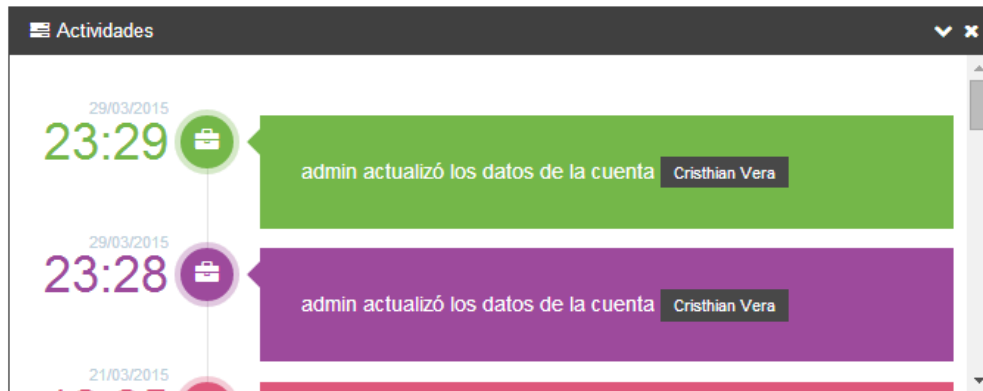
Ticket	Submotivo	Fecha Creación	Fecha Cierre	Estado	
0000001	Submotivo 1	18/02/2015 14:07:59	19/02/2015 14:00:00	Estado 1	 
0000003	Submotivo 1	18/02/2015 14:13:33	20/02/2015 15:00:00	Estado 1	 

Viendo 1-2 de 2 resultados.

 Agregar Incidencia

Figura 10: *Administrador de incidencias*

Finalmente tenemos la tabla de actividades en la cual se registra todas las actualizaciones modificaciones que se realicen con un registro de fecha para dar seguimiento:

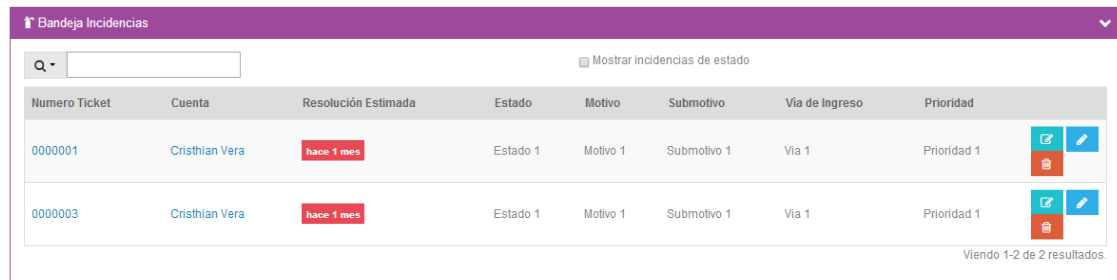


29/03/2015 23:29		admin actualizó los datos de la cuenta	Cristhian Vera
29/03/2015 23:28		admin actualizó los datos de la cuenta	Cristhian Vera
21/03/2015			

Figura 11: *Modulo de actividades*

Bandeja de Incidencias:

En esta última sección tenemos nuestra mesa de trabajo en esta encontraremos todas las incidencias que han sido creadas, en la misma se nos da una breve vista del cliente que lo solicita la fecha en que debe ser resuelta la incidencia, cual es el estado en el que se encuentra, el problema y la posible causa del inconveniente (posible porque es el técnico quien debe determinarla)



Numero Ticket	Cuenta	Resolución Estimada	Estado	Motivo	Submotivo	Vía de Ingreso	Prioridad
0000001	Cristhian Vera	hace 1 mes	Estado 1	Motivo 1	Submotivo 1	Vía 1	Prioridad 1
0000003	Cristhian Vera	hace 1 mes	Estado 1	Motivo 1	Submotivo 1	Vía 1	Prioridad 1

Viendo 1-2 de 2 resultados.

Figura 12: *Bandeja de Incidencias*

Luego que escojamos la incidencia que vamos a resolver e desplegara la siguiente vista:



Nueva Incidencia

Los campos con * son obligatorios.

Estado * Estado 1

Observaciones prueba 1

✓ Guardar ✕ Cancelar

Figura 13: *Ingreso de incidencias*

En esta deberemos cambiar el estado para dar por finalizado el proceso.

A.02.03. Manual técnico

CODIGO FUENTE REPORTE

Incidencias

```
<?php

$form = $this->beginWidget('ext.AweCrud.components.AweActiveForm', array(

    'type' => 'horizontal',

    'id' => 'reporte-form',

    'enableAjaxValidation' => true,

    'clientOptions' => array('validateOnSubmit' => true, 'validateOnChange' => false,),

    'enableClientValidation' => true,

    'htmlOptions' => array('class' => 'narrow'),

));

?>

<div class="widget blue">

    <div class="widget-title">

        <h4><i class="icon-fire-extinguisher"></i>Reporte Incidencias</h4>

        <span class="tools">

            <a href="javascript:;" class="icon-chevron-down"></a>

        </span>

    </div>

    <div class="widget-body">
```

```
<?php echo $form->errorSummary($model) ?>

<div class="row-fluid controles">

    <div class="control-label required">Filtro<span class="required">*</span></div>

    <?php

$this->widget('bootstrap.widgets.TbDateRangePicker', array(

    'model' => $model,

    'attribute' => 'fechas',

    'options' => array(

        'format' => 'dd/MM/yyyy',

        'ranges' => array(

            'Hoy' => array('today', 'today'),

            'Ayer' => array('yesterday', 'yesterday'),

            'Semana actual' => array('monday', 'today'),

            'Semana Anterior' => array(Util::calcularSemanaAnterior(0),
Util::calcularSemanaAnterior(1)),

            'Mes Actual' => array(Util::obtenerPrimerDiaMes(),
Util::obtenerUltimoDiaMes()),

            'Mes Anterior' => array(Util::calcularMesAnterior(0),
Util::calcularMesAnterior(1)),

        )

    ),

    'htmlOptions' => array(

        'readonly' => true,

        'class'=>'hasDatepicker'

    ),

));
```

```
?>

<input type="hidden" name="motivo" id="motivo"/>

<?php

$this->widget('bootstrap.widgets.TbButton', array(

    'buttonType' => 'submit',

    'type' => 'primary',

    'label' => Yii::t('AweCrud.app', 'Generar Reporte'),

));

?>

</div>

<?php if($visualizarReporte): ?>

<?php

if ($visualizarMenu) {

    echo $this->renderPartial('_menu_motivos', array(

        "botones" => $piereport['series']['0']['data'],

        ), TRUE, FALSE);

} else {

    $back = array(array('Regresar', " ", "));

    echo $this->renderPartial('_menu_motivos', array(

        "botones" => $back,

        ), TRUE, FALSE);

}

?>

<div class="row-fluid">

    <div class="row-fluid span6">

        <div class="span12 porcentaje" id="menormotivo">
```

```
<center>

    <h1>Total Incidencias: <?php echo $mayormotivo['total']; ?></h1>

</center>

</div>

<div class="span6 porcentaje segundalinea sinmargen" id="mayormotivo">

    <center>

        <h3>Mayor</h3>

        <h1><?php echo $mayormotivo['count']. '%'; ?></h1>

        <h4><?php echo Util::Truncate($mayormotivo['nombre'], 35); ?></h4>

    </center>

</div>

<div class="span6 porcentaje segundalinea" id="menormotivo">

    <center>

        <h3>Menor</h3>

        <h1><?php echo $menormotivo['count'] . ' %'; ?></h1>

        <h4><?php echo Util::Truncate($menormotivo['nombre'], 35); ?></h4>

    </center>

</div>

</div>

<div class="row-fluid span6" id="reportepie">

    <?php

        $this->Widget('ext.highcharts.HighchartsWidget', array(

            'scripts' => array(

                'modules/exporting', // adds Exporting button/menu to chart

            ),
```

```
'options' => $piereport

));

?>

</div>

</div>

<div class="row-fluid" id="reportebarras">

    <?php

        $this->Widget('ext.highcharts.HighchartsWidget', array(

            'scripts' => array(

                'modules/exporting', // adds Exporting button/menu to chart

            ),

            'options' => $columnreport

        ));

    ?>

</div>

<?php elseif (!$visualizarReporte && !empty($model->fechas)): ?>

    <div class="span12"></div>

    <div class="row-fluid"></div>

    <div class="alert alert-block alert-info fade in">

        <center>

            <h4 class="alert-heading">Alerta!</h4>

            <p>

                No existen datos entre las fechas indicadas

            </p>

        </center>
```

```
</div>
```

```
<?php endif; ?>
```

```
</div>
```

```
</div>
```

```
<?php $this->endWidget(); ?>
```

