



Facultad de Informática-Matemática  
Universidad de Holguín "Oscar Lucero Moya"

## Trabajo de diploma para optar por el título de Ingeniero en Informática



### Mantenimiento del Sistema de Tramitación de Interrupciones en los Sistemas de Informática y las Comunicaciones Zona Oriente Norte, ECASA s.a.

**Autor:** Leonardo Castro Bauta.

**Tutores:** M.Sc. Leydis Lamoth Borrero.

Ing. Glenis González Cedeño.

**Consultante:** Ing. Efren Cruz Julve.

Holguín, Cuba  
Mayo, 2009



## DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Departamento de Informática de la Universidad de Holguín “Oscar Lucero Moya” para que hagan el uso que estimen pertinente con este trabajo.

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

\_\_\_\_\_  
Leonardo Castro Bauta

\_\_\_\_\_  
MSc. Leydis Lamoth Borrero

\_\_\_\_\_  
Ing. Glenis González Cedeño



## OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado

\_\_\_\_\_, fue realizado en nuestra  
entidad \_\_\_\_\_. Se considera  
que, en correspondencia con los objetivos trazados, el trabajo realizado le  
satisface:  Totalmente

Parcialmente en un \_\_\_\_\_ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los  
beneficios siguientes:

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Como resultado de la implantación de este trabajo se reporta un efecto  
económico que asciende a \_\_\_\_\_ MN y/o \_\_\_\_\_ CUC.

Y para que así conste, se firma la presente a los \_\_\_\_\_ días del mes de  
\_\_\_\_\_ del año \_\_\_\_\_

\_\_\_\_\_

Nombre del representante de la entidad

Cargo

\_\_\_\_\_

Firma

Cuño



## OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

**Título:** Mantenimiento del Sistema de Tramitación de Interrupciones en los Sistemas de Informática y las Comunicaciones Zona Oriente Norte, ECASA s.a.

**Autor:** Leonardo Castro Bauta

Los tutores del presente Trabajo de Diploma consideran que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan.

<Aquí el tutor debe expresar cualitativamente su opinión y medir (usando la escala: muy alta, alta, adecuada) entre otras las cualidades siguientes:

- Independencia
- Originalidad
- Creatividad
- Laboriosidad
- Responsabilidad>

<Además, debe evaluar la calidad científico-técnica del trabajo realizado (resultados y documento) y expresar su opinión sobre el valor de los resultados obtenidos (aplicación y beneficios) >

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de <nota 2-Desaprobado, 3-Aprobado, 4-Bien, 5-Excelente>. <Además, si considera que los resultados poseen valor para ser publicados, debe expresarlo también>

---

MSc. Leydis Lamoth Borrero  
**Tutor**

---

Ing. Glenis González Cedeño  
**Tutor**

Fecha: \_\_\_\_\_



*«La intención sin acción es ilusión.»*

*Paulo Coelho*



## Dedicatoria

---

*A mi mamá a quien le debo lo que soy, por estar siempre a mi lado en la carrera hacia mis metas, por su infinito amor, por ser quien es.*

*A mis abues, a tía Tania y tío José por apoyarme y confiar en mí.*

*A mi tía Reyni y a Yamy, que estando tan lejos, sus pensamientos están aquí conmigo.*

*A todos los que hicieron posible este trabajo.*

*Es para ustedes...*



## Resumen

---

En la actual “Era de la Información”, como algunos expertos han llamado a estos tiempos, se concretiza la necesidad del empleo de herramientas informáticas en la automatización de procesos diarios; fundamentalmente, en el campo de las telecomunicaciones.

La Empresa Cubana de Aeropuertos y Servicios Aeronáuticos ECASA S.A., ha implantado un sistema informático para gestionar de una forma rápida y segura las interrupciones que ocurren en los sistemas de informática y las comunicaciones de los aeropuertos: **Sistema de Tramitación de Interrupciones en los Sistemas de Informática y las Comunicaciones (TI)**. El mismo, desde su puesta en marcha en el 2007 en el Aeropuerto Internacional de Holguín “Frank País”, ha contribuido a cumplimentar los objetivos de esta entidad además de favorecer a la correcta toma de decisiones de sus directivos.

Con la explotación diaria de dicho producto han surgido nuevas necesidades funcionales que aumentarán considerablemente la calidad del mismo.

Con el objetivo de adicionar las funcionalidades surgidas, adaptar el software a las nuevas condiciones y corregir los defectos; se desarrolla esta investigación que pretende realizar un proceso de mantenimiento del software al sistema **TI**, garantizando su calidad, sostenibilidad y la satisfacción del cliente.



## Abstract

---

In the current "Information's Era", like some experts have called to these times, the need of the employment of data processing tools in the automation of daily processes is specified, fundamentally, in the field of the telecommunications.

The Cuban Business of Airports and Aeronautic Services ECASA INC., has established a data processing system to manage of a sure and fast way the interruptions that occur in the informatics and communications systems of the airports: **Interruptions Processing System (TI)**. The same one, since its set in motion in the 2007 in the International Airport of Holguín "Frank País", has contributed to compliment the objectives of the company besides favoring to the correct takes of its executives' decisions.

With the daily exploitation of this product, new functional needs have arisen that will enlarge considerably the quality of the same one.

With the objective to add the functionalities arisen, to adapt the software to new conditions and to correct the defects; it develops this investigation that intends to carry out a software maintenance process to the system (**TI**), guaranteeing its quality, sustainability and the customer satisfaction.

# Índice

Introducción .....	1
<b>Capítulo 1. Fundamentos teóricos .....</b>	<b>7</b>
1.1. INTRODUCCIÓN .....	7
1.2. OBJETO DE ESTUDIO .....	7
La tramitación de las interrupciones en los sistemas de informática y las comunicaciones .....	8
1.2.1. <i>Modelo del negocio actual</i> .....	10
Procesos del negocio .....	10
1.2.2. <i>Actores del negocio</i> .....	11
1.2.3. <i>Trabajadores del negocio</i> .....	11
1.2.4. <i>Diagrama de casos de usos del negocio</i> .....	12
1.2.5. <i>Casos de uso del negocio</i> .....	13
1.2.6. <i>Reglas del negocio a considerar</i> .....	15
1.3. TENDENCIAS Y TECNOLOGÍAS ACTUALES .....	18
1.3.1. <i>Mantenimiento del software (MS)</i> .....	18
1.3.1.1. Técnicas del MS .....	19
1.3.1.2. Tipos de mantenimiento .....	20
1. Correctivo .....	20
2. Adaptativo .....	21
3. Perfectivo .....	22
4. Preventivo .....	23
1.3.1.3. Costos del MS .....	23
Causas del alto costo del MS .....	23
1.3.2. <i>Metodologías de desarrollo de software</i> .....	25
1.3.2.1. RUP .....	26
1.3.3. <i>Modelo cliente-servidor</i> .....	27
1.3.4. <i>Aplicación Web</i> .....	28
1.3.4.1. Lenguajes de programación para la Web .....	28
PHP (Hypertext Preprocessor) .....	29
1.3.4.2. Ajax ( <i>Asynchronous JavaScript and XML</i> ) .....	30
1.3.5. <i>Sistema Gestor de Base de Datos</i> .....	31
PostgreSQL .....	31
1.4. CONCLUSIONES .....	33
<b>Capítulo 2. Descripción y elaboración de la propuesta solución .....</b>	<b>35</b>
2.1. INTRODUCCIÓN .....	35
2.2. MANTENIMIENTO CORRECTIVO .....	35
2.3. MANTENIMIENTO ADAPTATIVO .....	36
2.4. MANTENIMIENTO PREVENTIVO .....	36
2.5. MANTENIMIENTO PERFECTIVO .....	38
2.5.1. <i>Requerimientos funcionales</i> .....	38
2.5.2. <i>Requerimientos no funcionales</i> .....	40
2.5.3. <i>Actores del sistema a automatizar</i> .....	42
2.5.4. <i>Diagrama de Casos de Uso del Sistema a automatizar</i> .....	44
2.5.5. <i>Descripción de los Casos de Uso</i> .....	47
2.5.6. <i>Paquetes y sus relaciones</i> .....	52
2.5.7. <i>Flujo de diseño</i> .....	53
2.5.8. <i>Flujo de implementación</i> .....	55
2.6. PRINCIPIOS DE DISEÑO .....	56
2.6.1. <i>Interfaz de usuario</i> .....	56
2.6.2. <i>Formato de salida de los informes</i> .....	58
2.7. DISEÑO DE LA BASE DE DATOS .....	61
2.8. DIAGRAMA DE DESPLIEGUE .....	62
2.9. VALORACIÓN DE SOSTENIBILIDAD .....	62



Dimensión Administrativa .....	63
Dimensión Socio- Humanista .....	68
Dimensión Ambiental .....	69
Dimensión Tecnológica .....	69
2.10. VALORACIÓN DE LA PROPUESTA.....	70
2.10.1. <i>Entrevista</i> .....	70
2.11. CONCLUSIONES .....	70
<b>Conclusiones Generales .....</b>	<b>72</b>
<b>Recomendaciones.....</b>	<b>73</b>
<b>Bibliografía y Referencias Bibliográficas .....</b>	<b>74</b>
<b>Glosario de términos .....</b>	<b>I</b>
<b>Anexos .....</b>	<b>IV</b>

