

# **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**



## **Facultad de Ingeniería en Electricidad y Computación**

“WORK FORCE ECUADOR

(SISTEMA DE GESTIÓN DE INCIDENCIAS PARA ERICSSON®  
DEL ECUADOR)”

### **EXAMEN DE GRADO (COMPLEXIVO)**

Previo a la obtención del título de:

**INGENIERO EN CIENCIAS COMPUTACIONALES  
ORIENTACIÓN SISTEMAS TECNOLÓGICOS**

**GEOVANNY ENRIQUE TRUJILLO DELGADO**

**GUAYAQUIL – ECUADOR**

**AÑO: 2015**

## **AGRADECIMIENTO**

Mi familia, que me dio todo el apoyo necesario para culminar mi carrera y a todos mis profesores de la ESPOL que compartieron sus conocimientos para ser de mí el profesional que soy ahora.

## DEDICATORIA

El presente proyecto lo dedico a la memoria de mi madre.

Por todas las veces que estuviste para apoyarme en ti. Por toda la verdad que me hiciste ver. Por toda la alegría que trajiste a mi vida. Por todos los errores que convertiste en aciertos. Por cada sueño que hiciste realidad. Por todo el amor que encontré en ti, estaré por siempre agradecido mamá. Tú fuiste la única que me sostuvo y nunca me dejó caer. Soy todo lo que soy, porque siempre me amaste.

## TRIBUNAL DE SUSTENTACIÓN

---

**MSIG. Jorge Rodríguez**

PROFESOR DELEGADO POR LA  
UNIDAD ACADÉMICA

---

**M.Sc. Cruz María Falcones**

PROFESOR DELEGADO POR LA  
UNIDAD ACADÉMICA

## **DECLARACIÓN EXPRESA**

“La responsabilidad del contenido de este Informe me corresponde exclusivamente; y, el patrimonio intelectual del mismo, a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”

(Reglamento de Graduación de la ESPOL).

.....  
Geovanny E Trujillo Delgado

## RESUMEN

Ericsson® del Ecuador en su deseo de mejorar el proceso de gestión de incidencias reportadas por parte de su cliente Claro®, sobre la red HFC (red híbrida de cable coaxial y fibra óptica) que sostiene los servicios de telefonía fija, televisión por cable e internet en las ciudades de Guayaquil, Quito, Manta, Ibarra y Machala, se ha visto en la necesidad de la implementación de un sistema de software que permita automatizar la administración y gestión de éstas incidencias de una manera rápida y eficiente acorde a sus procesos operativos, y que permita tener en tiempo real indicadores de gestión de productividad en el transcurso del manejo de cada incidencia. Por tal motivo se establece como estrategia el desarrollo de una mesa de ayuda con un ciclo de vida de desarrollo de software en espiral debido a que este modelo de desarrollo se adapta en gran parte a las necesidades del cliente y las características solicitadas para el sistema, todo esto con la finalidad de obtener en tiempo real diferentes KPIs (Key Performance Indicator) que Ericsson® requiere en sus operaciones y así asegurar una alta disponibilidad de la planta que presta sus servicios a la operadora Claro®.

## ÍNDICE GENERAL

AGRADECIMIENTO .....	ii
DEDICATORIA.....	iii
TRIBUNAL DE SUSTENTACIÓN .....	iv
DECLARACIÓN EXPRESA .....	v
RESUMEN.....	vi
ÍNDICE GENERAL.....	vii
ÍNDICE DE FIGURAS .....	x
INTRODUCCIÓN.....	xi
CAPÍTULO 1	1
1. METODOLOGÍA O SOLUCIÓN TECNOLÓGICA IMPLEMENTADA.....	1
1.1 Desarrollo en espiral. ....	2
1.1.1 Definición de objetivos.....	2
1.1.2 Evaluación y reducción de riesgos. ....	3
1.1.3 Desarrollo y validación. ....	3
1.1.4 Planificación .....	3
1.2 Identificación de tareas principales en el modelo en espiral .....	5
1.2.1 Comunicación con el cliente .....	5
1.2.2 Planificación .....	5
1.2.3 Análisis de riesgos .....	6

1.2.4	Ingeniería .....	6
1.2.5	Construcción y adaptación .....	6
1.2.6	Evaluación el cliente.....	6
1.3	Estado actual del proceso de gestión .....	6
1.4	Requerimientos identificados.....	8
1.5	Proceso de atención de incidencias .....	10
1.5.1	Alarmas Reactivas .....	10
1.5.2	Alarmas por Análisis.....	10
1.5.3	Alarma por Notificación de Claro® .....	10
1.6	Tecnología utilizada .....	14
1.6.1	PHP (Hypertext Pre-processor) .....	14
1.6.2	Elastix .....	16
1.6.3	MySQL .....	16
1.6.4	HTML5 .....	17
1.6.5	JQuery .....	18
1.7	Metáfora de interfaz de una mesa de ayuda.....	18
1.8	Propuesta tecnológica.....	20
1.8.1	Servidor de Alojamiento .....	21
1.8.2	Clientes locales/externos.....	21
1.8.3	Esquema de Software .....	22



CAPÍTULO 2	24
2. RESULTADOS OBTENIDOS.....	24
2.1 Estructura del Sistema WF-EC.....	24
2.2 Descripción de los componentes de WF-EC .....	25
2.2.1 Indicadores (Indicadores Clave de Desempeño) .....	25
2.2.2 Alertas y Notificaciones .....	26
2.2.3 Reportes .....	27
2.2.4 Seguimiento .....	28
2.2.5 Gestión de RRHH y Planificación .....	29
2.2.6 Web Service.....	29
2.2.7 Perfiles .....	30
2.2.8 Persistencia.....	32
2.2.9 Administración y Mantenimiento.....	32
CONCLUSIONES Y RECOMENDACIONES .....	34
BIBLIOGRAFÍA .....	36
ANEXOS	37

## ÍNDICE DE FIGURAS

Figura 1.1 Diagrama de modelo en espiral de Boehm .....	5
Figura 1.2 Diagrama de Bloques de la Gestión de Incidencias .....	9
Figura 1.3 Diagrama de Flujo de la Gestión de incidencias .....	13
Figura 1.4 Ciclo de vida en diagrama de actividades .....	14
Figura 1.5 Ciclo de vida de un ticket .....	20
Figura 1.6 Esquema de Hardware .....	22
Figura 1.7 Esquema de Software.....	23
Figura 2.1 Diagrama de componentes .....	25
Anexo 1 Modelo relacional de la BD .....	38
Anexo 2 Dashboard del WF-EC.....	39

## INTRODUCCIÓN

Anteriormente la gestión de incidencias reportadas por Claro® era realizada de manera manual a través de hojas de cálculo que permitían la administración de forma funcional pero no con eficiencia ni con eficacia, por lo que se propuso desarrollar un software para la administración de estas incidencias. En base a los conocimientos académicos adquiridos y conversaciones con el cliente para el levantamiento de información se estableció como estrategia de desarrollo de este proyecto de software, el modelo en espiral de Boehm [1], debido a que el proceso de gestión de incidencias se lo puede representar como una secuencia de actividades, es decir como una espiral en la que cada ciclo en la espiral representa una fase del proceso del desarrollo del software; el ciclo más interno podría referirse a la viabilidad del sistema, el siguiente ciclo a la definición de requerimientos, el siguiente ciclo al diseño del sistema, y así sucesivamente. También se decidió la utilización de herramientas de software libre de última generación como una manera de abaratar costos en el desarrollo del proyecto, siendo una de estas herramientas el Framework Elastix®, que permitió programar por medio de módulos las diferentes funciones requeridas según las necesidades indicadas por el cliente.