Controlador de Incidencias – Motivos

```
<?php
```

```
class IncidenciaController extends AweController {
```

```
/**
```

```
 * @var string the default layout for the views. Defaults to '//layouts/column2', meaning
```

```
 * using two-column layout. See 'protected/views/layouts/column2.php'.
```

```
 */
```

```
public $layout = '//layouts/column2';
```

```
public $defaultAction = 'admin';
```

```
public function filters() {
```

```
    return array(
```

```
        array('CrugeAccessControlFilter'),
```

```
);  
  
}  
  
/**  
  
 * Displays a particular model.  
  
 * @param integer $id the ID of the model to be displayed  
  
 */  
  
public function actionView($id, $tipo = null) {  
  
    //peticion de view por ajax  
  
    $model = $this->loadModel($id);  
  
    Yii::import("xupload.models.XUploadForm");  
  
    $archivos = new XUploadForm;  
  
    if (Yii::app()->request->isAjaxRequest && !isset($_GET['ajax'])) {  
  
        //envio de la vista parcial  
  
        $this->renderPartial('_view_modal', array(  
  
            'model' => $model, 'tipo' => $tipo,  
  
            ), false, true);  
  
    } else {  
  
        $this->render('view', array(  
  
            'model' => $model,  
  
            'archivos' => $archivos,  
  
        ));  
  
    }  
  
}  
  
/**
```

* Creates a new model.

* If creation is successful, the browser will be redirected to the 'view' page.

*/

```
public function actionCreate($contacto_id = null, $cuenta_id = null) {  
  
    //    die(var_dump($contacto_id,$cuenta_id));  
  
    $result = array();  
  
    $model = new Incidencia;  
  
    //    var_dump($_POST);die();  
  
    $model->owner_id = Constants::ID_MAIL_INCIDENCIAS;  
  
    $model->permisos = Incidencia::PERMISOS_ALL;  
  
    $model->usuario_creacion_id = Yii::app()->user->id;  
  
    $model->estado = Incidencia::ESTADO_ACTIVO;  
  
    $this->performAjaxValidation($model, 'incidencia-form');  
  
    if (Yii::app()->request->isAjaxRequest) {  
  
        $validadorPartial = false;  
  
        $model->contacto_id = $contacto_id;  
  
        $model->cuenta_id = $cuenta_id;  
  
  
        if (isset($_POST['Incidencia'])) {  
  
            $model->attributes = $_POST['Incidencia'];  
  
            //    $model->contacto_id    =    ($model->contacto_id    ==    '0')    ?    null    :  
$_POST['Incidencia']['contacto_id'];  
  
            //var_dump($model);  
  
            $result = $this->Crear($model);  
  
  
            if (!$result['success']) {
```

```
$result['mensaje'] = "Error al actualizar la fecha de la Incidencia";

}

$validadorPartial = TRUE;

echo json_encode($result);

}

if (!$validadorPartial) {

    $this->renderPartial('_form_modal', array('model' => $model,

        'cuenta_id' => $cuenta_id, 'contacto_id' => $contacto_id), false, true);

}

} else {

    if (isset($_POST['Incidencia'])) {

        $model->attributes = $_POST['Incidencia'];

        $result = $this->Crear($model);

        if ($result['success']) {

            $model = new Incidencia;

            $this->redirect('admin', array('model' => $model));

        }

    }

    $this->render('create', array(

        'model' => $model,

    ));

}

}
```

```
/**

* Updates a particular model.

* If update is successful, the browser will be redirected to the 'view' page.

* @param integer $id the ID of the model to be updated

*/

public function actionUpdate($id, $tipo = null, $isEstado = false, $bandeja = false) {

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

    $model->incidencia_categoria_id      =      $model->incidenciaSubmotivo->incidencia_categoria_id;

    $asignadoactual = $model->usuario_actualizacion_id;

    $model->incidencia_motivo_id          =          $model->incidenciaSubmotivo->incidencia_motivo_id;

    $model->fecha_est_resolucion  =  Util::FormatDate($model->fecha_est_resolucion,
"d/m/Y");

    $this->performAjaxValidation($model, 'incidencia-form');

    $result = array();

    $asignadoactual = $model->owner_id;

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

        $validadorPartial = false;

        if (isset($_POST['Incidencia'])) {

            $model->attributes = $_POST['Incidencia'];

            // $model->contacto_id  =  ($model->contacto_id  ==  '0')  ?  null  :
            $_POST['Incidencia']['contacto_id'];

            $result = $this->Update($model, $asignadoactual);

            if (!$result['success']) {

                $result['mensaje'] = "Error al actualizar la fecha de la oportunidad";

            }

        }

    }

}
```

```
}

$validadorPartial = TRUE;

echo json_encode($result);

}

if (!$validadorPartial && !$sisEstado) {

    // var_dump($validadorPartial);

    $this->renderPartial('_form_modal', array('model' => $model, 'tipo' => $tipo), false,
true);

} else if ($sisEstado) {

    $this->renderPartial('_form_modal_estado', array('model' => $model, 'tipo' =>
$tipo), false, true);

}

} else {

    if (isset($_POST['Incidencia'])) {

        $model->attributes = $_POST['Incidencia'];

        $result = $this->Update($model, $asignadoactual);

        if ($result['success']) {

            $model = new Incidencia;

            if (!$bandeja) {

                $this->redirect('admin', array('model' => $model));

            } else {

                $this->redirect('bandejaIncidencias', array('model' => $model));

            }

        }

    }

}
```

```
}

$this->render('update', array(

    'model' => $model,

));

}

}

/**

 * Deletes a particular model.

 * If deletion is successful, the browser will be redirected to the 'admin' page.

 * @param integer $id the ID of the model to be deleted

 */

public function actionDelete($id) {

    if (Yii::app()->request->isPostRequest) {

        // we only allow deletion via POST request

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

        $FechaHoraEliminacion = Util::FechaActual();

        $estado = Incidencia::model()->updateByPk($id, array('estado' =>

Incidencia::ESTADO_INACTIVO,

        'usuario_actualizacion_id' => Yii::app()->user->id,

        'fecha_actualizacion' => $FechaHoraEliminacion));

        if ($estado) {

            Evento::model()-

>ActualizarEstadoEntidad(Evento::ENTIDAD_TIPO_INCIDENCIA, $id);
```

```
Tarea::model()-
>ActualizarEstadoEntidad(Tarea::ENTIDAD_TIPO_INCIDENCIA, $id);

    Nota::model()->ActualizarEstadoEntidad(Nota::TIPO_INCIDENCIA, $id);

}

// Crear registro de actividad

Actividad::registrarActividad($model, Actividad::TIPO_DELETE);

// if AJAX request (triggered by deletion via admin grid view), we should not redirect
the browser

if (!isset($_GET['ajax']))

    $this->redirect(isset($_POST['returnUrl']) ? $_POST['returnUrl'] : array('admin'));

} else

    throw new CHttpException(400, 'Invalid request. Please do not repeat this request
again.');
```



```
/**

 * Lists all models.

 */

public function actionIndex() {

    $dataProvider = new CActiveDataProvider('Incidencia');

    $this->render('index', array(

        'dataProvider' => $dataProvider,

    ));

}

/**
```

* Manages all models.