## Índice de tablas

---

Tabla 1. Descripción de los actores del negocio .....	11
Tabla 2. Descripción de los trabajadores del negocio .....	11
Tabla 3. Descripción textual del caso de uso Calcular Eficacia (confiabilidad) de los sistemas de informática y comunicaciones.....	13
Tabla 4. Descripción textual del caso de uso Calcular Eficacia (disponibilidad) de los sistemas de informática y comunicaciones.....	14
Tabla 5. Definición de actores del sistema a automatizar.....	42
Tabla 6. Descripción del caso de uso del sistema: Crear informe de Eficacia (disponibilidad) General.....	47
Tabla 7. Descripción del caso de uso del sistema: Crear gráfico de tendencia Eficacia (confiabilidad) por Zona.....	48
Tabla 8. Descripción del caso de uso del sistema: Crear informe de Inventario de Equipos Existentes por Zona .....	49
Tabla 9. Descripción del caso de uso del sistema: Crear informe de las Quejas Inconsistentes por Zona .....	50
Tabla 10. Descripción del caso de uso del sistema: Crear informe de los Tiempos Empleados en atender y solucionar las interrupciones por Zona .....	51
Tabla 11. Puntos de función desajustados. ....	64
Tabla 12. Instrucciones fuentes por lenguaje utilizado. ....	65
Tabla 13. Instrucciones fuentes. ....	65

## Índice de figuras

---

Figura 1. Diagrama de casos de uso del negocio .....	12
Figura 2. Origen de defectos del software .....	21
Figura 3. Aumento de los recursos necesarios para el MS.....	23
Figura 4. Costo relativo aproximado de detectar y corregir defectos en las etapas del ciclo de vida del software. ....	25
Figura 5. Arquitectura Cliente-Servidor.....	27
Figura 6. Actores del Sistema.....	43
Figura 7. Diagrama de los casos de uso del sistema, paquete Administración. ....	44
Figura 8. Diagrama de los casos de uso del sistema, paquete Administrativo -> Confiabilidad. ....	45
Figura 9. Diagrama de los casos de uso del sistema, paquete Administrativo -> Disponibilidad. ....	46
Figura 10. Diagrama de Paquetes del Sistema. ....	52
Figura 11. Diagrama del Paquete "Atención de Interrupciones". ....	53
Figura 12. Crear el informe de Eficacia (disponibilidad). ....	57
Figura 13. Crear el informe de Tiempos Empleados en atender y solucionar las interrupciones. ....	58
Figura 14. Informe PDF de Eficacia (confiabilidad). ....	59
Figura 15. Gráfico de Tendencia Eficacia (disponibilidad) por Zona. ....	61
Figura 16. Diagrama de Despliegue. ....	62



## Introducción

---

El hombre de nuestros días, a consecuencia de las constantes transformaciones en la que se encuentra inmersa la sociedad actual, ha puesto su mayor empeño en el eficaz tratamiento de la información; contando para ello con el avance vertiginoso de las ciencias informáticas en los diversos sectores de la economía mundial.

Como parte de la estrategia económica aplicada por nuestro país a partir de la década de los años 90, a raíz del derrumbe del campo socialista en Europa y sus nefastas consecuencias, se asume el turismo internacional como un sector clave en vías de alcanzar un desarrollo sostenible.

“La Aviación Civil es un participante vital en el turismo y componente fundamental de la sociedad actual; ha pasado a ser parte integral de la economía, catalizadora de los intercambios culturales internacionales. En toda la historia moderna ningún otro medio ha contribuido tanto al movimiento de individuos y mercancías en todas partes del mundo. Los aeropuertos son elementos de los que no se puede prescindir, eslabones vitales de la cadena de producción del servicio aéreo, por él transitan las personas y los bienes que hacen funcionar a los destinos turísticos” [ 5 ]

Con el transcurso del tiempo se han incrementado notablemente los arribos al Aeropuerto Internacional “Frank País García” de la provincia de Holguín, terminal inaugurada en julio de 1996 por el Comandante en Jefe Fidel Castro Ruz como motivo del crecimiento anual del turismo en esta zona, por esta razón se hace necesario gestionar las interrupciones de los sistemas de informática y las comunicaciones existentes en el aeropuerto en períodos de tiempo ínfimos con el objetivo de solucionarlas y evitar que afecten la calidad de los servicios aeronáuticos. Para lograrlo se definió el Procedimiento Específico PE-2806-02 Revisión 1.0 basado en la Norma ISO 9001.2000, aplicada a la tramitación de las interrupciones y al cálculo de la disponibilidad y confiabilidad de los sistemas de comunicaciones.



A raíz del notable incremento de la informatización de los procesos diarios de la vida laboral, la dirección de la ECASA se interesó en informatizar los principales procesos que dan apoyo al correcto funcionamiento de los servicios aeronáuticos, entre estos se encuentra la Gestión de las Interrupciones en los Sistemas de Informática y Comunicaciones (**GISIC**). Esta acción contribuiría indiscutiblemente a mejorar la calidad y eficacia en el trabajo.

Para materializar este interés, en el año 2007, se realizó un trabajo de diploma para optar por el título de Ingeniero Informático; en el que se llevó a cabo un trabajo previo de informatización, donde se logró realizar un estudio sobre el flujo de la información concerniente al proceso de **GISIC**.

Como resultado del trabajo antes mencionado se obtuvo el **Sistema de Tramitación de Interrupciones en los Sistemas de Informática y Comunicaciones**(a partir de ahora **TI**), sistema Web Cliente-Servidor el cual gestiona las interrupciones que ocurren en los sistemas de informática y las comunicaciones en la Zona Oriente Norte de la ECASA, el mismo está basado en el uso de tecnologías de software libre en su totalidad.

Mediante entrevistas realizadas al cliente, se pudo determinar que dicho sistema, el cual se encuentra en explotación en el Aeropuerto “Frank País García” de la Zona Oriente Norte de la ECASA, carece de funcionalidades como:

- Métodos de seguridad.
- Gráficos de tendencias.
- Indicador técnico de calidad: Eficacia.
- Informes que apoyan la correcta toma de decisiones relacionados con los sistemas informáticos.

Y contiene como deficiencias:

- No se tuvieron en cuenta algunos de los parámetros de configuración del sistema.
- Detalles en algunas funcionalidades del sistema que afectan su rendimiento.



Unido al surgimiento de otros requisitos funcionales, y la falta de funcionalidades en el sistema que apoyen al proceso de toma de decisiones por parte de los administrativos de la ECASA; hacen que el sistema implantado no tenga la calidad que se requiere en la actualidad.

La **situación problemática** de esta investigación está dada por la inexistencia de un conjunto de funcionalidades y deficiencias presentes en el actual producto informático, lo cual atenta contra la calidad del proceso de **GISIC** en la Zona Oriente Norte de la ECASA.

Dando paso al **problema de la investigación**: ¿Cómo favorecer el proceso de **GISIC** en la Zona Oriente Norte de la ECASA, llevado a cabo en el sistema informático **TI**?

Dicho problema se enmarca en el **objeto de estudio**: El Proceso de Gestión de Interrupciones en los Sistemas de Informática y las Comunicaciones en la Zona Oriente Norte de la ECASA.

Por lo que se define como **objetivo**: Realizar el proceso de mantenimiento del software<sup>1</sup> en el sistema informático **TI**, para favorecer el proceso de **GISIC** en la Zona Oriente Norte de la ECASA.

El objetivo delimita el **campo de acción**: El proceso de informatización de la **GISIC** en la Zona Oriente Norte de la ECASA.

Para guiar la investigación se plantea la **hipótesis**:

El proceso de mantenimiento del software, sobre la base de la corrección de errores, adaptación a nuevas condiciones y extensión de funcionalidades, permitirá brindar, al sistema informático **TI**, mejores alternativas de seguridad y eficacia que favorecerán al proceso de **GISIC** en la Zona Oriente Norte de la ECASA.

---

<sup>1</sup> Mantenimiento del software: La modificación de un producto informático después de su entrega al cliente o usuario para corregir defectos, para mejorar el rendimiento u otras propiedades deseables, o para adaptarlo a un cambio de entorno.



Continuando el uso de las tecnologías de software libre para las cuales fue concebido el sistema informático **TI**, se utilizó el lenguaje de programación Web PHP y Ajax como técnica de desarrollo web para la creación de páginas dinámicas. Como sistema gestor de base de datos el PostgreSQL 8.0; el mismo es uno de los más potentes y avanzados en la actualidad.

Para la modelación del negocio y los nuevos requisitos funcionales se utilizó la metodología Proceso Unificado de Desarrollo de Software (RUP, por sus siglas en inglés), haciendo práctica del Lenguaje Unificado de Modelación (UML, por sus siglas en inglés). La Herramienta para Asistir por Computadora a la Ingeniería del Software (CASE, por sus siglas en inglés): Rational Rose sirvió de apoyo en la confección de los documentos y gráficos requeridos por la metodología RUP.