El principal resultado esperado mediante el desarrollo de este proyecto de software fue la generación automática de indicadores claves de rendimiento o KPI por sus siglas en inglés (Key Performance Indicator), los cuales habían sido generados de manera manual mediante hojas de cálculo. Todos estos indicadores vienen como resultado de la gestión de

incidencias por medio de la implementación de un método más ágil para su administración a través de su representación abstracta o metáfora de interfaz se ha definido como TICKET, y como parte de la solución un módulo de gestión del personal del centro de operaciones, un módulo de gestión de técnicos de campo y un módulo de monitoreo de incidencias clasificados por criticidad o prioridad.

## **CAPÍTULO 1**

### **1. METODOLOGÍA O SOLUCIÓN TECNOLÓGICA IMPLEMENTADA**

El proceso de desarrollo de software en sí es complejo, existiendo una gran combinación de factores que dificultan realizar una verificación exhaustiva de todas los posibles escenarios; por ejemplo la creación de un sistema operativo, la cual es una tarea que requiere proyección, gestión de gran cantidad de recursos y todo un equipo disciplinado de trabajo. Así también la complejidad parte desde la definición del producto y sus requisitos, más aún cuando no se tiene algún otro software similar como referencia. Esta situación hace que los requisitos sean difíciles de consolidar desde un inicio y que los cambios durante todo el proceso sean inevitables, sea cual fuese la metodología utilizada.

Según un artículo de investigación de la Universidad Carlos III de Madrid sobre el análisis de riesgos en la gestión de proyectos [2] se estima que del total de proyectos software grandes emprendidos, el 60% de los riesgos identificados

tienen consecuencias graves, por lo que cuando un proyecto de desarrollo de software “fracasa” muy rara vez es causado por fallas técnicas y principalmente el origen de los fallos y fracasos es la falta de aplicación de una buena metodología o procesos de desarrollo. Una fuerte tendencia, desde hace algunos años, es mejorar las metodologías y procesos, o crear nuevas e incentivar a los profesionales de la informática en su aplicación adecuada debido a que normalmente utilizan sus conocimientos especializados con modelos, paradigmas y procesos obsoletos ya diseñados.

### **1.1 Desarrollo en espiral.**

El modelo en espiral del proceso del software fue originalmente propuesto por Boehm (en su artículo “A Spiral Model of Software Development and Enhancement” [3], 1988). Más que representar el proceso del software como una secuencia de actividades con retrospectiva de una actividad a otra, se representa como una espiral. Cada ciclo en la espiral representa una fase del proceso del software. Así el ciclo más interno podría referirse a la viabilidad del sistema, el siguiente ciclo a la definición de requerimientos, el siguiente ciclo al diseño del sistema, y así sucesivamente.

Cada ciclo de la espiral se divide en cuatro sectores:

#### **1.1.1 Definición de objetivos.**

Para esta fase del proyecto se definen los objetivos específicos. Se identifican las restricciones del proceso y el producto, y se traza un plan detallado de gestión. Adicionalmente, se identifican los riesgos del

proyecto y dependiendo de estos riesgos se planean estrategias alternativas.

#### **1.1.2 Evaluación y reducción de riesgos.**

Se lleva a cabo un análisis detallado para cada uno de los riesgos identificados en el proyecto, definiéndose los pasos para reducirlos, por ejemplo, si existe el riesgo de tener requerimientos inapropiados entonces se puede desarrollar un prototipo del sistema.

#### **1.1.3 Desarrollo y validación.**

Después de la evaluación de riesgos, se elige un modelo para el desarrollo del sistema. Por ejemplo, si los riesgos en la interfaz de usuario son dominantes, un modelo de desarrollo apropiado podría ser la construcción de prototipos evolutivos. Si los riesgos de seguridad son la principal consideración, un desarrollo basado en transformaciones formales podría ser el más apropiado, y así sucesivamente. El modelo en cascada puede ser el más apropiado para el desarrollo si el mayor riesgo identificado es la integración de los subsistemas.

#### **1.1.4 Planificación**

El proyecto se revisa y se toma la decisión de si se debe continuar con un ciclo posterior de la espiral. Si se decide continuar entonces se desarrollan los planes para la siguiente fase del proyecto.

La diferencia principal entre el modelo en espiral y los otros modelos del proceso del software es la consideración explícita del riesgo en el modelo. Informalmente, el riesgo significa algo que puede ir mal, por ejemplo si la intención es utilizar un nuevo lenguaje de programación, un riesgo es que los compiladores disponibles sean poco fiables o que no produzcan código objeto eficiente. Los riesgos originan problemas en el proyecto, como los de confección de agendas y excesos en los costos; por lo tanto, la disminución de riesgos es una actividad muy importante en la gestión del proyecto.

Un ciclo de la espiral empieza con la elaboración de objetivos, como el rendimiento y la funcionalidad. Entonces se enumeran formas alternativas de alcanzar estos objetivos y las restricciones impuestas en cada una de ellas. Cada alternativa se evalúa contra cada objetivo y se identifican las fuentes de riesgo del proyecto. El siguiente paso es resolver estos riesgos mediante actividades de recopilación de información como la de detallar más el análisis, la construcción de prototipos y la simulación. Una vez que se han evaluado los riesgos, se lleva a cabo cierto desarrollo seguido de una actividad de planificación para la siguiente fase del proceso, ver Figura1.1.





### **1.2.3 Análisis de riesgos**

Esta es una de las tareas principales por lo que se aplica el modelo en espiral. Es requerida para evaluar los riesgos técnicos y otras informaciones relacionadas con el proyecto.

### **1.2.4 Ingeniería**

Esta es una tarea necesaria ya que se requiere construir una o más representaciones de la aplicación.

### **1.2.5 Construcción y adaptación**

Esta tarea es requerida en el modelo espiral porque se necesita construir, probar, instalar y proporcionar soporte al usuario.

### **1.2.6 Evaluación el cliente**

Esta también es una de las tareas principales. Es necesaria para adquirir la reacción del cliente según la evaluación de las representaciones del software creadas durante la etapa de ingeniería y la de implementación creada durante la etapa de instalación.

## **1.3 Estado actual del proceso de gestión**

El proceso de gestión de una incidencia puede iniciarse de varias maneras, las principales son:

- Reporte por parte del Front Office de Claro®.
- Detección de una alarma por parte del personal del Network Operator Center (NOC).

El reporte de incidencia por parte del Front Office de Claro® se lo realiza a través de un correo electrónico enviado al personal de NOC-ERICSSON, el cual se encarga de realizar las actividades respectivas para la asignación de los técnicos de campo y la resolución por parte de los mismos. Todo esto se lo ejecuta utilizando diferentes herramientas tecnológicas que ayudan a la gestión pero no lo hacen eficiente. De igual manera sucede con la gestión cuando la detección de una alarma se da por parte del personal de NOC-ERICSSON y se debe utilizar el sistema de gestión de Claro®, el cual es una herramienta externa que es administrada por agentes externos a Ericsson® lo que hace que su disponibilidad no sea del 100% y se vean afectados sus indicadores de productividad.

Otro de los problemas identificados es la administración de recursos humanos para la gestión de las incidencias la cual también se realiza de manera manual, acarreando que el tiempo en que el personal es asignado a los grupos de trabajo que dan soporte para las incidencias reportadas sea muy extenso.