*/

```
public function actionAdmin() {

    $model = new Incidencia('search');

    $Etapas_100 = IncidenciaEstado::model()->findByAttributes(array('porcentaje' => 100));

    $model->unsetAttributes(); // clear any default values

    $elemento = "validar";

    if (isset($_GET['search'])) {

        $elemento = $_GET['element'];

        $model->attributes = $this->assignParams($_GET['search']);

    }

    // if ($model->activos()->findAll())

    {

        $this->render('admin', array('Etapas_100' => $Etapas_100, 'elemento' => $elemento,
        'model' => $model));

    }

    // else {

    //     $this->render('empty', array('model' => $model));

    // }

}

public function actionBandejaIncidencias() {

    $model = new Incidencia('search');

    $Etapas_100 = IncidenciaEstado::model()->findByAttributes(array('porcentaje' => 100));

    $model->unsetAttributes(); // clear any default values
```

```
$elemento = "validar";

if (isset($_GET['search'])) {

    $elemento = $_GET['element'];

    $model->attributes = $this->assignParams($_GET['search']);

}

$this->render('bandeja_incidencias', array('Etapa_100' => $Etapa_100, 'elemento' =>
$elemento, 'model' => $model));

}

/**

 * Returns the data model based on the primary key given in the GET variable.

 * If the data model is not found, an HTTP exception will be raised.

 * @param integer the ID of the model to be loaded

 */

public function loadModel($id, $modelClass = __CLASS__) {

    $model = Incidencia::model()->findByPrimaryKey($id);

    if ($model === null)

        throw new CHttpException(404, 'The requested page does not exist.');
```

```
    return $model;

}

/**

 * Performs the AJAX validation.

 * @param CModel the model to be validated

 */

protected function performAjaxValidation($model, $form = null) {
```

```
if (isset($_POST['ajax']) && $_POST['ajax'] === 'incidencia-form') {

    echo CActiveForm::validate($model);

    Yii::app()->end();

}

}

/**

* Usado para obtener los contacto que pertenecen a una cuenta.

* @autor Armando Maldonado <amaldonado@tradesystem.com.ec>

*/

public function actionCuentaContacto() {

    if ($_POST['Incidencia']['cuenta_id']) {

        $contacto = CHtml::listData(Contacto::model()->activos()-
>de_cuenta($_POST['Incidencia']['cuenta_id']->findAll(), 'id', 'nombre_completo');

        if (!empty($contacto)) {

            echo CHtml::tag('option', array('value' => ""), CHtml::encode('- Sin Contacto -'),
true);

            foreach ($contacto as $value => $name) {

                echo CHtml::tag('option', array('value' => $value), CHtml::encode($name), true);

            }

        }

    }

}

/**

* Realiza en render de la vista Kanban
```

```
* @author Armando Maldonado

*/

public function actionKanban() {

    $incidencia_estados = IncidenciaEstado::model()->activos()->orden()->findAll();

    $result = array();

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

        $result['success'] = true;

        $result['html'] = $this->renderPartial('_kanban', array('incidencia_estados' =>
$incidencia_estados), TRUE, TRUE);

        echo json_encode($result);

    } else {

        $this->render('kanban', array(

            'incidencia_estados' => $incidencia_estados,

        ));

    }

}

/**

* Realiza en render de la vista Kanban

* @author Armando Maldonado

*/

public function actionKanbanBandeja() {

    $incidencia_estados = IncidenciaEstado::model()->activos()->orden()->findAll();

    $incidencia_bandeja = true;

    //    var_dump($incidencia_estados);die();

    $result = array();
```

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

    $result['success'] = true;

    $result['html'] = $this->renderPartial('_kanban', array(

        'incidencia_estados' => $incidencia_estados,

        'incidencia_bandeja' => $incidencia_bandeja,

        ), TRUE, TRUE);

    echo json_encode($result);

} else {

    $this->render('kanban_bandeja', array(

        'incidencia_estados' => $incidencia_estados,

        'incidencia_bandeja' => $incidencia_bandeja,

    ));

}

}

/**

 * Asigna los parametro de busqueda para el widget search

 * @param {array} $params

 * @return {array}

 */

public function assignParams($params) {

    $result = array();

    if ($params['filters'][0] == 'all') {

        foreach (Incidencia::model()->searchParams() as $param) {

            $result[$param] = $params['value'];

        }

    }

}
```

* Acción usada en vista kanban, en el cual actualiza el estado de la incidencia.

* @author Armando Maldonado

* @param type \$id_data

* @param type \$id_estado

*/

```
public function actionAjaxUpdateEstado($id_data = null, $id_estado = null) {  
  
    $FechaHoraActualizacion = Util::FechaActual();  
  
    Incidencia::model()->updateByPk($id_data, array('incidencia_estado_id' => $id_estado,  
        'usuario_actualizacion_id' => Yii::app()->user->id,  
        'fecha_actualizacion' => $FechaHoraActualizacion));  
}
```

/**

*

* @param type \$id

* @param type \$fecha

* Accion para actualizar la fecha mediante ajax

*/

```
public function actionAjaxUpdateFecha($id, $fecha) {  
  
    $result = array();  
  
    if (Yii::app()->request->isAjaxRequest) {  
  
        $model = $this->loadModel($id);  
  
        $model->incidencia_categoria_id = $model->incidenciaSubmotivo->incidencia_categoria_id;  
  
        $model->incidencia_motivo_id = $model->incidenciaSubmotivo->incidencia_motivo_id;
```

```
$assignadoactual = $model->usuario_actualizacion_id;

$model->fecha_est_resolucion = $fecha;

$result = $this->Update($model, $assignadoactual);

if ($result['success']) {

} else {

    $result['success'] = false;

    $result['mensaje'] = "Error al actualizar la fecha de la incidencia";

}

echo json_encode($result);

}

}

public function actionAjaxCargarActividades($id) {

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

    $result = array();

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

        $result['success'] = true;

        $result['html'] = $this->renderPartial('portlets/_actividades', array('model' => $model,
'modal' => TRUE), TRUE, TRUE);

        echo json_encode($result);

    }

}

/**

* @author Alex Yepez <ayepez@tradesystem.com.ec>
```

```
* @param type $post

* @return array $b_Guardado

* Crea una nueva oportunidad

*/

private function Crear($model) {

    $b_Guardado = array();

    if ($model->fecha_est_resolucion) {

        $model->fecha_est_resolucion = Util::FormatDate($model->fecha_est_resolucion,
"Y/m/d");

    }

    $b_Guardado['success'] = $model->save();

    $model->nombre = $this->number_pad($model->id, 7);

    $model->save();

    $incidenciaGuardada = Incidencia::model()->findByPk($model->id);

    $usuario = Yii::app()->user->um->loadUserById(Constants::ID_MAIL_INCIDENCIAS, true);

    $porcentajeInci = IncidenciaEstado::model()->findByPk($incidenciaGuardada->incidencia_estado_id);

    // var_dump($incidenciaGuardada);

    // var_dump($porcentajeInci);

    // die();

    if (!$usuario == null && $porcentajeInci->porcentaje < 100) {

        //TODO: cambiar correos por $correos=$Usuario->email;

        $correo = $usuario->email;
```

```
$body = utf8_decode($this->renderPartial('_mail', array('incidenciaGuardada' =>
$incidenciaGuardada, 'usuario' => $usuario), TRUE, TRUE));

//      Mailer::enviarEmail($correo, 'Incidencia #' . $incidenciaGuardada->nombre, $body);

    }

    if ($b_Guardado['success']) {

        Actividad::registrarActividad($model, Actividad::TIPO_CREATE);

        $signadonuevo = $model->owner_id;

        if ($signadonuevo != Yii::app()->user->id) {

            Alerta::registrarAlertaA($model, Alerta::ASIGNADO);

        }

    }

    return $b_Guardado;

}

/**
 *
 * @param type $model
 * @return type
 */

private function Update($model, $signadoactual) {

    $b_Guardado = array();