Para cumplimentar el objetivo de esta investigación se realizaron las siguientes **tareas**:

1. Diagnosticar el estado del proceso para la Gestión de Interrupciones en los Sistemas de Informática y las Comunicaciones en la Zona Oriente Norte de la ECASA.
2. Realizar la fundamentación teórica del proceso de mantenimiento del software y de las nuevas tendencias en el campo de la Programación Web basada en el uso de software libre.
3. Capturar los nuevos requisitos funcionales.
4. Valorar la sostenibilidad y factibilidad del producto informático.
5. Desarrollo de los nuevos requisitos funcionales que se desprenden del proceso de mantenimiento del sistema informático.
6. Implantación del sistema informático resultante.
7. Evaluación de la satisfacción de los usuarios con respecto al producto informático obtenido.

En la realización de estas tareas los métodos empleados fueron:

- Métodos empíricos:
  - La entrevista: permitió conocer cómo se maneja la información en el proceso de **GISIC** ayudando indiscutiblemente en la



determinación de los nuevos requisitos funcionales. Además se empleó para evaluar el grado de aceptación, por parte del usuario, de la calidad del sistema posterior al proceso de mantenimiento del mismo.

- La revisión de documentos: facilitó la obtención de conocimientos importantes en la tramitación de las interrupciones así como en el estudio de las nuevas funcionalidades implementadas.
- Métodos teóricos:
  - Análisis y síntesis: se empleó para el estudio y diagnóstico de la situación existente referente al proceso de **GISIC**, permitiendo determinar las características fundamentales del mismo y profundizar en este proceso que es el objeto de estudio de la investigación.
  - Histórico-lógico: permitió recopilar toda la información relacionada con los procesos de gestión de interrupciones. Además creó las bases para llevar a cabo el mantenimiento del sistema (**TI**) que se propuso como solución a la situación problemática.
  - Hipotético-Deductivo: se utilizó en la elaboración de la hipótesis y su posterior evaluación.
  - Modelación: para la construcción abstracta de la realidad, la cual es de vital importancia en esta investigación, ya que permitió una mayor comprensión de los procesos que se realizan en el sistema y los nuevos requisitos funcionales que se integran al mismo.
- Métodos estadísticos y matemáticos:
  - Método de los mínimos cuadrados: se usó para graficar las líneas de tendencia lineal.

Para mostrar el transcurso del proceso de mantenimiento del sistema en cuestión y los resultados obtenidos, la documentación de esta investigación se ha estructurado en dos capítulos:

En el Capítulo 1: se exponen las bases teóricas que sustentan la investigación, siguiéndole la descripción de las principales herramientas que se usan para la elaboración de la propuesta de solución y la metodología de software utilizada en la investigación.



En el Capítulo 2: se resume la elaboración de solución propuesta, modelando los nuevos requisitos funcionales y describiendo los cambios y adaptaciones realizadas en los existentes. Se encuentra también, como uno de sus elementos más importantes, la Valoración de Sostenibilidad del producto informático transformado resultante del proceso de mantenimiento del software.



# Capítulo 1. Fundamentos teóricos

---

## 1.1. Introducción

---

En este capítulo se lleva a cabo un análisis del objeto de estudio de la investigación para nutrirnos del cómo se desarrollan los procesos del negocio del sistema de gestión de interrupciones en los sistemas de informática y las comunicaciones. Posteriormente se hace una caracterización de los diferentes tipos de mantenimiento del software que se realizaron así como las herramientas usadas en la elaboración de la propuesta de solución.

## 1.2. Objeto de estudio

---

“La Empresa Cubana de Aeropuertos y Servicios Aeronáuticos, ECASA S.A., constituye una organización subsidiaria de la denominada Corporación de la Aviación Cubana S.A. (CACSA), y como tal garantiza con todos sus activos los financiamientos que reciba dicha compañía y asume por su cuenta y en forma subsidiaria el pago de los mismos” [ 4 ]

“ECASA se encarga de la atención a aeronaves, garantiza el funcionamiento del Tránsito Aéreo, la Informática, Comunicaciones, Navegación y Meteorología. Lleva el control del tránsito aéreo en el territorio nacional y el enmarcado por los organismos internacionales de la aviación civil; así como el aprovisionamiento de combustible, lubricantes y líquidos especiales a las aeronaves. Además brinda los servicios de Seguridad y Protección Aeroportuaria, provee servicios de Handling en cualquiera de sus modalidades, entre otras actividades fundamentales” [ 5 ]

La misión de los Servicios Aeronáuticos es: Aplicar, Coordinar y Desarrollar las normas, procedimientos y técnicas de las Dependencias de Comunicaciones, Control de Tránsito Aéreo, Centro de Mensaje y ARO/AIS/MET (Información Aeronáutica y Meteorología) para garantizar la seguridad de las operaciones de los servicios aéreos de forma regular, eficiente y económica.



La Zona Aeroportuaria Oriente Norte es una Unidad Empresarial de Base, parte integrante de la ECASA. Esta compuesta por 5 aeropuertos, de ellos 2 internacionales, los que a continuación se menciona:

- Aeropuerto Internacional “Frank País García”. Holguín.
- Aeropuerto “Orestes Acosta”. Moa.
- Aeropuerto “Carlos Manuel de Céspedes”. Bayamo.
- Aeropuerto Internacional “Sierra Maestra”. Manzanillo.
- Aeropuerto “Eddy Caballero”. Nicaro.

La Zona Oriente Norte realiza la función metodológica y de control técnico hacia los aeropuertos Orestes Acosta de Moa, Carlos Manuel de Céspedes en Bayamo y Sierra Maestra en Manzanillo, en las diferentes especialidades enmarcadas dentro de los Servicios Aeronáuticos.

### **La tramitación de las interrupciones en los sistemas de informática y las comunicaciones**

En toda empresa actual, para lograr su correcto funcionamiento, se tiene muy en cuenta el papel esencial que juegan la tramitación y la gestión de la información que esta maneja. Con el avance tecnológico de estos tiempos, se ha incrementado indiscutiblemente el uso de las nuevas tecnologías en el desarrollado mundo de la computación, la electrónica y las comunicaciones; brindando infinitas facilidades en los entornos empresariales.

A consecuencia de la gran significación que tienen los sistemas de informática y las comunicaciones en los aeropuertos, la tramitación de las interrupciones que ocurren en estos es de vital importancia para el restablecimiento de los servicios aeronáuticos.

La interrupción en un sistema de comunicación se define como “cualquier acontecimiento inesperado que pueda dar lugar a un período operacionalmente importante, durante el cual una instalación, equipo o sistema no facilite servicio dentro de las tolerancias especificadas” [ 11 ]



Glenis González Cedeño afirma que...” La tramitación a las interrupciones de los sistemas de comunicaciones desempeña un papel fundamental, sobre todo si se habla de gestionarlas y atenderlas de forma correcta en el momento adecuado para solucionar las averías y restablecer los servicios. Es precisamente este detalle el que puede marcar la diferencia con la competencia ya que existe una estrecha relación entre la tramitación, la eficiencia y el éxito” [ 5 ]

Para garantizar el correcto funcionamiento de los aeropuertos del país, se necesita conocer el comportamiento de las diferentes instalaciones, equipos o sistemas que en ellos existen; entre los aspectos característicos del funcionamiento de los mismos se encuentran los indicadores técnicos:

- Disponibilidad
- Confiabilidad

La disponibilidad de una estación se define como: “La relación entre el tiempo real de funcionamiento y el tiempo de funcionamiento especificado de una determinada estación, equipo o sistema” [ 11 ]

La confiabilidad de una estación se define como: “La probabilidad de que una instalación terrestre, equipo o sistema funcione dentro de las tolerancias especificadas en un período de tiempo determinado” [ 11 ]

El cálculo de estos indicadores está definido en el Procedimiento de Calidad Específico basado en la Norma ISO 9001.2000 en el contexto de la gestión de la calidad de la Dirección de los Servicios Aeronáuticos de la Empresa Cubana de Aeropuertos y Servicios Aeronáuticos (ECASA).

Resultado de esta investigación surge otro indicador técnico, que aunque no está definido en el Procedimiento de Calidad antes mencionado, ayuda a la correcta toma de decisiones de los directivos de los aeropuertos:

- Eficacia

El mismo resume los índices de disponibilidad y de confiabilidad. Para lograr la conformidad de este indicador, el 100% de las instalaciones, equipos o sistemas deberán obtener índices (antes mencionados) superiores a un valor porcentual previamente establecido.



### 1.2.1. Modelo del negocio actual

En los trabajos para optar por el Título de Ingeniero Informático realizados por Oscar G. Reyes Pupo y María del Pilar Pérez Céspedes se realizó un estudio profundo acerca de los procesos que se realizan en la Unidad Territorial de Servicios Aeronáuticos Zona Oriente Norte de la ECASA, describiéndose de forma general las actividades que se llevan a cabo en este lugar.

Basándonos en estos trabajos se ha decidido describir y modelar solo aquellos procesos que se encuentran relacionados con el cálculo de la eficacia de los sistemas de informática y las comunicaciones, llevados a cabo en la Unidad Territorial de Servicios Aeronáuticos Zona Oriente Norte de la ECASA.

La descripción y modelamiento de los procesos antes mencionados se realizó a través de la metodología RUP y los artefactos que brinda el lenguaje de modelado UML.