Otra de las actividades que NOC-ERICSSON realiza manualmente es la planificación de tareas de prevención y mantenimientos.

El análisis de indicadores se realizaba de manera manual utilizando hojas de cálculo para crear los gráficos estadísticos que reflejan disponibilidad, casos atendidos, indicadores claves de productividad internos, tiempos de solución, recurrencia entre otros.

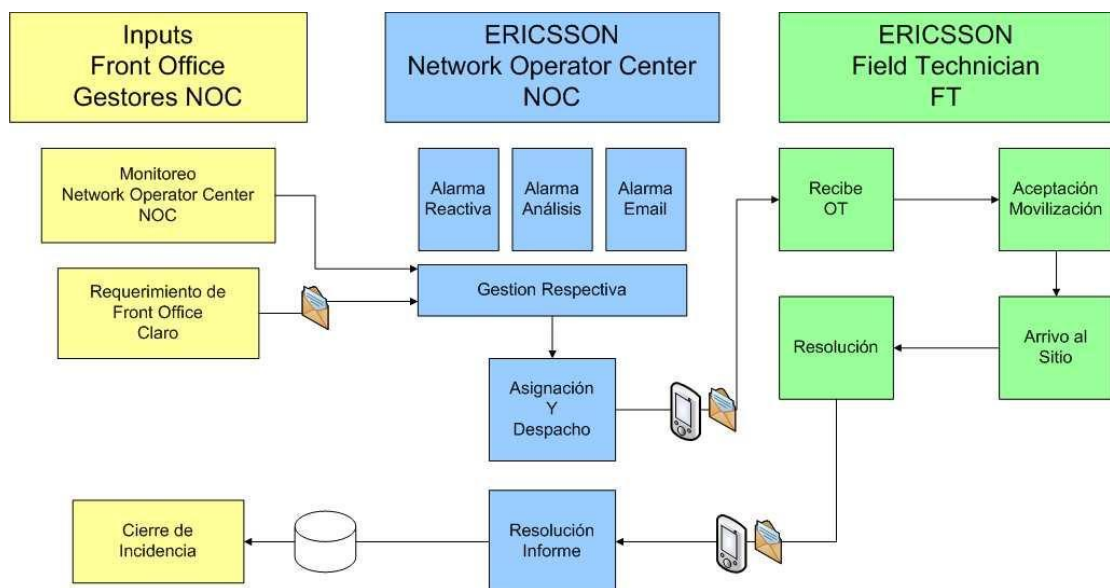
#### 1.4 Requerimientos identificados

En base a las entrevistas mantenidas con Ericsson, se han identificado los siguientes requerimientos que engloban la gestión de incidencias y que permitieron agilizar el proceso incrementando la productividad de los operadores involucrados. Entre los requerimientos se tiene:

- Minutos acumulados de afectación por ciudad
- Minutos acumulados de afectación por ciudad sin "Waiting External"
- Tickets con y sin afectación por HUB
- Tickets con y sin afectación por Ciudad
- Back Log diario agrupado por Ciudad
- Total de tickets abiertos y cerrados
- Minutos de Afectación agrupados por HUB
- Minutos de Afectación agrupados por Ciudad
- Casos según el factor de riesgo
- Minutos de afectación según el factor de riesgo
- Casos según Causa Raíz
- Impacto de disponibilidad por Causa Raíz
- Prioridades de los tickets
- Gestión de Tickets
- Gestión acumulada de tickets por Semana
- Productivity Factors of Field technician (FT)
- Resolved tickets by FT
- Productivity Factors of Network Operator Center (NOC)

- Gestión de seguimiento de una incidencia en cualquier instante del tiempo que se está desarrollando
- Gestión de Recursos Humanos del NOC-ERICSSON
- Gestión de Recursos Humanos de FT-ERICSSON.
- Indicadores del Proyecto en función de la información de Tickets
- Alertas para el grupo de despacho NOC
- Alerta de los estados de los tickets

Para la implementación de estos requerimientos es necesario describir el proceso de atención de una incidencia desde el punto de vista de un “ticket” (TKT) que maneja su proceso de vida, es decir desde su generación hasta su solución, como se muestra en la Figura 1.2.



**Figura 1.2: Diagrama de Bloques de la Gestión de Incidencias**

## **1.5 Proceso de atención de incidencias**

En base a los tres tipos de alarmas que pueden generar una incidencia se describe a continuación el proceso de atención de incidencias y como WF-EC automatizará dichos procesos.

### **1.5.1 Alarmas Reactivas**

Cuando la gestión del NOC detecta la existencia de una incidencia por medio del sistema de monitoreo de la red HFC, se inicia de forma automática la creación de un ticket asociado a dicha incidencia, para esto se genera un código NOC de un ticket y se notifica al Front Office de Claro® a la vez que se solicita el código RT para el ticket creado.

### **1.5.2 Alarmas por Análisis**

El operador del NOC, por medio del análisis del sistema de monitoreo de la red HFC, identifica la existencia de irregularidades dentro de la red HFC y por tanto determina la existencia de una incidencia, para lo cual inicia la creación de un ticket a la vez que notifica al Front Office de Claro® y solicita el Código RT para el ticket creado.

### **1.5.3 Alarma por Notificación de Claro®**

El operador del NOC, recibe un correo electrónico por parte de Claro®, el cual reporta una incidencia, que es ingresada en el sistema de gestión de incidencias de Claro®, y por lo tanto también debe ser

creado un ticket con código NOC que es asociado a la incidencia reportada.

#### **1.5.4 Registro de alarma**

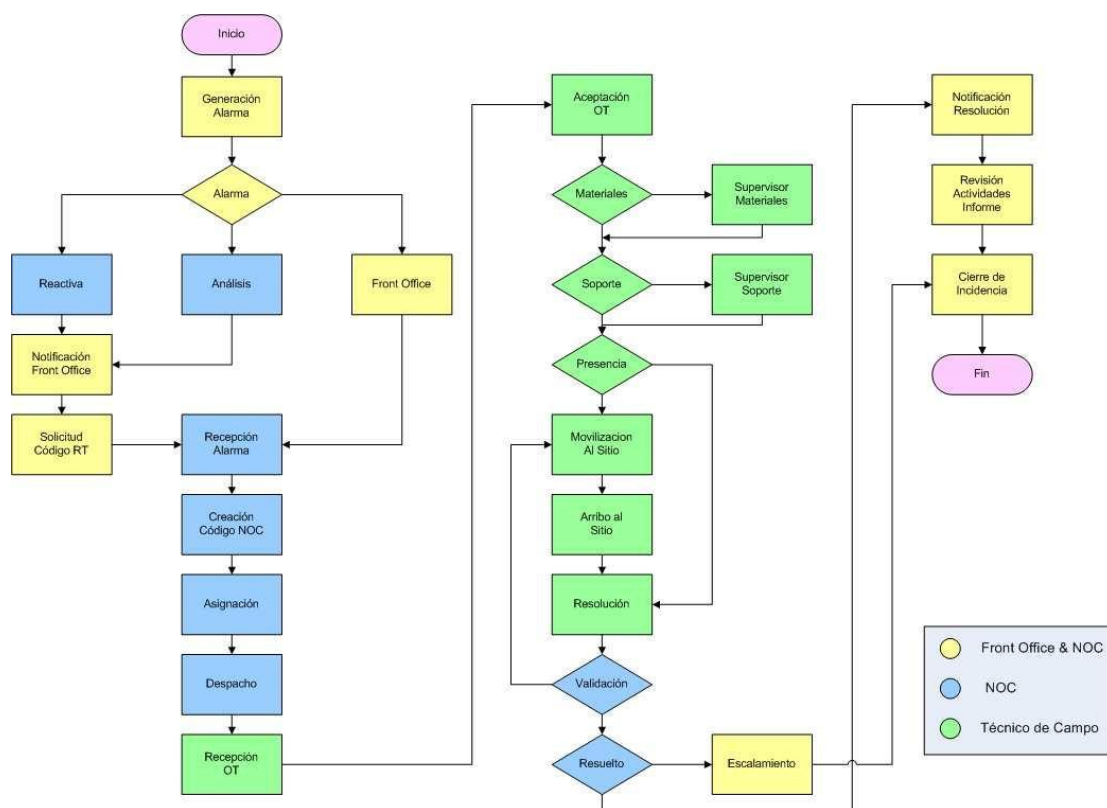
Luego de que se ha reconocido la alarma y se ha creado el ticket dentro del sistema WorkForce Ericsson (WF-EC), el sistema notificará por medio de correo electrónico a las personas involucradas dentro de la gestión, empezando así el conteo del tiempo desde el reconocimiento de la alarma hasta la asignación de la orden de trabajo. Una vez que el operador del NOC asigna la orden de trabajo dentro del sistema WF-EC, en función del nodo que se ve afectado, el indicador del tiempo de asignación se detiene para controlar los indicadores de productividad del NOC.