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

```
$model->fecha_est_resolucion = Util::FormatDate($model->fecha_est_resolucion,
"Y/m/d");

}

$model->usuario_actualizacion_id = Yii::app()->user->id;

$b_Guardado['success'] = $model->save();

if ($b_Guardado['success']) {

    $asignadonuevo = $model->owner_id;

    // Crear registro de actividad

    Actividad::registrarActividad($model, Actividad::TIPO_UPDATE);

    if ($asignadoactual != $asignadonuevo) {

        if ($asignadonuevo == Yii::app()->user->id) {

            Alerta::registrarAlertaNA($model, $asignadoactual, Alerta::NOASIGNADO);

        } if ($asignadoactual == Yii::app()->user->id) {

            Alerta::registrarAlertaA($model, Alerta::ASIGNADO);

        }

        if ($asignadonuevo != Yii::app()->user->id && $asignadoactual != Yii::app()-
>user->id) {

            Alerta::registrarAlertaNA($model, $asignadoactual, Alerta::NOASIGNADO);

            Alerta::registrarAlertaA($model, Alerta::ASIGNADO);

        }

    }

}

return $b_Guardado;

}
```

```
public function number_pad($number, $n) {  
  
    return str_pad((int) $number, $n, "0", STR_PAD_LEFT);  
  
}  
  
}
```

Llamadas

<?php

```
//Yii::import('llamadas.models._base.BaseReporte');  
  
//class LlamadaReporte extends BaseReporte {  
  
class LlamadaReporte extends CFormModel {  
  
  
  
  
  
  
  
  
  
    //Atributos  
  
    public $fechas;  
  
  
  
  
  
  
  
  
  
    // public static $tiposLlamadas = array(  
  
    //     'Recibida','Atendida','Abandonada'  
  
    // );  
  
  
  
  
    public static function model($className = __CLASS__) {  
  
        return parent::model($className);  
  
    }
```

```
public function rules() {

    return array(

        array('fechas', 'required'), //pendiente

    );

}

/**

 * @return array customized attribute labels (name=>label)

 */

public function attributeLabels() {

    return array(

        'fechas' => Yii::t('app', 'Filtro'),

    );

}

public function generatePieReport($inicio, $fin, $motivo_id) {

    $inicio = DateTime::createFromFormat('d/m/Y', $inicio);

    $fin = DateTime::createFromFormat('d/m/Y', $fin)->modify('+1 day');

    $report = array();

    $report['title']['text'] = $motivo_id? 'Submotivos':'Motivos';

    $report['credits']['enabled'] = false;

    $report['chart']['height'] = '280';

    $report['plotOptions'] = array(

        'pie' => array(

            'allowPointSelect' => true,

            'cursor' => 'pointer',
```

```
'dataLabels' => array(

    'enabled' => false,

    'format' => '<b>{point.name}</b>: {point.percentage:.1f} %'

),

'showInLegend' => true,

'tooltip' => array(

    'pointFormat' => '{series.name}: <b>{point.percentage:.1f}%</b>'

),

);

$report['series'] = array();

$consulta = Yii::app()->db->createCommand()

    ->select('$motivo_id ? 'mot.id, sub.nombre, count(lla.id) as count' : 'mot.id,
mot.nombre, count(lla.id) as count')

    ->from('llamada lla')

    ->join('llamada_submotivo sub', 'sub.id = lla.submotivo_id')

    ->join('llamada_motivo mot', 'mot.id = sub.motivo_id');

if ($motivo_id) {

    $consulta->where('mot.estado=:estado and lla.tipo=:tipo and mot.id=:motivo_id and
lla.fecha_creacion between :inicio and :fin', array(':estado' => 'ACTIVO', ':tipo' =>
'ENTRANTE', ':motivo_id' => $motivo_id, ':inicio' => $inicio->format('Y-m-d'), ':fin' =>
$fin->format('Y-m-d')))

    ->group('sub.id')

    ->order('sub.nombre');

} else {
```

```
$consulta->where('mot.estado=:estado and lla.tipo=:tipo and lla.fecha_creacion
between :inicio and :fin', array(':estado' => 'ACTIVO', ':tipo' => 'ENTRANTE', ':inicio' =>
$inicio->format('Y-m-d'), ':fin' => $fin->format('Y-m-d')))
```

```
->group('mot.id')
```

```
->order('mot.nombre')
```

```
;
```

```
}
```

```
$options = $consulta->queryAll();
```

```
$series = array(
```

```
    'type' => 'pie',
```

```
    'name' => 'Cantidad',
```

```
    'data' => array()
```

```
);
```

```
foreach ($options as $option) {
```

```
    $series['data'][] = array(Util::truncate($option['nombre'], 21), (int) $option['count'],
$option['id']);
```

```
}
```

```
$report['series'][] = $series;
```

```
return $report;
```

```
}
```

```
public function generateColumnReport($inicio, $fin, $motivo_id) {
```

```
    $inicio = DateTime::createFromFormat('d/m/Y', $inicio)->setTime(0, 0, 0);
```

```
    $fin = DateTime::createFromFormat('d/m/Y', $fin)->setTime(0, 0, 0);
```

```
    $interval = DateInterval::createFromDateString('1 day');
```

```
    $period = new DatePeriod($inicio, $interval, $fin); //frecuencia del intervalo a calcular
```

```
$dias = 0;

$report = array();

$report['title']['text'] = 'Llamadas';

$report['credits']['enabled'] = false;

$report['chart']['height'] = '320';

$report['subtitle']['text'] = ' Desde: ' . $inicio->format('d-m-Y') . ' Hasta: ' . $fin-
>format('d-m-Y');

$report['xAxis']['labels'] = array("rotation" => -45);

$report['yAxis']['min'] = 0;

$report['yAxis']['title']['text'] = "Número de Llamadas";

$report['yAxis']['allowDecimals'] = false;

$report['xAxis']['categories'] = array();

$report['series'] = array();

if ($motivo_id) {

    $motivos = LlamadaSubmotivo::model()->activos()->findAll('motivo_id =
:motivo_id', array(':motivo_id' => $motivo_id));

} else {

    $motivos = LlamadaMotivo::model()->entrantes()->activos()->findAll();

}

foreach ($period as $date) {

    $dias++;

}

switch (true) {

//horas
```

```
case ($dias == 0):

    $interval = DateInterval::createFromDateString('1 hour');

    $fin->add(new DateInterval('PT23H59M59S'));

    $period = new DatePeriod($inicio, $interval, $fin); //frecuencia del intervalo a
calcular

    foreach ($motivos as $motivo) {

        $data = array();

        foreach ($period as $date) {

            $inicio = $date->format('Y-m-d H:i:s');

            $categoria = $date->format('H:i');

            $date->add(new DateInterval('PT59M59S'));

            $fin = $date->format('Y-m-d H:i:s');

            $report['xAxis']['categories'][] = $categoria;

            array_push($data, $this->consulta($motivo, $inicio, $fin, $motivo_id));

        }

        array_push($report['series'], array('name' => Util::Truncate($motivo->nombre,
21), 'data' => $data, 'type' => 'column')

    );

    }

    break;

//dias

case ($dias >= 1 && $dias <= 30):

    $fin->add(new DateInterval('PT23H59M59S'));

    $interval = DateInterval::createFromDateString('1 day');
```