#### Procesos del negocio

“Los procesos de negocio son un grupo de actividades lógicamente relacionadas que se llevan a cabo para dar soporte al funcionamiento del mismo, empleando sus recursos para obtener resultados en correspondencia con su misión” [ 8 ]

Los principales procesos del negocio que dan soporte a la gestión de interrupciones de los sistemas de informática y las comunicaciones, en la subdirección de Servicios Aeronáuticos en la Zona Oriente Norte de la ECASA son:

- Gestionar Interrupción.
- Procesar la información obtenida del:
  - Cálculo de la Disponibilidad de los sistemas de informática y las comunicaciones.
  - Cálculo de la Confiabilidad de los sistemas de informática y las comunicaciones.
  - Cálculo de la Eficacia (disponibilidad) de los sistemas de informática y las comunicaciones



- Cálculo de la Eficacia (confiabilidad) de los sistemas de informática y las comunicaciones
  - Enviar Parte Diario de la Zona

### 1.2.2. Actores del negocio

“En los procesos de negocio intervienen partes, que se benefician de los resultados obtenidos de los procesos, denominados actores del negocio” [ 8 ]. En la **Tabla 1**, se muestran los actores del negocio que se relacionan con los procesos del negocio mencionados anteriormente.

**Tabla 1. Descripción de los actores del negocio**

Nombre del actor	Descripción
Cliente	Representa a los clientes que formulan quejas al producirse alguna interrupción o avería en los sistemas de informática y las comunicaciones.  Da inicio al proceso: Gestionar Interrupción.
Supervisor de Comunicaciones de la UET Habana	Representa al Supervisor de comunicaciones de la Unidad Empresarial Territorial (UET) que radica en la Habana.  Se relaciona con el proceso: Enviar Parte Diario, Calcular Disponibilidad de los sistemas de informática y las comunicaciones, Calcular Confiabilidad de los sistemas de informática y las comunicaciones, Calcular Eficacia (confiabilidad) de los sistemas de informática y las comunicaciones, Calcular Eficacia (disponibilidad) de los sistemas de informática y las comunicaciones

### 1.2.3. Trabajadores del negocio

“También en los procesos de negocio intervienen trabajadores y otras partes de la Empresa que participan directamente en la realización de estos procesos, denominados trabajadores del negocio” [ 8 ]. En la **Tabla 2** que se muestra a continuación, se presentan los trabajadores relacionados con los procesos antes mencionados.

**Tabla 2. Descripción de los trabajadores del negocio**

Nombre del trabajador	Descripción
-----------------------	-------------

Supervisor	Representa al responsable del área de comunicaciones que gestiona las interrupciones y los reportes.
Técnico	Representa a los encargados de solucionar las interrupciones o averías de los sistemas de informática y las comunicaciones para restablecer los servicios afectados.

### 1.2.4. Diagrama de casos de usos del negocio

“Un diagrama de casos de uso del negocio es un diagrama que representa gráficamente a los procesos del negocio y su interacción con los actores del negocio “[ 8 ]. Esta interacción se muestra como sigue:

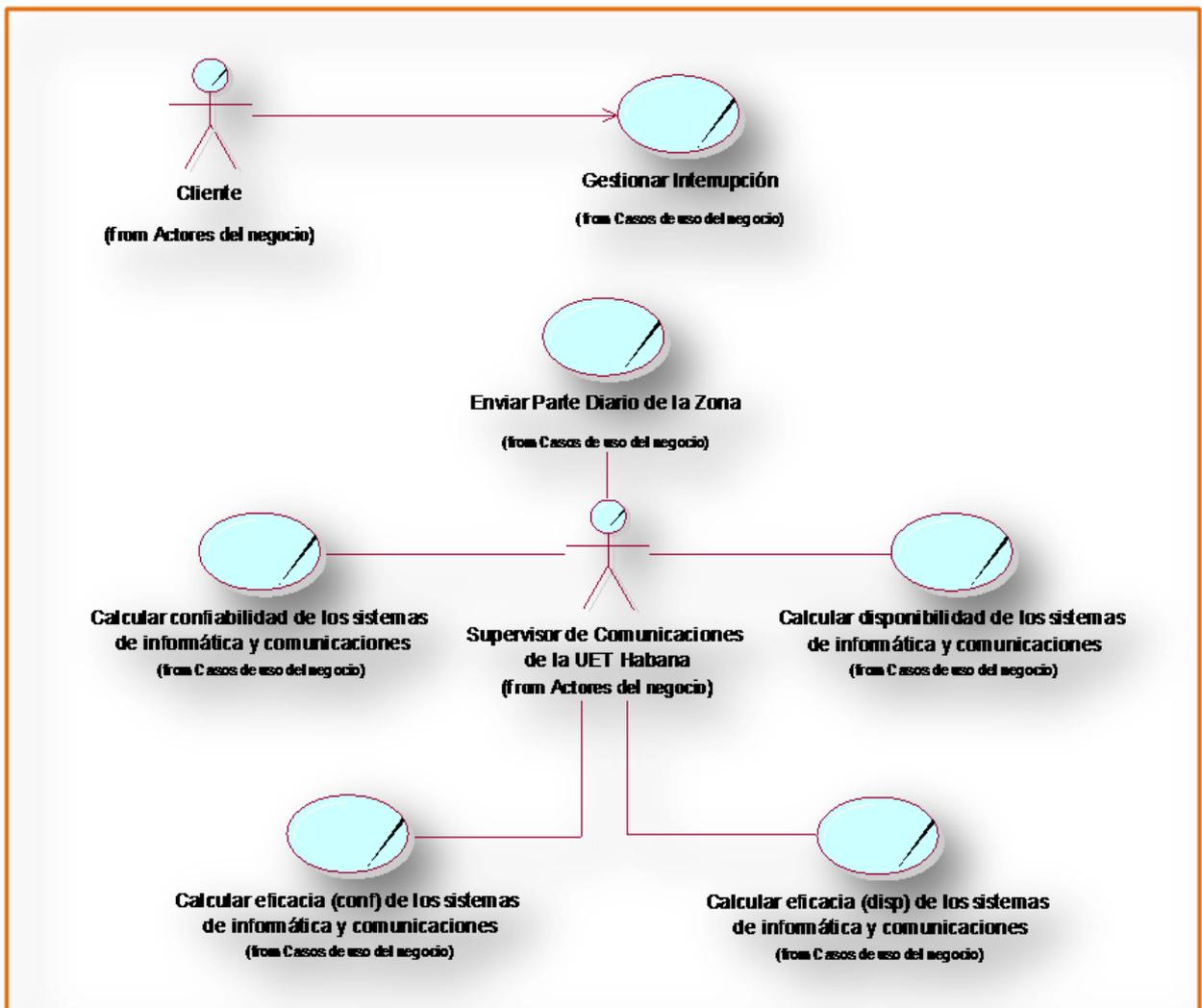


Figura 1. Diagrama de casos de uso del negocio



### 1.2.5. Casos de uso del negocio

“Un caso de uso especifica una secuencia de acciones, que producen un resultado observable de valor para un actor concreto” [ 8 ]. Los diagramas de actividades pueden ser consultados en el **Anexo 1**.

**Tabla 3. Descripción textual del caso de uso Calcular Eficacia (confiabilidad) de los sistemas de informática y comunicaciones.**

<b>Caso de uso del negocio:</b>	Calcular Eficacia (confiabilidad) de los sistemas de informática y las comunicaciones
<b>Actores del negocio:</b>	-
<b>Propósito:</b>	Calcular la Eficacia (confiabilidad) de los sistemas de informática y las comunicaciones en un periodo determinado.
<b>Resumen:</b>	
El caso de uso se inicia cuando el Supervisor necesita calcular la Eficacia del indicador técnico: Confiabilidad que han tenido los diferentes sistemas de informática y las comunicaciones de los aeropuertos en su Zona durante todo un trimestre. El caso de uso finaliza cuando el Supervisor realiza el informe de Eficacia (confiabilidad) y es enviado a las instancias superiores.	
<b>Caso de uso asociado:</b>	-
<b>Flujo de eventos:</b>	
<b>Acción del actor :</b>	<b>Respuesta del negocio:</b>
5- El Supervisor de Comunicaciones de la UET Habana recibe el informe y	1-El Supervisor analiza la carpeta de reporte la cual contiene almacenada los reportes que ocurrieron en todo un trimestre. 2- El supervisor calcula por cada aeropuerto y Zona a la que pertenece, la eficacia del indicador técnico: Confiabilidad, escogiendo aquellos reportes que afectaron o afectan los servicios. 3- El Supervisor confecciona el informe de eficacia (confiabilidad) de los sistemas. 4-El Supervisor envía el informe al Supervisor de Comunicaciones de la UET Habana.



culmina el caso de uso.	
<b>Prioridad:</b>	Alta. Permite calcular la eficacia (confiabilidad) de los sistemas de informática y las comunicaciones.
<b>Mejoras:</b>	Permite una mayor eficiencia y rapidez en el cálculo de la eficacia (confiabilidad) y confección del informe, así como reduce los errores en los cálculos.
<b>Cursos alternos:</b>	-

**Tabla 4. Descripción textual del caso de uso Calcular Eficacia (disponibilidad) de los sistemas de informática y comunicaciones.**