La asignación del grupo de trabajo previamente ha sido realizada por el supervisor de técnicos de campo en el módulo de planificación semanal y de gestión de RRHH con el que contará sistema WF-EC. Esta asignación se realizará internamente en el sistema WF-EC y por medio de una notificación de correo electrónico se informará al líder del grupo de trabajo que tiene una orden de trabajo asignada y que debe ser aceptada. El líder del grupo de trabajo deberá acceder al sistema WF-EC por medio de un navegador de internet y aceptar la orden de trabajo, para de esta manera comience a transcurrir un indicador de productividad que controla el tiempo de movilización del técnico de

campo, hasta cuando este llegue al sitio de la incidencia y marque su arribo por medio del sistema WF-EC. Luego de realizada esta actividad se pone en marcha el indicador de tiempo de solución de la incidencia, el cual se detiene una vez que se la ha resuelto. Una vez resuelta la incidencia, se informa vía telefónica y por correo electrónico al operador del NOC-ERICSSON que en conjunto con el técnico de campo complete la información necesaria del reporte de actividades y dejando la incidencia como resuelta.

El siguiente diagrama de flujo, Figura 1.3, representa todo el proceso de atención de la incidencia según los participantes dentro de cada acción.





**Figura 1.3: Diagrama de Flujo de la Gestión de incidencias**

Una vez que se ha descrito el proceso de atención dentro de la gestión realizada por los operadores del NOC-ERICSSON, se puede definir un ciclo de vida para los tickets que manejan cada incidencia.

En la figura 1.4 se muestra el ciclo de vida de un ticket como gestión de una incidencia, desde el reconocimiento de la alarma hasta la solución de la incidencia y el cierre de la misma.

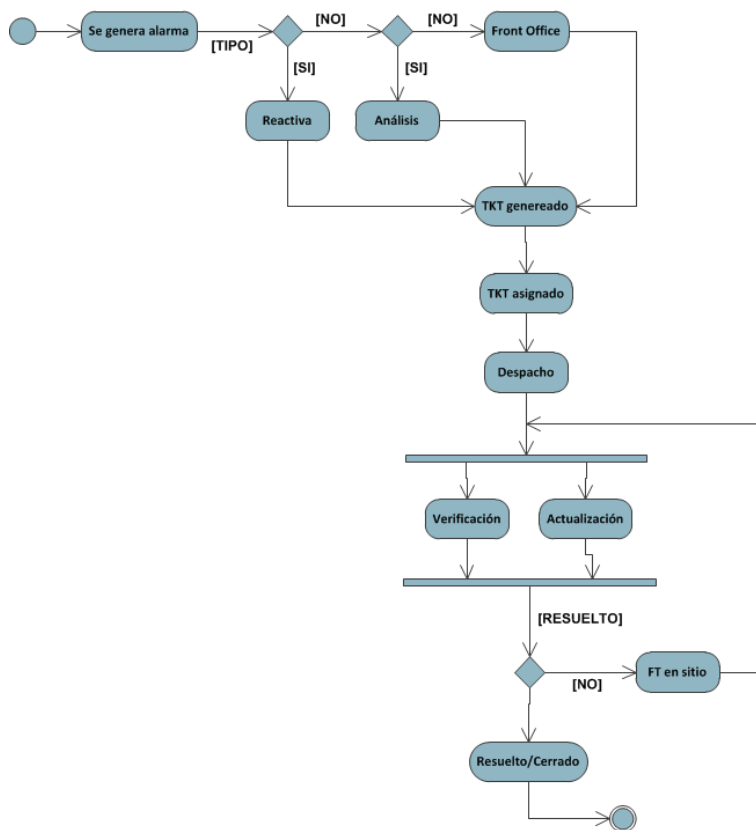


Figura 1.4: Ciclo de vida en diagrama de actividades

## 1.6 Tecnología utilizada

### 1.6.1 PHP (Hypertext Pre-processor)

PHP [4] tal y como se conoce hoy en día es en realidad el sucesor de un producto llamado PHP/FI. Creado en 1994 por Rasmus Lerdorf, la primera encarnación de PHP era un conjunto simple de ficheros binarios Common Gateway Interface (CGI) escritos en el lenguaje de programación C, al que llamó "Personal Home Page Tools", más frecuentemente referenciado como "PHP Tools". Con el paso del tiempo se quiso más funcionalidad, y Rasmus reescribió PHP Tools,

produciendo una implementación más grande y rica. Este nuevo modelo fue capaz de interactuar con bases de datos, y mucho más, proporcionando un entorno de trabajo sobre cuyos usuarios podían desarrollar aplicaciones web dinámicas sencillas tales como libros de visitas. En junio de 1995, Rasmus » publicó el código fuente de PHP Tools, lo que permitió a los desarrolladores usarlo como considerasen apropiado. Esto también permitió -y animó- a los usuarios a proporcionar soluciones a los errores del código, y generalmente a mejorarlo.

PHP 5 fue lanzado en Julio del 2004 después de un largo desarrollo y varias liberaciones previas. Está básicamente impulsado por su núcleo, Zend Engine 2.0 que contiene un nuevo modelo de objetos y docenas de nuevas opciones.

El equipo de desarrollo de PHP incluye docenas de desarrolladores, así como docenas de otras personas trabajando en proyectos relacionados y de soporte para PHP, como PEAR, PECL, y documentación, y una infraestructura en red subyacente de más de cien servidores web individuales en seis de los siete continentes del mundo.

### 1.6.2 Elastix

Elastix [5] es una distribución libre de Servidor de Comunicaciones Unificadas que integra en un solo paquete:

- VoIP PBX
- Fax
- Mensajería Instantánea
- Correo electrónico
- Colaboración

Elastix implementa gran parte de su funcionalidad sobre cuatro programas de software muy importantes como son Asterisk, Hylafax, Openfire y Postfix. Estos brindan las funciones de PBX, Fax, Mensajería Instantánea y Correo electrónico respectivamente. Elastix corre sobre CentOS como sistema operativo y actualmente su versión más estable es Elastix 2.4.0.