\$period = new DatePeriod(\$inicio, \$interval, \$fin); //frecuencia del intervalo a calcular

```
foreach ($motivos as $motivo) {

    $data = array();

    foreach ($period as $date) {

        $inicio = $date->format('Y-m-d H:i:s');

        $categoria = $date->format('d-M');

        $date->add(new DateInterval('PT23H59M59S'));

        $fin = $date->format('Y-m-d H:i:s');

        $report['xAxis']['categories'][] = $categoria;

        array_push($data, $this->consulta($motivo, $inicio, $fin, $motivo_id));

    }

    array_push($report['series'], array('name' => Util::Truncate($motivo->nombre,
2121), 'data' => $data, 'type' => 'column')

    );

}

break;

//semanas

case ($dias >= 31 && $dias <= 83):

    if ($inicio->format('D') != 'Mon') {

        $inicio->modify('last monday');

    }

    if ($fin->format('D') == 'Mon') {

        $fin->add(new DateInterval('P6DT23H59M59S'));

    }

}
```

```
$interval = DateInterval::createFromDateString('1 week');

$period = new DatePeriod($inicio, $interval, $fin); //frecuencia del intervalo a
calcular

foreach ($motivos as $motivo) {

    $data = array();

    foreach ($period as $date) {

        $inicio = $date->format('Y-m-d H:i:s');

        $categoria = $date->format('d/M');

        $date->add(new DateInterval('P6DT23H59M59S'));

        $fin = $date->format('Y-m-d H:i:s');

        $categoria.=' - ' . $date->format('d/M');

        $report['xAxis']['categories'][] = $categoria;

        array_push($data, $this->consulta($motivo, $inicio, $fin, $motivo_id));

    }

    array_push($report['series'], array('name' => Util::Truncate($motivo->nombre,
21), 'data' => $data, 'type' => 'column')

    );

}

break;

//meses

case ($dias >= 84):

    $interval = DateInterval::createFromDateString('1 month');

    $period = new DatePeriod($inicio, $interval, $fin); //frecuencia del intervalo a
calcular

    foreach ($motivos as $motivo) {

        $data = array();
```

```
foreach ($period as $date) {

    $date->modify('first day of this month');

    $inicio = $date->format('Y-m-d H:i:s');

    $categoria = $date->format('F');

    $date->modify('last day of this month');

    $date->add(new DateInterval('PT23H59M59S'));

    $fin = $date->format('Y-m-d H:i:s');

    $report['xAxis']['categories'][] = $categoria;

    array_push($data, $this->consulta($motivo, $inicio, $fin, $motivo_id));

}

array_push($report['series'], array('name' => Util::Truncate($motivo->nombre,
21), 'data' => $data, 'type' => 'column')

);

}

break;

}

return $report;

}

private function consulta($motivo, $inicio, $fin, $motivo_id) {

    $data = array();

    $consulta = Yii::app()->db->createCommand()

        ->select($motivo_id ? 'sub.nombre, count(la.id) as count' : 'mot.nombre,
count(la.id) as count')
```

```
->from('llamada lla')

->join('llamada_submotivo sub', 'sub.id = lla.submotivo_id')

->join('llamada_motivo mot', 'mot.id = sub.motivo_id');

if ($motivo_id) {

    $consulta->where('sub.estado=:estado and lla.submotivo_id = :motivo and
lla.tipo=:tipo and lla.fecha_creacion between :inicio and :fin', array(':estado' => 'ACTIVO',
':motivo' => $motivo->id, ':tipo' => 'ENTRANTE', ':inicio' => $inicio, ':fin' => $fin))

    ->group('sub.id')

    ->order('mot.nombre');

} else {

    $consulta->where('mot.estado=:estado and mot.id = :motivo and lla.tipo=:tipo and
lla.fecha_creacion between :inicio and :fin', array(':estado' => 'ACTIVO', ':motivo' =>
$motivo->id, ':tipo' => 'ENTRANTE', ':inicio' => $inicio, ':fin' => $fin))

    ->group('mot.id')

    ->order('mot.nombre');

}

$options = $consulta->queryAll();

if (!empty($options)) {

    $data[] = (int) $options[0]['count'];

} else {

    $data[] = (int) '0';

}

return $data;

}
```

```
public function mayorMenorMotivo($inicio, $fin, $tipo, $motivo_id) {

    if ($tipo) {

        $tipo = 'desc';

    } else {

        $tipo = 'asc';

    }

    $inicio = DateTime::createFromFormat('d/m/Y', $inicio);

    $fin = DateTime::createFromFormat('d/m/Y', $fin)->modify('+1 day');

    $consulta = Yii::app()->db->createCommand()

        ->select('$motivo_id ? 'sub.nombre, count(lla.id) as count' : 'mot.nombre,
count(lla.id) as count')

        ->from('llamada lla')

        ->join('llamada_submotivo sub', 'sub.id = lla.submotivo_id')

        ->join('llamada_motivo mot', 'mot.id = sub.motivo_id');

    if ($motivo_id) {

        $consulta->where('mot.estado=:estado and lla.tipo=:tipo and mot.id=:motivo_id and
lla.fecha_creacion between :inicio and :fin', array(':estado' => 'ACTIVO', ':tipo' =>
'ENTRANTE', ':motivo_id' => $motivo_id, ':inicio' => $inicio->format('Y-m-d'), ':fin' =>
$fin->format('Y-m-d')))

        ->group('sub.id');

    } else {

        $consulta->where('mot.estado=:estado and lla.tipo=:tipo and lla.fecha_creacion
between :inicio and :fin', array(':estado' => 'ACTIVO', ':tipo' => 'ENTRANTE', ':inicio' =>
$inicio->format('Y-m-d'), ':fin' => $fin->format('Y-m-d')))

        ->group('mot.id');

    }

}
```

```
$consulta->order('count ' . $tipo)

->limit(1);

$options = $consulta->queryAll();

$consulta = Yii::app()->db->createCommand()

->select('count(lla.id) as total')

->from('llamada lla')

->join('llamada_submotivo sub', 'sub.id = lla.submotivo_id')

->join('llamada_motivo mot', 'mot.id = sub.motivo_id');

if ($motivo_id) {

    $consulta->where('mot.estado=:estado and lla.tipo=:tipo and mot.id=:motivo_id and
    lla.fecha_creacion between :inicio and :fin', array(':estado' => 'ACTIVO', ':tipo' =>
    'ENTRANTE', ':motivo_id' => $motivo_id, ':inicio' => $inicio->format('Y-m-d'), ':fin' =>
    $fin->format('Y-m-d')));

    } else {

        $consulta->where('mot.estado=:estado and lla.tipo=:tipo and lla.fecha_creacion
        between :inicio and :fin', array(':estado' => 'ACTIVO', ':tipo' => 'ENTRANTE', ':inicio' =>
        $inicio->format('Y-m-d'), ':fin' => $fin->format('Y-m-d')));

    }

$total = $consulta->queryAll();

if (!empty($options)) {

    $options['0']['count'] = round(($options['0']['count'] * 100) / $total['0']['total']);

    $options['0']['total'] = $total['0']['total'];

    } else {

        $options['0'] = array();

    }
```

```
}  
  
return $options['0'];  
  
}
```

```
}
```

Prioridad de incidencia

```
class IncidenciaPrioridadController extends AweController {
```

```
/**
```

```
 * @var string the default layout for the views. Defaults to '//layouts/column2', meaning  
 * using two-column layout. See 'protected/views/layouts/column2.php'.
```

```
*/
```

```
public $layout = '//layouts/column2';
```

```
public $defaultAction = 'admin';
```

```
public $admin = true;
```

```
public function filters() {
```

```
    return array(  
        array('CrugeAccessControlFilter'),  
    );  
}
```

```
/**
```

* Displays a particular model.

* @param integer \$id the ID of the model to be displayed

*/

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

```
    $this->render('view', array(
```

```
        'model' => $this->loadModel($id),
```

```
    ));
```

```
}
```

```
/**
```

* Creates a new model.

* If creation is successful, the browser will be redirected to the 'view' page.