<b>Caso de uso del negocio:</b>	Calcular Eficacia (disponibilidad) de los sistemas de informática y las comunicaciones
<b>Actores del negocio:</b>	-
<b>Propósito:</b>	Calcular la Eficacia (disponibilidad) de los sistemas de informática y las comunicaciones de un periodo determinado.
<b>Resumen:</b>	
El caso de uso se inicia cuando el Supervisor necesita calcular la Eficacia del indicador técnico: Disponibilidad que han tenido los diferentes sistemas de informática y comunicaciones de los aeropuertos en su Zona durante todo un trimestre. El caso de uso finaliza cuando el Supervisor realiza el informe de Eficacia (disponibilidad) y es enviado a las instancias superiores.	
<b>Caso de uso asociado:</b>	-
<b>Flujo de eventos:</b>	
<b>Acción del actor :</b>	<b>Respuesta del negocio:</b>
	<p>1-El Supervisor analiza la carpeta de reporte la cual contiene almacenada los reportes que ocurrieron en todo un trimestre.</p> <p>2- El supervisor calcula por cada aeropuerto y Zona a la que pertenece, la eficacia del indicador técnico: Disponibilidad, escogiendo aquellos reportes que afectaron o afectan los servicios.</p> <p>3- El Supervisor confecciona el informe de</p>



<p>5- El Supervisor de Comunicaciones de la UET Habana recibe el informe y culmina el caso de uso.</p>	<p>eficacia (disponibilidad) de los sistemas.                  4-El Supervisor envía el informe al Supervisor de Comunicaciones de la UET Habana.</p>
<p><b>Prioridad:</b></p>	<p>Alta. Permite calcular la eficacia (disponibilidad) de los sistemas de informática y las comunicaciones.</p>
<p><b>Mejoras:</b></p>	<p>Permite una mayor eficiencia y rapidez en el cálculo de la eficacia (disponibilidad) y confección del informe, así como reduce los errores en los cálculos.</p>
<p><b>Cursos alternos:</b></p>	<p>-</p>

### 1.2.6. Reglas del negocio a considerar

A continuación se mencionan las reglas del negocio que rigen y velan por el correcto funcionamiento del negocio que caracteriza el proceso en cuestión, relacionando solo las que tienen relación con los procesos del negocio planteados.

Un servicio interrumpido representa a cualquier interrupción o avería que se presente en cualquiera de los sistemas de comunicaciones atendidos por la especialidad o relacionados con dependencias externas que soportan la actividad de comunicaciones en los Servicios Aeronáuticos.

En cada zona de Servicios Aeronáuticos o grupo de trabajo del Nivel Central existirá un responsable del área de comunicaciones (llámese técnico de guardia, supervisor o jefe de turno) que tomará los reportes de las interrupciones de los servicios reclamados por los clientes.

Los reportes se realizarán sobre todos los sistemas de informática y las comunicaciones con el fin de atender con calidad, eficiencia y rapidez cualquier sistema o equipo de informática/comunicaciones ante una avería o interrupción. Cada servicio interrumpido provoca la apertura de un reporte que se asentará sobre un registro de reportes.



La evaluación de los reportes se llevará a cabo por la parte técnica si el reporte abierto es de carácter interno o externo, siendo los de carácter interno todos aquellos reportes que son atendidos directamente por la Zona o grupo de trabajo y los externos todos aquellos reportes que deben ser atendidos por otra zona o grupo de trabajo o por una dependencia externa a ECASA.

Los reportes de carácter interno constituidos en el servicio a restablecer o equipo a reparar pasan a ser automáticamente continuados por la parte técnica relacionada con el mismo, sea el propio técnico de guardia, el personal técnico del taller, brigadas móviles.

Si el reporte es de carácter externo, se contactará con la dependencia o entidad encargada de reparar este servicio.

Una vez reparado el servicio afectado o restablecido el servicio interrumpido, las dependencias antes mencionadas notificarán al supervisor de comunicaciones que el servicio interrumpido se encuentra restablecido y estarán disponibles para el chequeo del servicio en cuestión.

Una vez resuelta la interrupción, el reporte debe cerrarse en consulta con quien lo generó y una vez recibida la aprobación del cliente, se procede a cerrar el reporte. Cerrado el reporte se procede a completar todos los datos inherentes a este en el Registro de Reportes.

Un reporte está integrado por un identificador, el supervisor que lo abrió, el supervisor que lo cierra, la hora de apertura, la hora en que es cerrado, el tipo (interno, externo), el estado del reporte (en proceso, restablecido, pendiente a compra, servicio restablecido), Si afecta servicio o no y una nota técnica.

Una queja esta formada por un identificador, el subsistema interrumpido o averiado, el aeropuerto en el que se ha generado la queja, quien la emitió, la fecha y la descripción de la misma.

Los tipos de reportes son internos y externos. El estado de un reporte es abierto y cerrado. Cada técnico tiene asignado un grupo de subsistemas a atender en varios aeropuertos y un subsistema puede ser atendido por varios técnicos. Los técnicos, supervisores y jefes de grupo también pueden formular quejas.



Un reporte solo es cerrado por el supervisor y puede cerrarse aún si el cliente no envía notificación de conformidad. Una vez cerrado el reporte no se puede modificar. El técnico es el único que puede modificar el estado de un reporte. Un reporte puede tener asociadas varias quejas.

Todas las zonas o grupos de trabajo deben emitir un parte diario al Supervisor de Comunicaciones de la Zona Occidental, relacionando en dicho parte las interrupciones más significativas ocurridas en el día y las interrupciones pendientes de solución en la zona, especificando si, según su consideración afectan o no a los servicios aeronáuticos.

La disponibilidad de una estación se calcula mediante la relación entre el tiempo real de funcionamiento y el tiempo de funcionamiento especificado de dicha estación.

Disponibilidad = Tiempo de funcionamiento real \* 100 / Tiempo de funcionamiento requerido.

La *confiabilidad* de una estación es la probabilidad de que una instalación terrestre, equipo o sistema funcione dentro de las tolerancias especificadas en un período de tiempo determinado y se calcula de la siguiente forma:

$$\text{Confiabilidad} = 100 * \text{EXP} (-t/m)$$

Donde: t = 24 horas para analizar la confiabilidad de que un equipo trabaje sin fallos en un día,

m= MTBF (tiempo medio entre fallas de una instalación, equipo o sistema)

La *eficacia* es un indicador técnico que resume el comportamiento de los índices de confiabilidad y disponibilidad de una instalación, equipo o sistema, en un período de tiempo determinado. Para lograr la conformidad de dicho indicador, el 100% de las instalaciones deberán obtener índices (antes mencionados) superiores a un valor porcentual específico.

Resumiendo el proceso de gestión de las interrupciones: Cuando se produce una interrupción en un sistema de informática o de comunicación el cliente se lo



notifica al responsable del área de comunicaciones, el cual elaborará un reporte técnico especificando si el reporte abierto es de carácter interno o externo. Los reportes de carácter interno pasan a ser automáticamente continuados por la parte técnica relacionada con el mismo. Si el reporte es de carácter externo, se contactará con la dependencia o entidad encargada de reparar este servicio. Una vez reparado el servicio afectado o restablecido el servicio interrumpido el técnico notificará al supervisor de guardia el restablecimiento del servicio afectado, el cual contactará con quienes generaron la queja para conocer el grado de satisfacción y conformidad con la solución. Una vez recibida la aprobación del cliente, se procede a cerrar el reporte el cual pasará a un registro de reportes existentes.

### **1.3. Tendencias y tecnologías actuales**

---

Para alcanzar un buen desarrollo del software, se analizaron con detenimiento las características y tendencias de las tecnologías actuales existentes en el mundo.

#### **1.3.1. Mantenimiento del software (MS)**

---

El software no se deteriora con el uso ni con el paso del tiempo, a diferencia de los materiales mecánicos. Mejor dicho, no sufre de un deterioro físico. No obstante, se suele considerar que el software tiene un deterioro en su estructura, cuando a lo largo del tiempo se van incluyendo más y más cambios que hacen que su estructura interna sea cada vez más difícil de entender.

En 1970 ya se había popularizado el término “Crisis del Software” para referir a la situación que acabamos de describir. Los síntomas de esta crisis han estado repercutiendo desde entonces en la industria de desarrollo de software y todavía se sienten sus efectos. Para resolver el problema surgió un área de la informática que recibe el nombre de Ingeniería del Software.

La definición posiblemente más utilizada de la Ingeniería del Software es la siguiente: “La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, la operación y el mantenimiento del software; esto es, la aplicación de la ingeniería al software.” [ 12 ]



El proceso de desarrollo del software se intenta estructurar en diversas etapas, a esta descomposición se le conoce como Ciclo de Vida del Software. Las tareas de mantenimiento son las últimas en realizarse en dicho ciclo. Al ser la actividad de mantenimiento la última en el ciclo, no quiere decir que es la menos importante dentro de este.

Una de las definiciones de “Mantenimiento del Software” (en lo adelante **MS**) es: “El mantenimiento del software es la modificación de un producto software después de la entrega para corregir fallos, para mejorar el rendimiento u otros atributos, o para adaptar el producto a un entorno modificado”. [ 12 ]

Todo software evoluciona para adaptarse a las necesidades de sus usuarios y como todo cambio tiene como característica fundamental el hecho de que primero se necesita una comprensión del objeto que se ha de cambiar, para poder hacer efectiva la modificación; resulta de gran importancia prever la mantenibilidad (es decir, la “facilidad de mantenimiento”), aún antes de la entrega del producto. De esta forma se puede implementar un cambio con un menor esfuerzo que en un sistema que es menos mantenible.