### 1.6.3 MySQL

MySQL [6] es un sistema de gestión de bases de datos relacional, multihilo y multiusuario que es desarrollada por Oracle Corporation desde abril del 2009, como software libre en un esquema de licenciamiento dual.

Podremos utilizar la licencia GPL de MySQL siempre que el programa que lo use también lo sea, en caso contrario se debe adquirir la “licencia comercial” que les permita su uso.

Inicialmente, MySQL carecía de algunos elementos esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de esto, atrajo a los desarrolladores de páginas web con contenido dinámico, debido a su simplicidad, de tal manera que los elementos faltantes fueron complementados por la vía de las aplicaciones que la utilizan. Poco a poco estos elementos faltantes, han sido incorporados tanto por desarrolladores internos, como por desarrolladores de software libre.

#### **1.6.4 HTML5**

HTML5 [7] (HyperText Markup Language, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, es el lenguaje básico de la web para la creación de contenido.

HTML5 trae a la web, vídeo y pistas de audio sin necesidad de plugins; acceso programático a un lienzo de mapa de bits que es útil para la prestación de los gráficos, gráficos de juegos, u otras imágenes visuales en tiempo de ejecución; soporte nativo para gráficos vectoriales escalables (SVG) y matemáticas (MathML); anotaciones importantes para la tipografía de Asia Oriental (Rubí); características que permitan la accesibilidad de aplicaciones ricas; y mucho más.

### **1.6.5 JQuery**

jQuery [8] es una biblioteca JavaScript rápida, pequeña y rica en funciones que hace cosas como recorrido HTML, manipulación de documentos, manejo de eventos, animación y Ajax de una manera mucho más simple con una API fácil de usar que funciona a través de una multitud de navegadores logrando grandes resultados en menos tiempo y espacio.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados

### **1.7 Metáfora de interfaz de una mesa de ayuda**

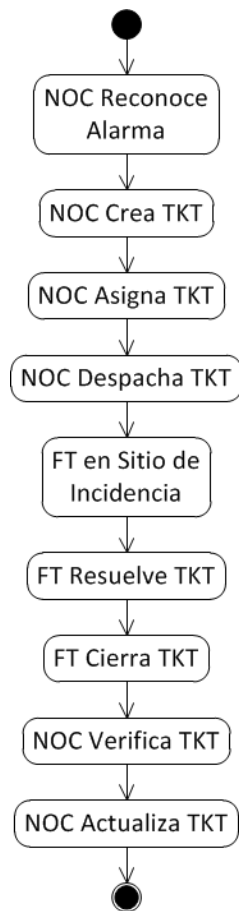
Se establece como método de solución para el proyecto propuesto una metáfora de Mesa de Ayuda (en inglés: Help Desk), o Mesa de Servicio (Service Desk) que se define como un conjunto de recursos tecnológicos y humanos, para prestar servicios con la posibilidad de gestionar y solucionar todas las posibles incidencias de manera integral, junto con la atención de requerimientos relacionados a las Tecnologías de la Información y la Comunicación (TIC).

El personal o recurso humano encargado de Mesa de Ayuda (MDA) debe proporcionar respuestas y soluciones a los usuarios finales, clientes o beneficiarios (destinatarios del servicio), así como dar asesoramiento en

relación con una organización o institución, productos y servicios. El propósito de la mesa de ayuda es solucionar problemas u orientar acerca de computadoras, equipos electrónicos o software.

Las organizaciones suelen proporcionar soporte de mesa de ayuda a sus usuarios a través de varios canales, como números de teléfono gratuitos, sitios web, mensajería instantánea o correo electrónico. Y también brindar asistencia con miras a los usuarios o empleados, dentro de la organización. Por lo tanto, los usuarios finales de esta aplicación pueden ser internos o ajenos a la organización donde se encuentre la mesa de ayuda.

Para gestionar las incidencias se establece un ciclo de vida para un ticket que se ha definido en conjunto con el cliente y se presenta en la Figura 1.5.



**Figura 1.5: Ciclo de vida de un ticket**

### **1.8 Propuesta tecnológica.**

Una vez conocida la necesidad de Ericsson® por medio de las entrevistas realizadas, se propone una solución tecnológica que cubra esta necesidad abarcando aspectos de Hardware y Software detallados a continuación.



### **1.8.1 Servidor de Alojamiento**

Como requerimiento de hardware para que el sistema funcione y sea alojado, se requiere un equipo con las siguientes características mínimas:

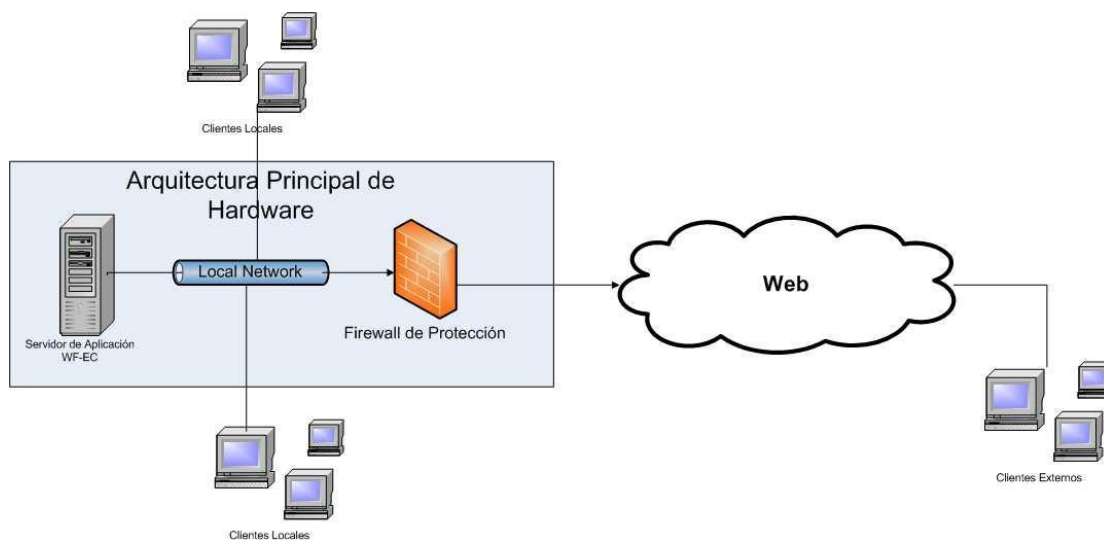
- Procesador Intel® Core™ i5
- Memoria RAM de 6GB
- Disco Duro de 1TB
- Tarjeta de red 10/100MB
- Equipo dentro de red LAN con Firewall y salida a Internet

### **1.8.2 Clientes locales/externos**

Para poder acceder al sistema, las características mínimas de hardware para un computador deben ser las siguientes:

- Procesador Intel® Dual Core™
- Memoria RAM de 2GB
- Tarjeta de red 10/100MB
- Disco Duro de 500GB
- Pertenecer a la red LAN del servidor o tener acceso a Internet

En la Figura 1.6 se puede apreciar una representación gráfica del esquema que se usó para el desarrollo de la propuesta tecnológica en cuanto a Hardware.