*/

```
public function actionCreate() {
```

```
    $model = new IncidenciaPrioridad;
```

```
    $this->performAjaxValidation($model, 'incidencia-prioridad-form');
```

```
    if (isset($_POST['IncidenciaPrioridad'])) {
```

```
        $model->attributes = $_POST['IncidenciaPrioridad'];
```

```
        $model->estado = IncidenciaPrioridad::ESTADO_ACTIVO;
```

```
        $model->peso = IncidenciaPrioridad::model()->getPesoMaximo() + 1;
```

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

```
            $this->redirect(array('admin'));
```

```
        }
```

```
}
```

```
$this->render('create', array(

    'model' => $model,

));

}

/**

* Updates a particular model.

* If update is successful, the browser will be redirected to the 'view' page.

* @param integer $id the ID of the model to be updated

*/

public function actionUpdate($id) {

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

    $this->performAjaxValidation($model, 'incidencia-prioridad-form');

    if (isset($_POST['IncidenciaPrioridad'])) {

        $model->attributes = $_POST['IncidenciaPrioridad'];

        if ($model->save()) {

            $this->redirect(array('admin'));

        }

    }

    $this->render('update', array(

        'model' => $model,

    ));
```

```
}

/**

* Deletes a particular model.

* If deletion is successful, the browser will be redirected to the 'admin' page.

* @param integer $id the ID of the model to be deleted

*/

public function actionDelete($id) {

    if (Yii::app()->request->isPostRequest) {

        // we only allow deletion via POST request

        IncidenciaPrioridad::model()->updateByPk($id,          array('estado'          =>
IncidenciaPrioridad::ESTADO_INACTIVO));

        // if AJAX request (triggered by deletion via admin grid view), we should not redirect
the browser

        if (!isset($_GET['ajax']))

            $this->redirect(isset($_POST['returnUrl']) ? $_POST['returnUrl'] : array('admin'));

        } else

            throw new CHttpException(400, 'Invalid request. Please do not repeat this request
again.');
```

```
}

/**

* Lists all models.

*/

public function actionIndex() {
```

```
$dataProvider = new CActiveDataProvider('IncidenciaPrioridad');

$this->render('index', array(

    'dataProvider' => $dataProvider,

));

}

/**

 * Manages all models.

 */

public function actionAdmin() {

    $model = new IncidenciaPrioridad('search');

    $model->unsetAttributes(); // clear any default values

    if (isset($_GET['IncidenciaPrioridad']))

        $model->attributes = $_GET['IncidenciaPrioridad'];

    $this->render('admin', array(

        'model' => $model,

    ));

}

/**

 * Returns the data model based on the primary key given in the GET variable.

 * If the data model is not found, an HTTP exception will be raised.

 * @param integer the ID of the model to be loaded

 */

public function loadModel($id, $modelClass = __CLASS__) {
```

```
$model = IncidenciaPrioridad::model()->findByPk($id);

if ($model === null)

    throw new CHttpException(404, 'The requested page does not exist.');
```

```
return $model;

}
```

```
/**

 * Performs the AJAX validation.

 * @param CModel the model to be validated

 */

protected function performAjaxValidation($model, $form = null) {

    if (isset($_POST['ajax']) && $_POST['ajax'] === 'incidencia-prioridad-form') {

        echo CActiveForm::validate($model);

        Yii::app()->end();

    }

}
```

```
}
```

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';

-- Estructura de tabla para la tabla `actividad`

CREATE TABLE IF NOT EXISTS `actividad` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `entidad_tipo` varchar(64) NOT NULL,
  `entidad_id` int(11) NOT NULL,
  `tipo` enum('CREATE','UPDATE','DELETE','RESTORE') NOT NULL,
  `usuario_id` int(11) NOT NULL,
  `fecha` datetime NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=201 ;

--

-- Estructura de tabla para la tabla `alerta`

--

CREATE TABLE IF NOT EXISTS `alerta` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `tipo_codigo` int(11) NOT NULL,
  `entidad_tipo` varchar(64) NOT NULL,
  `entidad_id` int(11) NOT NULL,
  `leido` tinyint(1) NOT NULL,
  `visitado` tinyint(1) NOT NULL,
  `usuario_id` int(11) NOT NULL,
  `fecha_creacion` datetime NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=6 ;
```

-- Estructura de tabla para la tabla `contacto`

--

```
CREATE TABLE IF NOT EXISTS `contacto` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `nombre` varchar(32) DEFAULT NULL,  
  `apellido` varchar(32) NOT NULL,  
  `titulo` varchar(10) DEFAULT NULL,  
  `documento` varchar(20) DEFAULT NULL,  
  `cargo` varchar(64) DEFAULT NULL,  
  `departamento` varchar(64) DEFAULT NULL,  
  `telefono` varchar(24) DEFAULT NULL,  
  `celular` varchar(24) DEFAULT NULL,  
  `fax` varchar(24) DEFAULT NULL,  
  `email_1` varchar(255) DEFAULT NULL,  
  `email_2` varchar(255) DEFAULT NULL,  
  `descripcion` text,  
  `rating` int(11) DEFAULT '0',  
  `estado` enum('ACTIVO','INACTIVO') NOT NULL DEFAULT 'ACTIVO',  
  `usuario_creacion_id` int(11) NOT NULL,  
  `usuario_actualizacion_id` int(11) DEFAULT NULL,  
  `fecha_creacion` datetime NOT NULL,  
  `fecha_actualizacion` datetime DEFAULT NULL,  
  `direccion_principal_id` int(11) DEFAULT NULL,  
  `direccion_secundaria_id` int(11) DEFAULT NULL,
```

```
`cuenta_id` int(11) DEFAULT NULL,

`permisos` enum('OWNER','ALL') NOT NULL,

`owner_id` int(11) NOT NULL,

PRIMARY KEY (`id`),

KEY `fk_contacto_direccion1_idx` (`direccion_principal_id`),

KEY `fk_contacto_direccion2_idx` (`direccion_secundaria_id`),

KEY `fk_contacto_cuenta1_idx` (`cuenta_id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

--

-- Estructura de tabla para la tabla `cruge_authassignment`

--

CREATE TABLE IF NOT EXISTS `cruge_authassignment` (

  `userid` int(11) NOT NULL,

  `bizrule` text,

  `data` text,

  `itemname` varchar(64) NOT NULL,

  PRIMARY KEY (`userid`,`itemname`),

  KEY `fk_crug_authassignment_crug_authitem1` (`itemname`),

  KEY `fk_crug_authassignment_user` (`userid`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Estructura de tabla para la tabla `crug_authitem`

--

CREATE TABLE IF NOT EXISTS `crug_authitem` (

  `name` varchar(64) NOT NULL,
```



```
`type` int(11) NOT NULL,  
  
`description` text,  
  
`bizrule` text,  
  
`data` text,  
  
PRIMARY KEY (`name`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
--  
  
-- Estructura de tabla para la tabla `cruge_authitemchild`  
  
--  
  
CREATE TABLE IF NOT EXISTS `cruge_authitemchild` (  
  
  `parent` varchar(64) NOT NULL,  
  
  `child` varchar(64) NOT NULL,  
  
  PRIMARY KEY (`parent`,`child`),  
  
  KEY `child` (`child`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
--  
  
-- Estructura de tabla para la tabla `cruge_field`  
  
--  
  
CREATE TABLE IF NOT EXISTS `cruge_field` (  
  
  `idfield` int(11) NOT NULL AUTO_INCREMENT,  
  
  `fieldname` varchar(20) NOT NULL,  
  
  `longname` varchar(50) DEFAULT NULL,  
  
  `position` int(11) DEFAULT '0',  
  
  `required` int(11) DEFAULT '0',  
  
  `fieldtype` int(11) DEFAULT '0',
```

```
`fieldsize` int(11) DEFAULT '20',
`maxlength` int(11) DEFAULT '45',
`showinreports` int(11) DEFAULT '0',
`useregexp` varchar(512) DEFAULT NULL,
`useregexpmsg` varchar(512) DEFAULT NULL,
`predetvalue` mediumblob,
PRIMARY KEY (`idfield`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

--

-- Estructura de tabla para la tabla `cruge_fieldvalue`
--

CREATE TABLE IF NOT EXISTS `cruge_fieldvalue` (
  `idfieldvalue` int(11) NOT NULL AUTO_INCREMENT,
  `iduser` int(11) NOT NULL,
  `idfield` int(11) NOT NULL,
  `value` blob,
  PRIMARY KEY (`idfieldvalue`),
  KEY `fk_crug_fieldvalue_crug_user1` (`iduser`),
  KEY `fk_crug_fieldvalue_crug_field1` (`idfield`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