### **1.3.1.1. Técnicas del MS**

Dentro de la ingeniería del software se proporcionan soluciones técnicas que permiten abordar el mantenimiento de manera que su impacto en costo dentro del ciclo de vida sea menor. Las soluciones técnicas pueden ser de tres tipos:

1. La Ingeniería directa: Desarrollo del software tradicional.
2. Ingeniería inversa: Análisis de un sistema para identificar sus componentes y las relaciones entre ellos, así como para crear representaciones del sistema en otra forma o en un nivel de abstracción más elevado.
3. Reestructuración del software: Cambio de representación de un producto software, pero dentro del mismo nivel de abstracción

El objetivo de estas técnicas es proporcionar métodos para reconstruir el software, ya sea reprogramándolo, redocumentándolo, rediseñándolo, o rehaciendo alguna(s) característica(s) del producto.



Cualquiera de estas técnicas se puede aplicar a lo largo de todas las fases del ciclo de vida o bien entre algunas de sus fases.

También existen otras tecnologías como:

- La remodelarización: consiste en cambiar la estructura modular de un sistema de forma que se obtenga una nueva estructura.
- Análisis de la facilidad de mantenimiento: normalmente la mayor parte del mantenimiento se centra relativamente en unos pocos módulos del sistema.
- Análisis y mediciones: son importantes tecnologías que estudian ciertas propiedades en el rendimiento de los programas.

### **1.3.1.2. Tipos de mantenimiento**

“La fase de mantenimiento se centra en el cambio que va asociado a la corrección de errores, a las adaptaciones requeridas a medida que evoluciona el entorno del software y a cambios debidos a las mejoras producidas por los requisitos cambiantes del cliente. Por lo tanto luego de haberse realizado la fase de mantenimiento se puede obtener una nueva versión del software”. [ 14 ]

Existen cuatro tipos reconocidos de operaciones de mantenimiento, los cuales están en función del objetivo particular para el cual son puestos en marcha, y en función a los recursos utilizados, así tenemos:

#### **1. Correctivo**

Este mantenimiento también es denominado “mantenimiento reactivo”, tiene lugar luego que ocurre una falla o avería, es decir, solo actuará cuando se presenta un error en el sistema.

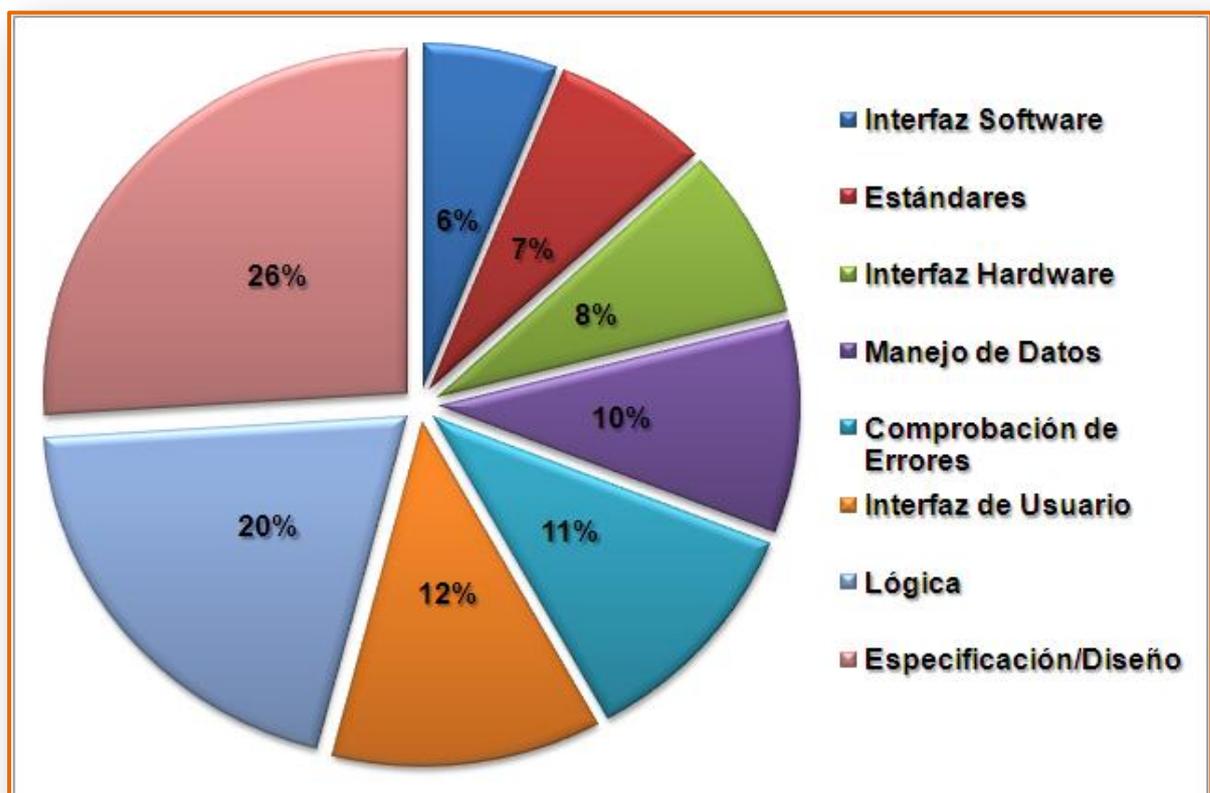
El mantenimiento correctivo, como su nombre lo indica, cambia el software para corregir los defectos. Su objetivo es localizar y eliminar los problemas o defectos de los programas, incluso luego de aplicar las pruebas en fases anteriores a este proceso.

“Se entiende como defecto de un programa, a una característica del programa capaz de causar un fallo. Un fallo ocurre cuando el comportamiento de un programa es diferente del establecido en la especificación” [ 16 ]

Los fallos en el software pueden estar dados por:

- Procesamiento: salidas incorrectas de un programa.
- Rendimiento: tiempo de respuesta demasiado alto.
- Programación: inconsistencia en el diseño de un programa.
- Documentación: inconsistencias entre la funcionalidad de un programa y el manual de usuario.

Un estudio estadístico de los orígenes de defectos en el software se muestra en la figura siguiente [16]:



**Figura 2. Origen de defectos del software**

## 2. Adaptativo

Este tipo de mantenimiento consiste en la modificación de un programa debido a cambios en el entorno (hardware o software) en el cual se ejecuta.

Los cambios pueden afectar a:

- el sistema operativo (cambio a uno más moderno)



- la arquitectura física del sistema informático (paso de una arquitectura de red de área local a Internet/Intranet)
- al entorno de desarrollo del software (incorporación de nuevos elementos o herramientas)

La envergadura del cambio necesario puede ser muy diferente: desde un pequeño retoque en la estructura de un módulo hasta tener que reescribir prácticamente todo el programa para su ejecución en un ambiente distribuido en una red.

Los cambios en el entorno software pueden ser de dos clases:

- Datos: por ejemplo, al dejar de trabajar con un sistema de ficheros clásico y sustituirlo por un sistema de gestión de bases de datos relacionales.
- Procesos: por ejemplo, migrando a una nueva plataforma de desarrollo.

Este tipo de mantenimiento es cada vez más frecuente debido principalmente al cambio, cada vez más rápido, en los diversos aspectos de la informática: nuevas generaciones de hardware, nuevos sistemas operativos o versiones de los antiguos, y mejoras en otros elementos del sistema.

### 3. Perfectivo

Cambios en la especificación, normalmente debidos a cambios en los requerimientos de un producto software, implican un nuevo tipo de mantenimiento llamado perfectivo. La casuística es muy variada. Desde algo tan simple como cambiar el formato de impresión de un informe, hasta la incorporación de un nuevo módulo funcional. Podemos definir el mantenimiento perfectivo como el conjunto de actividades para mejorar o añadir nuevas funcionalidades requeridas por el usuario.

Algunos autores dividen este tipo de mantenimiento en dos:

- Mantenimiento de Ampliación: orientado a la incorporación de nuevas funcionalidades.
- Mantenimiento de Eficiencia: que busca la mejora de la eficiencia de ejecución.

## 4. Preventivo

Este último tipo de mantenimiento consiste en la modificación del software para mejorar las propiedades de dicho software (por ejemplo, aumentando su calidad y/o su mantenibilidad) sin alterar sus especificaciones funcionales.

Algunas maneras de hacerlo son:

- incluir sentencias que comprueben la validez de los datos de entrada
- reestructurar los programas para mejorar su legibilidad
- incluir nuevos comentarios que faciliten la posterior comprensión del programa.