**Figura 1.6: Esquema de Hardware**

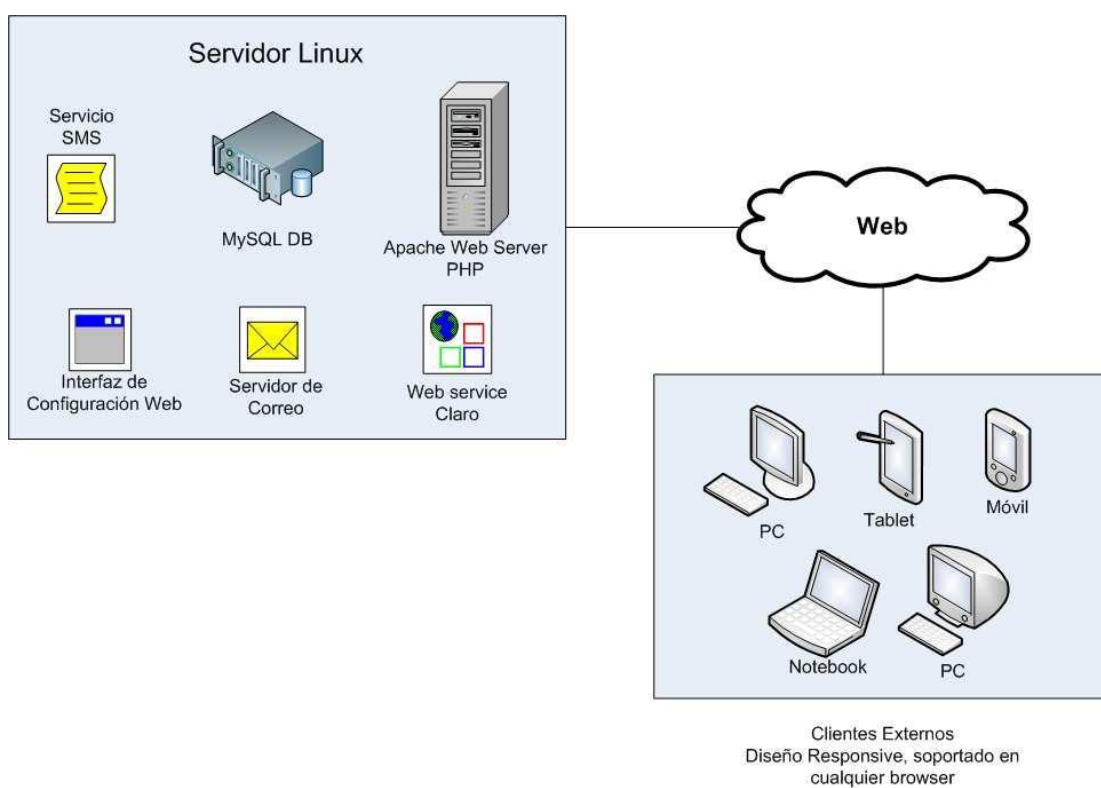
### 1.8.3 Esquema de Software

La arquitectura de Software para el servidor en la que se alojara el proyecto WF-EC requiere las siguientes características:

- Sistema Operativo Linux
- Servidor Apache 2
- PHP versión 5.5
- Servidor de Base de Datos MySQL
- Servidor de Correo Postfix
- Servicio de envío de SMS
- Webservice desarrollado para actualizar datos desde el sistema de Gestión de Claro®

- Browser Mozilla Firefox, Google Chrome, Internet Explorer (En sus últimas versiones)

En la Figura 1.7 se puede apreciar una representación gráfica del esquema de software que se propuso como solución tecnológica.



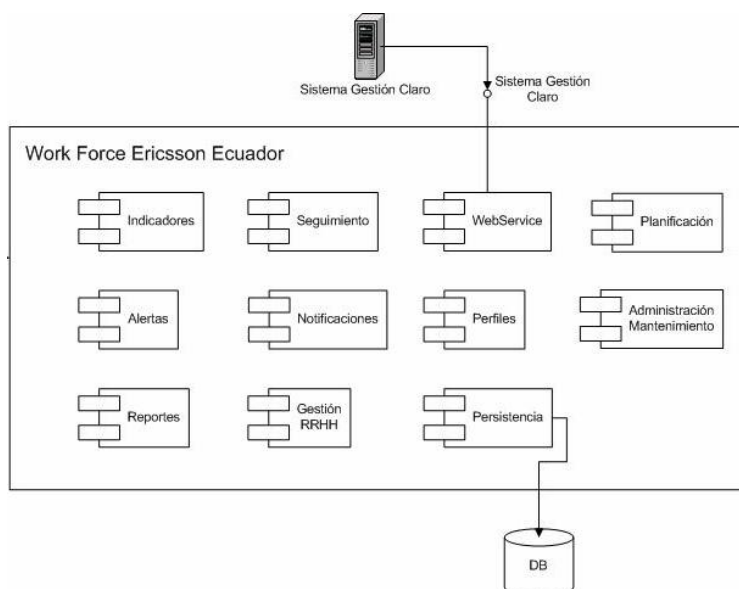
**Figura 1.7: Esquema de Software**

## **CAPÍTULO 2**

### **2. RESULTADOS OBTENIDOS**

#### **2.1 Estructura del Sistema WF-EC**

El sistema WF-EC fue diseñado como una aplicación web responsive, que permite ser ejecutada sobre cualquier plataforma de hardware/software, utilizando tecnología PHP como motor de ejecución de la aplicación. Se utilizó un motor de base de datos MySQL para manejar la persistencia de los datos, en Anexo 1 se puede observar el modelo relacional inicial de la base de datos, adicionalmente se usó servicios como correo electrónico o de notificación vía SMS. En la Figura 2.1 se muestra el diagrama de componentes globales del sistema WF-EC.



**Figura 2.1: Diagrama de componentes**

## 2.2 Descripción de los componentes de WF-EC

Para el desarrollo del Sistema WF-EC se automatizará el proceso de gestión distribuyendo las tareas en módulos o componentes que serán especializados en las diferentes actividades así se tendrá los siguientes componentes:

### 2.2.1 Indicadores (Indicadores Clave de Desempeño)

Permitirá en cualquier momento dado consultar indicadores de productividad de los operadores del NOC-ERICSSON así como

también de los Técnicos de Campo. Un ejemplo de estos es el identificar el número de tickets que han sido gestionados por parte de un operador de NOC-ERICSSON, o los Operadores que han participado del seguimiento de un ticket.

Se han establecido los tickets y sus diferentes estados:

- Reconocido
- Asignado
- Despachado
- Aceptado
- En Sitio
- Resuelto
- Cerrado
- Cancelado
- Waiting External
- Waiting Internal

Estos estados y su duración en la línea de tiempo, definirán indicadores para los técnicos de campo, como para los operadores del NOC-ERICSSON.

### **2.2.2 Alertas y Notificaciones**

Este módulo administrará las alertas generales de las personas involucradas dentro del proceso de gestión, informando

periódicamente a los operadores del NOC y del sistema WF-EC los estados de tickets abiertos.

Al operador del NOC, se le mostrarán las alarmas cuando los tiempos de control para el estado de cada ticket se hayan vencido. Además se le mostrará alarmas cuando su tiempo de asignación se ha vencido.

Para los Técnicos de Campo, así mismo se mostraran alertas indicando los tiempos de vencimiento de cada estado, según los tiempos del ticket. Adicionalmente, se incluye en el módulo el envío de correos electrónicos para informar a las personas involucradas en el proceso

### **2.2.3 Reportes**

A través de este módulo se podrá obtener los reportes que necesariamente deben ser presentados al cliente Claro, como también a las personas involucradas en la gestión de incidencias.