--

-- Estructura de tabla para la tabla `cruge_session`
--

CREATE TABLE IF NOT EXISTS `cruge_session` (
  `idsession` int(11) NOT NULL AUTO_INCREMENT,
```

```
`iduser` int(11) NOT NULL,

`created` bigint(30) DEFAULT NULL,

`expire` bigint(30) DEFAULT NULL,

`status` int(11) DEFAULT '0',

`ipaddress` varchar(45) DEFAULT NULL,

`usagecount` int(11) DEFAULT '0',

`lastusage` bigint(30) DEFAULT NULL,

`logoutdate` bigint(30) DEFAULT NULL,

`ipaddressout` varchar(45) DEFAULT NULL,

PRIMARY KEY (`idsession`),

KEY `crugesession_iduser` (`iduser`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=25 ;

--

-- Estructura de tabla para la tabla `cruge_system`

--

CREATE TABLE IF NOT EXISTS `cruge_system` (

  `idsystem` int(11) NOT NULL AUTO_INCREMENT,

  `name` varchar(45) DEFAULT NULL,

  `largename` varchar(45) DEFAULT NULL,

  `sessionmaxdurationmins` int(11) DEFAULT '30',

  `sessionmaxsameipconnections` int(11) DEFAULT '10',

  `sessionreusesessions` int(11) DEFAULT '1' COMMENT '1yes 0no',

  `sessionmaxsessionsperday` int(11) DEFAULT '-1',

  `sessionmaxsessionsperuser` int(11) DEFAULT '-1',

  `systemnonewsessions` int(11) DEFAULT '0' COMMENT '1yes 0no',
```

```
`systemdown` int(11) DEFAULT '0',
`registerusingcaptcha` int(11) DEFAULT '0',
`registerusingterms` int(11) DEFAULT '0',
`terms` blob,
`registerusingactivation` int(11) DEFAULT '1',
`defaultroleforregistration` varchar(64) DEFAULT NULL,
`registerusingtermslabel` varchar(100) DEFAULT NULL,
`registrationonlogin` int(11) DEFAULT '1',
PRIMARY KEY (`idsystem`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;
-- Estructura de tabla para la tabla `cruge_user`
--
CREATE TABLE IF NOT EXISTS `cruge_user` (
  `iduser` int(11) NOT NULL AUTO_INCREMENT,
  `regdate` bigint(30) DEFAULT NULL,
  `actdate` bigint(30) DEFAULT NULL,
  `logondate` bigint(30) DEFAULT NULL,
  `username` varchar(64) DEFAULT NULL,
  `email` varchar(45) DEFAULT NULL,
  `password` varchar(64) DEFAULT NULL COMMENT 'Hashed password',
  `authkey` varchar(100) DEFAULT NULL COMMENT 'llave de autentificacion',
  `state` int(11) DEFAULT '0',
  `totalsessioncounter` int(11) DEFAULT '0',
  `currentsessioncounter` int(11) DEFAULT '0',
  PRIMARY KEY (`iduser`)
```

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;

-- Estructura de tabla para la tabla `cuenta`

--

```
CREATE TABLE IF NOT EXISTS `cuenta` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `nombre` varchar(64) NOT NULL,  
  `razon_social` varchar(64) DEFAULT NULL,  
  `documento` varchar(20) DEFAULT NULL,  
  `celular` varchar(24) DEFAULT NULL,  
  `email` varchar(255) DEFAULT NULL,  
  `telefono` varchar(24) DEFAULT NULL,  
  `fax` varchar(24) DEFAULT NULL,  
  `website` varchar(255) DEFAULT NULL,  
  `tipo` enum('PROSPECTO','CLIENTE') DEFAULT NULL,  
  `empleados` int(11) DEFAULT NULL,  
  `ingresos_anuales` double DEFAULT NULL,  
  `descripcion` text,  
  `ultima_actividad` datetime DEFAULT NULL,  
  `rating` int(11) DEFAULT '0',  
  `estado` enum('ACTIVO','INACTIVO') NOT NULL DEFAULT 'ACTIVO',  
  `usuario_creacion_id` int(11) NOT NULL,  
  `usuario_actualizacion_id` int(11) DEFAULT NULL,  
  `fecha_creacion` datetime NOT NULL,  
  `fecha_actualizacion` datetime DEFAULT NULL,
```

```
`direccion_facturacion_id` int(11) DEFAULT NULL,

`direccion_entrega_id` int(11) DEFAULT NULL,

`industria_id` int(11) DEFAULT NULL,

`permisos` enum('OWNER','ALL') NOT NULL,

`owner_id` int(11) NOT NULL,

PRIMARY KEY (`id`),

KEY `fk_cuenta_direccion_idx` (`direccion_facturacion_id`),

KEY `fk_cuenta_direccion1_idx` (`direccion_entrega_id`),

KEY `fk_cuenta_industria1_idx` (`industria_id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;

--

-- Estructura de tabla para la tabla `direccion`

--

CREATE TABLE IF NOT EXISTS `direccion` (

  `id` int(11) NOT NULL AUTO_INCREMENT,

  `calle_1` varchar(128) DEFAULT NULL,

  `calle_2` varchar(128) DEFAULT NULL,

  `ciudad` varchar(32) DEFAULT NULL,

  `estado` varchar(32) DEFAULT NULL,

  `codigo_postal` varchar(16) DEFAULT NULL,

  `pais` varchar(32) DEFAULT NULL,

  PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

--
```

-- Estructura de tabla para la tabla `formulario`

--

```
CREATE TABLE IF NOT EXISTS `formulario` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `nivel1_id` int(11) NOT NULL,  
  `nivel3_id` int(11) DEFAULT NULL,  
  `archivo` varchar(80) NOT NULL,  
  `descripcion` text,  
  `fecha_creacion` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `id_UNIQUE` (`id`),  
  KEY `fk_formulario_nivel1_idx` (`nivel1_id`),  
  KEY `fk_formulario_nivel3_idx` (`nivel3_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

--

-- Estructura de tabla para la tabla `grupo`

--

```
CREATE TABLE IF NOT EXISTS `grupo` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `nombre` varchar(64) NOT NULL,  
  `descripcion` text,  
  `peso` int(11) DEFAULT NULL,  
  `estado` enum('ACTIVO','INACTIVO') NOT NULL DEFAULT 'ACTIVO',  
  PRIMARY KEY (`id`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

```
--
```

```
-- Estructura de tabla para la tabla `grupo_cuenta`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `grupo_cuenta` (
```

```
  `grupo_id` int(11) NOT NULL,
```

```
  `cuenta_id` int(11) NOT NULL,
```

```
  PRIMARY KEY (`grupo_id`,`cuenta_id`),
```

```
  KEY `fk_grupo_has_cuenta_cuenta1_idx` (`cuenta_id`),
```

```
  KEY `fk_grupo_has_cuenta_grupo1_idx` (`grupo_id`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--
```

```
-- Estructura de tabla para la tabla `incidencia`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `incidencia` (
```

```
  `id` int(11) NOT NULL AUTO_INCREMENT,
```

```
  `nombre` varchar(64) DEFAULT NULL,
```

```
  `descripcion` text,
```

```
  `estado` enum('ACTIVO','INACTIVO') NOT NULL,
```

```
  `escalamiento` tinyint(1) DEFAULT '0',
```

```
  `usuario_creacion_id` int(11) NOT NULL,
```

```
  `usuario_actualizacion_id` int(11) DEFAULT NULL,
```

```
  `fecha_est_resolucion` date DEFAULT NULL,
```

```
  `hora_est_resolucion` time DEFAULT NULL,
```

```
  `fecha_creacion` datetime NOT NULL,
```

```
`fecha_actualizacion` datetime 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) DEFAULT NULL,