### 1.3.1.3. Costos del MS

Múltiples estudios señalan que el mantenimiento es la parte más costosa del ciclo de vida del software. Estadísticamente está comprobado que el coste de mantenimiento de un producto software a lo largo de toda su vida útil supone más del doble que los costes de su desarrollo. La tendencia es creciente con el paso del tiempo. En general, el porcentaje de recursos necesarios para mantenimiento se incrementa a medida que se produce más software [ 16 ]:

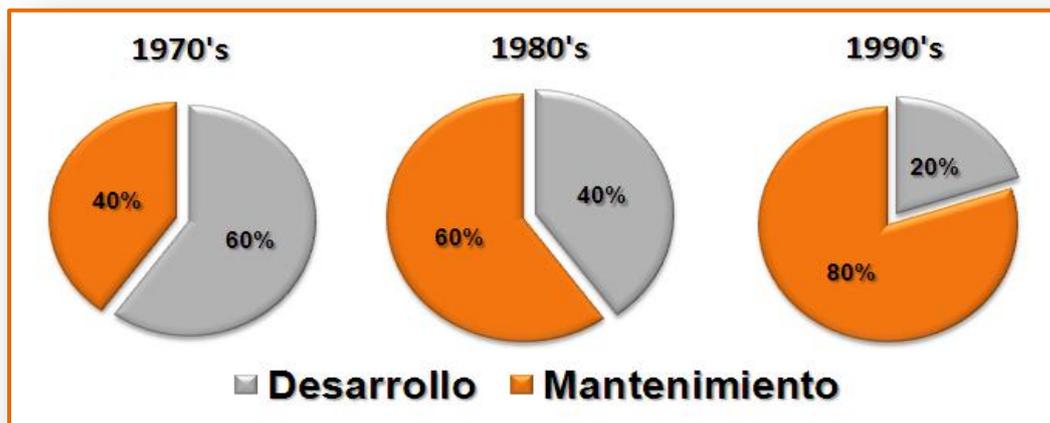


Figura 3. Aumento de los recursos necesarios para el MS.

### Causas del alto costo del MS

Son varias las causas de que en la mayoría de las organizaciones actuales se requiera mucho trabajo de mantenimiento, entre algunas podemos citar a:



- Una gran cantidad del software que existe actualmente ha sido desarrollado hace más de 10 años. Aunque estos programas fuesen creados utilizando las mejores técnicas de diseño y codificación existentes en su momento, se construyeron con restricciones de tamaño y espacio de almacenamiento y se desarrollaron con herramientas tecnológicamente desfasadas.
- Estos programas han sufrido una o varias migraciones a nuevas plataformas o sistemas operativos.
- Han experimentado múltiples modificaciones para mejorarlos y adaptarlos a las nuevas necesidades de los usuarios.
- Todos estos cambios se realizaron sin tener en cuenta la arquitectura general del sistema (no se aplicaron técnicas de ingeniería inversa o reingeniería).

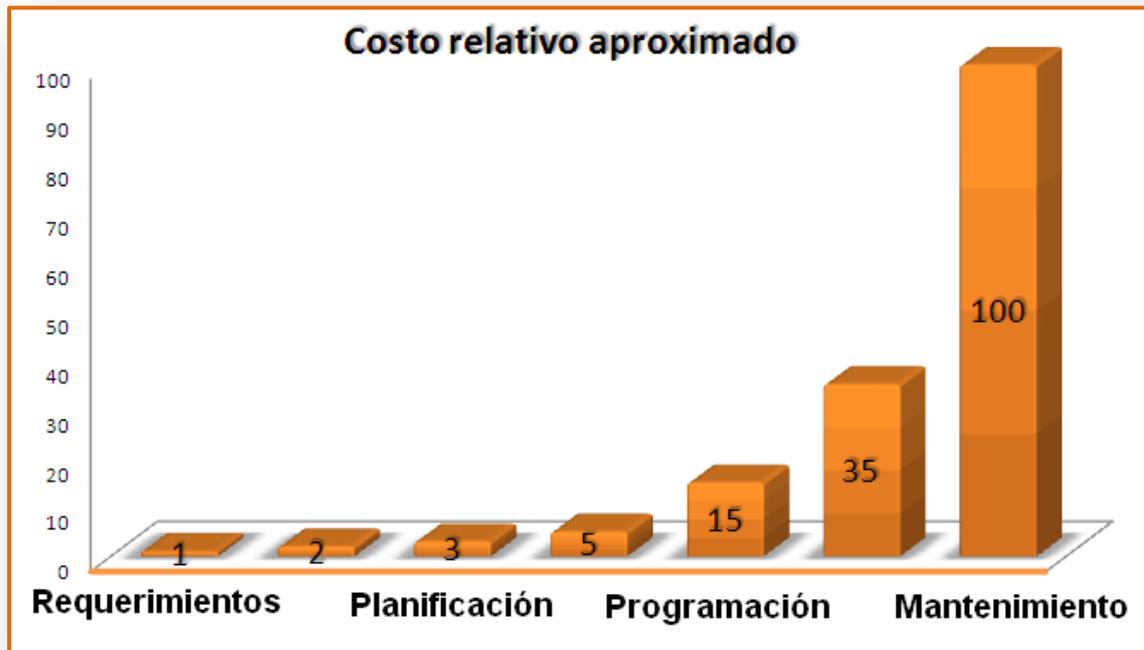
El resultado de lo antes expuesto, son los sistemas informáticos con baja calidad, los mismos se caracterizan generalmente por: diseño pobre de las estructuras de datos, mala codificación, lógica defectuosa, y documentación escasa.

Otra causa directa de los grandes costos del **MS** es que el costo relativo de reparar un defecto aumenta considerablemente en las últimas etapas del ciclo de vida del software, de forma que la relación entre el costo de detectar y reparar un defecto en la fase de análisis de requisitos y en la fase de mantenimiento es de 1 a 100 respectivamente (ver **figura 4**). [ 16 ]

Algunas de las razones por las que es menos costoso detectar y corregir un error durante las etapas iniciales del ciclo de vida que durante las etapas últimas son:

- Es más fácil cambiar la documentación (por ejemplo, los documentos de especificación o de diseño) que modificar el código.
- Un cambio durante una fase tardía puede requerir que sea modificada la documentación de todas las fases anteriores.

- Es más fácil encontrar un defecto durante la fase en la cual se ha introducido el defecto que tratar de detectar y corregir los efectos provocados por el defecto en una fase posterior.
- La causa de un defecto puede esconderse en la inexistencia o falta de actualización de los documentos de especificación o diseño.



**Figura 4. Costo relativo aproximado de detectar y corregir defectos en las etapas del ciclo de vida del software.**

### **1.3.2. Metodologías de desarrollo de software**

Una de las claves para el éxito final de un software lo proporciona la forma en que este es concebido. Las metodologías de desarrollo de software, definidas como: “un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de Sistemas Informáticos” [ 13 ]. Las mismas brindan estrategias de desarrollo centradas en las personas o los equipos, orientadas hacia la funcionalidad y la entrega, de comunicación intensiva y que requieren implicación directa del cliente.



Existen varias metodologías de desarrollo de software que pueden ser aplicadas en dependencia del sistema que se desea desarrollar. Entre las más sobresalientes se encuentran las que proponen los procesos XP (siglas en inglés de Programación Extrema) y RUP (por sus siglas en inglés Rational Unified Process). La primera puede ser aplicada eficazmente en el desarrollo de pequeños sistemas, además es conocida como una metodología ágil por sus escasos flujos de trabajo y la poca documentación generada. RUP en cambio, puede ser utilizado para la construcción de sistemas de cualquier tamaño, generando una extensa documentación, que bien administrada puede ser de gran utilidad al desarrollador.

### 1.3.2.1. RUP

Es un proceso de desarrollo iterativo e incremental, en el sentido de que el software no es liberado de una sola vez al final del proyecto, sino que es desarrollado por partes. Cada iteración contiene todas las fases del ciclo de vida de un software: análisis, diseño, implementación y prueba.

RUP es una propuesta de proceso para el desarrollo de software orientado a objetos que utiliza UML<sup>2</sup> (su traducción al español Lenguaje Unificado de Modelación) para describir todo el proceso. Está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas.

Sus características principales son:

- Guiado/Manejado por casos de uso.
- Centrado en la arquitectura.
- Iterativo e Incremental.
- Desarrollo basado en componentes.
- Utilización de un único lenguaje de modelación (UML).

---

<sup>2</sup> Es un lenguaje de especificación, visualización, construcción y documentación de propósito general, aunque especializado en sistemas software. No define un proceso concreto que determine las fases de un sistema, las empresas pueden utilizarlo como lenguaje para definir sus propios procesos y lo único que tendrán en común con otras organizaciones que lo utilicen serán los tipos de diagramas.

- Proceso Integrado.

La metodología seleccionada para el desarrollo de la presente investigación es la descrita por el Proceso Unificado de Rational, el cual se ha detallado anteriormente. Esta selección se basa en las propias características de RUP y las facilidades que éste aporta a todo el proceso de desarrollo. Además, RUP como producto, viene acompañado de una herramienta muy útil que soporta cada uno de sus flujos de trabajo: Rational Rose, la cuál también será empleada.

### 1.3.3. Modelo cliente-servidor

El modelo Cliente-Servidor es un sistema en el que el cliente establece una conexión con el servidor para recurrir a los servicios que ofrece. Es posible que aplicaciones de servidor sirvan de forma concurrente a varios clientes. La **arquitectura cliente-servidor** (ver **figura 5**), como también se le conoce a este modelo, es una forma de dividir y especializar programas y equipos de cómputo a fin de que la tarea que cada uno de ellos realiza se efectúe con la mayor eficiencia, y permita simplificar las actualizaciones y mantenimiento del sistema.