A continuación, se listan los más importantes que se han definido dentro del sistema:

- Indicadores Clave de Desempeño Interno
- Recurrencia

- Estado de los Tickets (Open, Resolved, New) Diario, Semanal, Mensual
- Tickets con afectación, Tickets sin Afectación, Porcentajes
- Disponibilidad por ciudad, diario, semanal y mensual
- Disponibilidad por HUB, diario, semanal y mensual
- Incidencias por Estado
- Disponibilidad Ponderada vs Disponibilidad en bruto
- Porcentaje de Disponibilidad
- Tiempos de Afectación
- Causa Raíz por ciudad
- Tiempo de Tickets por ciudad
- Afectación por minutos HP
- Tiempo de Afectación por HUB-Anillo-Enlace-Tramo
- HP por Ciudad-HUB-Nodo
- Tiempo de solución
- Promedio de tiempo de afectación
- Porcentajes de afectación
- Seguimiento de Actividades (Tracking)

#### **2.2.4 Seguimiento**



Es necesario para los operadores del NOC o supervisores, tener a la mano de manera rápida y ágil el estado de un ticket, así como el historial de este y poder tomar decisiones. Este módulo permitirá realizar este seguimiento a través de la retroalimentación de cada uno de los estados de un ticket asociado a una incidencia indicando su historial y actividades que cada uno de los operadores/técnicos de campo ha realizado sobre el ticket.

#### **2.2.5 Gestión de RRHH y Planificación**

La asignación de incidencias a grupos de trabajo deberá ser realizada en función de los elementos humanos dentro del sistema y de su correcta gestión, este módulo permitirá realizar la planificación necesaria para gestionar el recurso humano dentro del sistema WFEC, también contempla la planificación de actividades preventivas o correctivas calendarizándolas a través de este módulo.

#### **2.2.6 Web Service**

Para mantener una sincronización adecuada de las incidencias reportadas por Claro dentro de su sistema de gestión, WF-EC

contará con un servicio web que constantemente revisará dicha información y reflejará el estado de la misma en el sistema WF-EC. Como requerimiento para el desarrollo de este servicio web es necesario que los datos del sistema de gestión de Claro puedan ser consultados a través de una URL pública, obteniendo como resultado a la consulta, un archivo XML que contenga todos los datos creados al momento de crear una incidencia, identificando cada tag de composición del archivo con el nombre del campo dentro del sistema de Gestión de Claro para poder sincronizar dichos datos con la información que maneja las incidencias dentro del sistema WF-EC.

### 2.2.7 Perfiles

Dentro del sistema WF-EC, se definirá diferentes roles o perfiles para los usuarios que operen dentro del sistema, asignando permisos para ejecutar ciertas tareas o actividades, y así mismo poder administrar en el sistema el proceso de gestión de incidencias.

Los perfiles manejados dentro del sistema serán los siguientes:

- **Súper Administrador**, encargado de la Administración y Mantenimiento interno del sistema WF-EC, tanto en configuración de funcionamiento del sistema, como de

mantenimiento de los datos almacenados dentro de la BD de WF-EC.

- **Team Manager**, segundo usuario en nivel de jerarquía dentro del sistema WF-EC, con la capacidad de acceder a todos los módulos y de modificar la información que considere conveniente.
- **Operador de NOC**, usuario con permisos limitados solo a los módulos que gestionan el reconocimiento de alarmas, la asignación y despacho de las ordenes de trabajo, y de resolver la incidencia en conjunto con el técnico de campo.
- **Supervisor de Grupos de Trabajo**, usuario con la potestad de asignar los RRHH dentro de los grupos de trabajo y Hub en cada ciudad, además de monitorear el estado de los técnicos de campo, sus indicadores y poder tomar decisiones en base a estos indicadores.
- **Técnicos de Campo**, usuarios solo con acceso a sus indicadores, resolver las actividades que han sido asignadas dentro de la gestión de cada ticket, y de resolver las incidencias en conjunto con el operador del NOC.
- **Invitados**, usuario que puede revisar de manera general, indicadores y reportes públicos dentro de WF-

EC sin necesidad de credenciales para el acceso al sistema.

### **2.2.8 Persistencia**

Un aspecto importante dentro del sistema es el correcto almacenamiento de la información que se maneje dentro del proceso de gestión de las incidencias. Este módulo manejará la persistencia de dicha información, su generación o exportación en formatos útiles (por ejemplo archivos CSV) para análisis en herramientas externas o de manipulación de datos y también obtención de respaldos de la base de datos bajo petición del usuario con el rol de súper administrador.

### **2.2.9 Administración y Mantenimiento**

Este módulo permitirá gestionar el funcionamiento del sistema WF-EC y las configuraciones necesarias para una adecuada interacción con los usuarios del sistema. Las siguientes son actividades que se definen en este módulo:

- Configuración de direcciones IP de los servidores de Base de Datos
- URLs de conexión a los sistemas externos como el sistema de gestión de Claro.
- Respaldos periódicos de la Base de Datos para que se mantenga la agilidad.
- Mantenimiento de Usuarios (Creación, Modificación, Eliminación) dentro del sistema.

## **CONCLUSIONES Y RECOMENDACIONES**

### **Conclusiones**

Luego de aproximadamente 9 meses de desarrollo y de pruebas continuas con el cliente, la implementación del sistema WorkForce se llevó a cabo obteniendo las siguientes conclusiones (el Anexo 2 muestra la página principal de la aplicación):

1. Se redujo la carga operativa del seguimiento de una incidencia para los operadores de NOCERICSSON.
2. Permite obtener estadísticas de los casos de soporte abiertos por el cliente principal, Claro.
3. Se obtiene indicadores de productividad para los operadores de NOC-ERICSSON y para los técnicos de campo.
4. Se permite realizar el seguimiento de toda incidencia reportada en cualquier momento dado.

5. Se logró integración con otros sistemas, por medio de la exportación de información necesaria para reportes o presentación de indicadores.
6. Alerta a los operadores de NOC-ERICSSON, Técnicos de Campo, Supervisores y Administradores de los vencimientos en los tiempos y criticidad de las incidencias.

### **Recomendaciones**

Después de un tiempo de implementación y de análisis de rendimientos durante 9 meses, por parte del equipo de desarrollo y del cliente, se recomienda:

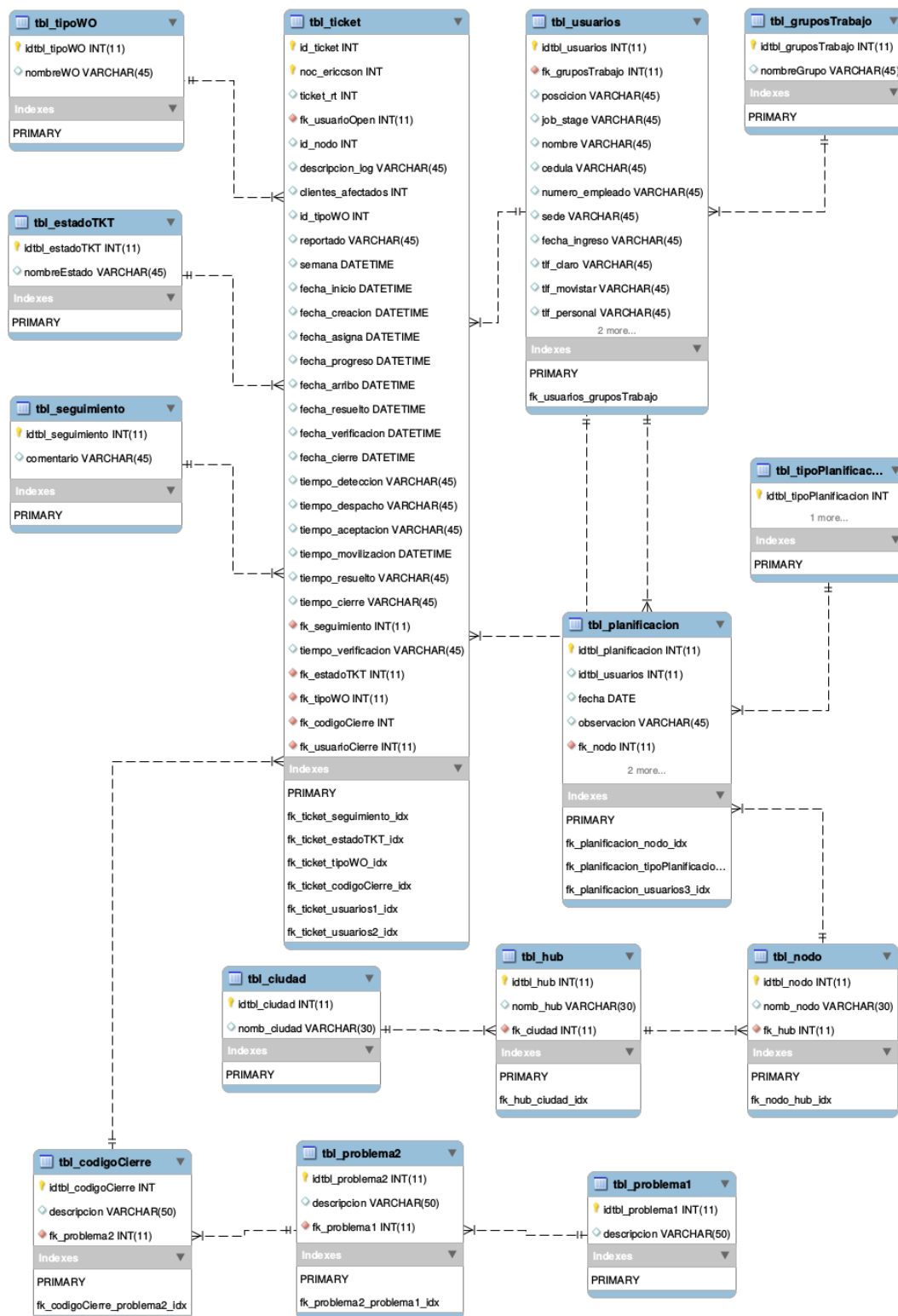
1. Para un mayor número de incidencias mensuales (Superiores a las 5000) se recomienda que algunas plataformas sean mejoradas, por ejemplo el motor de BD debe ser más robusto para gestionar mayor número de transacciones.
2. Si se amplía la red HFC y se requiere gestionar más puntos de la red, es necesario reestructurar la BD para poder cubrir dicha necesidad.
3. Debido a algunas solicitudes del cliente, ciertos módulos no tienen muy buena interacción dentro de dispositivos móviles, debido a la gran cantidad de información presentada en pantalla, lo cual es incómodo de manejar en pantallas de tamaño pequeño, se recomienda restringir el acceso a estos módulos desde dispositivos móviles.

## BIBLIOGRAFÍA

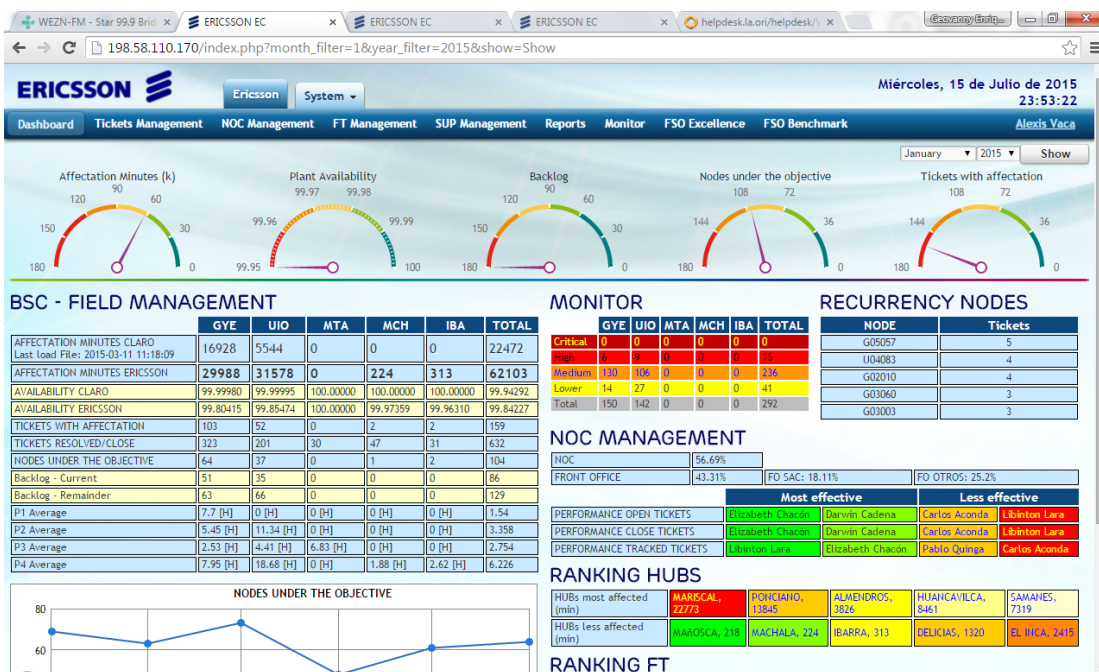
- [1] I. Sommerville, SOFTWARE ENGINEERING, Boston, Massachusetts: Addison-Wesley, 2005.
- [2] A. Garbajosa, Análisis de la Literatura de Gestión de Proyectos en Relación al Análisis de Riesgos en la Producción de Software, Madrid: Control de la Producción Software – Ingeniería en Informática, Universidad Carlos III de Madrid, 2012.
- [3] B. Boehm, A spiral model of software development and enhancement, New York, NY, USA: ACM SIGSOFT Software Engineering Notes, 1986.
- [4] PHP, «PHP sitio oficial,» 2001. [En línea]. Available: <https://secure.php.net/>. [Último acceso: 29 Julio 2015].
- [5] P. Solutions, «Elastix, freedom to communicate,» 2006. [En línea]. Available: <http://www.elastix.org/index.php/en/>. [Último acceso: 29 Julio 2015].
- [6] M. G. G. Ó. P. M. Luis Alberto Casillas Santillán, «Repositorio Universitat de Catalunya,» 21 Septiembre 2012. [En línea]. Available: <http://www.uoc.edu/>. [Último acceso: 29 Julio 2015].
- [7] W. School, «W3C,» 28 Octubre 2014. [En línea]. Available: <http://www.w3.org/html/>. [Último acceso: 29 Julio 2015].
- [8] F. jQuery, «jQuery write less, do more,» 14 Agosto 2014. [En línea]. Available: <https://jquery.org/>. [Último acceso: 29 Julio 2015].



## **ANEXOS**



Anexo 1 Modelo relacional de la BD



Anexo 2 Dashboard del WF-EC