`incidencia_prioridad_id` int(11) DEFAULT NULL,

`cuenta_id` int(11) DEFAULT NULL,

`contacto_id` int(11) DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `fk_incidencia_incidencia_estado1_idx` (`incidencia_estado_id`),

KEY `fk_incidencia_incidencia_via_ingreso1_idx` (`incidencia_via_ingreso_id`),

KEY `fk_incidencia_incidencia_prioridad1_idx` (`incidencia_prioridad_id`),

KEY `fk_incidencia_incidencia_submotivo1_idx` (`incidencia_submotivo_id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=6 ;

-- Estructura de tabla para la tabla `incidencia_categoria`
--

CREATE TABLE IF NOT EXISTS `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 CHARSET=latin1 AUTO_INCREMENT=2 ;

-- Estructura de tabla para la tabla `incidencia_estado`
```

```
CREATE TABLE IF NOT EXISTS `incidencia_estado` (  
    `id` int(11) NOT NULL AUTO_INCREMENT,  
    `nombre` varchar(64) NOT NULL,  
    `porcentaje` int(11) NOT NULL,  
    `peso` int(11) DEFAULT NULL,  
    `estado` enum('ACTIVO','INACTIVO') NOT NULL,  
    PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;  
  
--  
-- Estructura de tabla para la tabla `incidencia_motivo`  
--  
CREATE TABLE IF NOT EXISTS `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`),  
    KEY `fk_incidencia_motivo_incidencia_categoria1_idx`  
    (`incidencia_categoria_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;  
  
-- Estructura de tabla para la tabla `incidencia_prioridad`  
--  
CREATE TABLE IF NOT EXISTS `incidencia_prioridad` (  
    `id` int(11) NOT NULL AUTO_INCREMENT,  
    `nombre` varchar(64) NOT NULL,
```

```
`color` varchar(10) NOT NULL,

`peso` int(11) DEFAULT NULL,

`estado` enum('ACTIVO','INACTIVO') NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;

-- Estructura de tabla para la tabla `incidencia_submotivo`

--

CREATE TABLE IF NOT EXISTS `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`),

  KEY `fk_incidencia_submotivo_incidencia_motivo1_idx` (`incidencia_motivo_id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=7 ;

-- Estructura de tabla para la tabla `incidencia_via_ingreso`

--

CREATE TABLE IF NOT EXISTS `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 CHARSET=latin1 AUTO_INCREMENT=2 ;

-- Estructura de tabla para la tabla `industria`
```

--

```
CREATE TABLE IF NOT EXISTS `industria` (  
    `id` int(11) NOT NULL AUTO_INCREMENT,  
    `nombre` varchar(64) NOT NULL,  
    PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

--

-- Estructura de tabla para la tabla `llamada`

--

```
CREATE TABLE IF NOT EXISTS `llamada` (  
    `id` int(11) NOT NULL AUTO_INCREMENT,  
    `tipo` enum('ENTRANTE','SALIENTE') NOT NULL,  
    `estado_llamada` enum('CONTACTADO','NO_CONTACTADO') NOT NULL,  
    `observaciones` text,  
    `fecha_creacion` datetime NOT NULL,  
    `usuario_creacion_id` int(11) NOT NULL,  
    `contacto_id` int(11) DEFAULT NULL,  
    `subestado_id` int(11) DEFAULT NULL,  
    `submotivo_id` int(11) NOT NULL,  
    PRIMARY KEY (`id`),  
    KEY `fk_llamada_llamada_subestado1_idx` (`subestado_id`),  
    KEY `fk_llamada_llamada_submotivo1_idx` (`submotivo_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=11 ;
```

-- Estructura de tabla para la tabla `llamada\_motivo`

--

```
CREATE TABLE IF NOT EXISTS `llamada_motivo` (  
    `id` int(11) NOT NULL AUTO_INCREMENT,  
    `nombre` varchar(64) NOT NULL,  
    `descripcion` text,  
    `estado` enum('ACTIVO','INACTIVO') NOT NULL,  
    `tipo` enum('SALIENTE','ENTRANTE') NOT NULL,  
    PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;  
  
-- Estructura de tabla para la tabla `llamada_subestado`  
--  
  
CREATE TABLE IF NOT EXISTS `llamada_subestado` (  
    `id` int(11) NOT NULL AUTO_INCREMENT,  
    `nombre` varchar(64) NOT NULL,  
    `descripcion` text,  
    `estado_llamada` enum('CONTACTADO','NO_CONTACTADO') NOT NULL,  
    `estado_gestion` enum('EFECTIVA','NO_EFECTIVA','RELLAMAR') NOT  
NULL,  
    `estado` enum('ACTIVO','INACTIVO') NOT NULL,  
    PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;  
  
--  
  
-- Estructura de tabla para la tabla `llamada_submotivo`  
--  
  
CREATE TABLE IF NOT EXISTS `llamada_submotivo` (  

```

```
`id` int(11) NOT NULL AUTO_INCREMENT,

`nombre` varchar(64) NOT NULL,

`descripcion` text,

`estado` enum('ACTIVO','INACTIVO') NOT NULL,

`motivo_id` int(11) NOT NULL,

PRIMARY KEY (`id`),

KEY `fk_llamada_submotivo_llamada_motivo1_idx` (`motivo_id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=7 ;

--

-- Filtró para la tabla `incidencia_motivo`

--

ALTER TABLE `incidencia_motivo`

ADD CONSTRAINT `fk_incidencia_motivo_incidencia_categoria1` FOREIGN

KEY (`incidencia_categoria_id`) REFERENCES `incidencia_categoria` (`id`) ON

DELETE NO ACTION ON UPDATE NO ACTION;

--

-- Filtró para la tabla `incidencia_submotivo`

--

ALTER TABLE `incidencia_submotivo`

ADD CONSTRAINT `fk_incidencia_submotivo_incidencia_motivo1` FOREIGN

KEY (`incidencia_motivo_id`) REFERENCES `incidencia_motivo` (`id`) ON

DELETE NO ACTION ON UPDATE NO ACTION;

--

-- Filtró para la tabla `llamada`

--
```

```
ALTER TABLE `llamada`
```

```
    ADD CONSTRAINT `fk_llamada_llamada_subestado1` FOREIGN KEY  
    (`subestado_id`) REFERENCES `llamada_subestado` (`id`) ON DELETE NO  
    ACTION ON UPDATE NO ACTION,
```

```
    ADD CONSTRAINT `fk_llamada_llamada_submotivo1` FOREIGN KEY  
    (`submotivo_id`) REFERENCES `llamada_submotivo` (`id`) ON DELETE NO  
    ACTION ON UPDATE NO ACTION;
```

```
--
```

```
-- Filtros para la tabla `llamada_submotivo`
```

```
--
```

```
ALTER TABLE `llamada_submotivo`
```

```
    ADD CONSTRAINT `fk_llamada_submotivo_llamada_motivo1` FOREIGN KEY  
    (`motivo_id`) REFERENCES `llamada_motivo` (`id`) ON DELETE NO ACTION  
    ON UPDATE NO ACTION
```

BIBLIOGRAFIA

SLIDER (2014). Información acerca de Arquitectura 3 capas. [En Línea] [Consultado el 01 de Noviembre 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 Noviembre del 2014]: Disponible en: http://www.kmkey.com/productos/software_help_desk

PROYECTOSAGILES (2013) Help Desk [En Línea] [Consultado el 01 de Noviembre del 2014]: Disponible en: <http://www.proyectosagiles.org/que-es-scrum>

OPENACCESS (2014) Metodología SCRUM [En Línea] [Consultado el 01 de Noviembre del 2014]: Disponible en:
<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>

BIBLIOTECA UNIRIOJA (2014) Metodología SCRUM [En Línea] [Consultado el 01 de Noviembre del 2014]: Disponible en: http://biblioteca.unirioja.es/tfe_e/TFE000283.pdf

SCRIBD (2014) Metodología ITIL [En Línea] [Consultado el 01 de Noviembre del 2014]: Disponible en: <http://es.scribd.com/doc/30511024/Metodologia-ITIL#scribd>