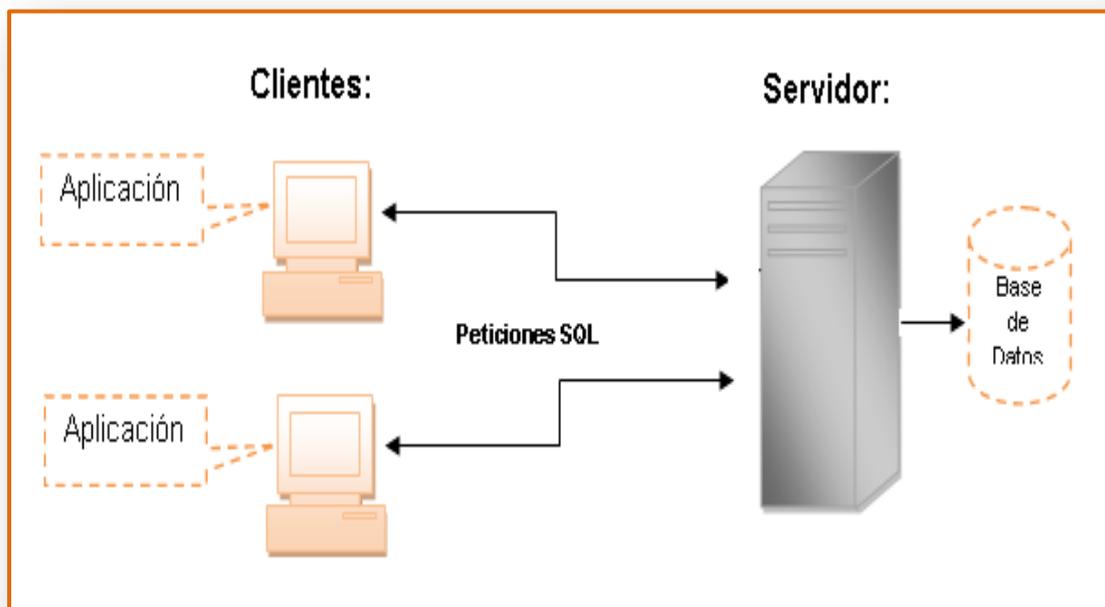


Figura 5. Arquitectura Cliente-Servidor



---

### 1.3.4. Aplicación Web

---

Una aplicación Web es un conjunto de páginas Web enlazadas que visualizan la información que se quiere mostrar a través de ella. Constituye una de las mejores herramientas para divulgar, gestionar y compartir la información por lo que trae consigo un aumento de la eficiencia en cuanto a la manipulación de gran cantidad de elementos.

Las aplicaciones Web se desarrollan como una extensión de los sistemas Web para agregar funcionalidad de negocio al proceso; o sea, estas permiten a los usuarios ejecutar lógica de negocio a través de un Navegador, es decir, modificar el estado del negocio.

“Una de las ventajas de las aplicaciones Web es que el mantenimiento del sistema se concentra en el servidor, reduciendo así el gasto. En general, es el proveedor del servicio quien se preocupa de tener la aplicación siempre disponible y actualizada. Se podrían encontrar numerosas ventajas más, pero probablemente la propiedad más destacada sea la conectividad que proporciona a Internet, permitiendo el acceso a la aplicación desde cualquier punto. Esto permite ahorrarse el invertir en costosas infraestructuras de comunicaciones que en muchos casos podrían ser sencillamente no viables” [ 15 ].

#### 1.3.4.1. Lenguajes de programación para la Web

---

Uno de los ejes fundamentales que diferencian a Internet de otros medios de comunicación es la interacción y personalización de la información con el usuario. Esto se logra por medio de alguno de los diferentes lenguajes de programación para Web que existen hoy en día, dichos lenguajes se clasifican en dos partes fundamentales que reconocen la propia arquitectura cliente/servidor de esta plataforma de desarrollo: los lenguajes del lado del servidor y los lenguajes del lado del cliente.

“Entre los lenguajes del lado del servidor se pueden encontrar entre los más sobresalientes al *Practical Extracting and Reporting Language* (PERL), *Active Server Pages* (ASP), *Personal Home Page* (PHP), *Java Server Pages* (JSP). Estos se caracterizan por desarrollar la lógica de negocio dentro del servidor, además de ser los encargados del acceso a bases de datos, tratamiento de la



información, etc. Del lado del cliente se encuentran principalmente el *JavaScript* y el *Visual Basic Script*, que son los encargados de aportar dinamismo a la aplicación en los navegadores” [ 2 ]

A continuación se describen las ventajas que conlleva utilizar el PHP en el desarrollo de un sistema informático.

## **PHP (Hypertext Preprocessor)**

PHP es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. Es también un lenguaje interpretado y embebido en el HTML.

Fue creado originalmente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. Actualmente PHP se encuentra en su versión 5 que utiliza el motor Zend Engine desarrollado con mayor meditación para cubrir las necesidades de las aplicaciones Web actuales.

El PHP corre en 7 plataformas, funciona en 11 tipos de servidores, ofrece soporte sobre unas 20 Bases de Datos y contiene unas 40 extensiones estables sin contar las que se están experimentando, sin contar que:

- Es software libre y abierto, lo que implica menos costes y servidores más baratos que otras alternativas.
- Es muy rápido. Su integración con la base de datos PostgreSQL y el servidor Apache, le permite constituirse como una de las alternativas más atractivas del mercado.
- Su sintaxis está inspirada en el lenguaje de programación C, ligeramente modificada para adaptarlo al entorno en el que trabaja, de modo que si se está familiarizado con esta sintaxis, resultará muy fácil aprender PHP.
- Su librería estándar es realmente amplia, lo que permite reducir los llamados "costes ocultos", uno de los principales defectos de ASP.
- PHP tiene una de las comunidades más grandes en Internet, por lo que no es complicado encontrar ayuda, documentación, artículos, noticias y más recursos.



Se puede agregar que posee una potente variedad de extensiones para el acceso a la mayoría de los sistemas de gestión de bases de datos, por lo que una migración a otro sistema de gestión es mucho menos costosa que en otras plataformas. [ 2 ]

#### **1.3.4.2. AJAX (Asynchronous JavaScript and XML)**

**AJAX**, (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante *XMLHttpRequest*, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

Además es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores, dado que está basado en estándares abiertos como JavaScript.

Unas de las razones que hacen de AJAX una técnica popular son [ 18 ]:

- Basado en los estándares abiertos.
- Usabilidad
- Válido en cualquier plataforma y navegador
- Beneficia las aplicaciones Web.
- No es difícil su utilización
- Constituye una de las características que debe poseer una página Web para clasificarse en Web 2.0.
- Es independiente del tipo de tecnología de servidor que se utilice.
- Mejora la estética de la Web.



### 1.3.5. Sistema Gestor de Base de Datos

Un Sistema de Gestión de Bases de Datos (**SGBD**) puede definirse como un paquete generalizado de software que se ejecuta en un sistema computacional anfitrión, centralizando los accesos a los datos y actuando de interfaz entre los datos físicos y el usuario. Las principales funciones que debe cumplir un **SGBD** se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad de los datos. Los **SGBD** permiten al programador convencional ahorrarse horas de trabajo dedicadas a la seguridad, gestión de los datos, chequeo de errores,...

Para comunicarse con un **SGBD**, tanto para definir datos y estructuras como para hacer consultas sobre los datos, se puede utilizar **SQL** (**S**tructured **Q**uery **L**anguage), que no es más que un lenguaje de consultas estructurado compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en grupos de instrucciones (consultas) para actualizar y manipular las bases de datos [ 10 ].

Entre los **SGBD** más utilizados en el mundo están: Oracle, MySQL, Microsoft SQL Server, PostgreSQL, Interbase, entre otros. Todos presentan un enfoque relacional con un basamento matemático centrado en el Álgebra Relacional [17 ]

#### PostgreSQL

Se diseñó como un **SGBD** objeto relacional. Esto significa, que las tablas se manejan como objetos y las tuplas o registros son instancias de ese objeto. Se pueden crear nuevos tipos de datos y hacer herencias entre objetos. PostgreSQL es una herramienta muy potente para los desarrolladores de sistemas de bases de datos, tiene todo de lo que carece MySQL.

El mismo tiene transacciones, integridad referencial, vistas y multitud de funcionalidades, pero es un poco más lento y pesado que MySQL, aunque en las últimas versiones del mismo esto ha mejorado mucho. Con la aparición de nuevas versiones, los desarrolladores de PostgreSQL argumentaron que empezaba una nueva era: más rápido, más fiable.



### Características Operacionales:

- Transacciones.
- Disparadores.
- Restricciones.
- Replicación.
- Salva y Recuperación.
- Reglas.
- Procedimientos Almacenados/Funciones.
- Integridad Referencial.
- Uniones externas (*Outer Joins*).
- Sintaxis ANSI SQL 89, 92 y 99.
- Trazas.
- Extensivo y programable.
- Orientado a Objetos.
- Características sofisticadas de integridad de datos.
- Tipos de datos y funciones definidos por el usuario.
- Cliente/servidor.

Entre otros.

### Límites de una base de datos en PostgreSQL:

- Máximo tamaño de una base de datos: ilimitado, sólo limitado por la capacidad de almacenamiento del *hardware*.
- Máximo tamaño de una tabla: hasta 64 TB. (*terabytes*).
- Máximo tamaño de un campo: 1Gb.



- Máxima cantidad de tuplas o registros: ilimitado.
- Máxima cantidad de columnas en una tabla: hasta 1600.
- Máxima cantidad de índices por tabla: ilimitado.

Requisitos mínimos para su instalación:

- Memoria principal: 8 MB.
- Espacio libre en disco: 100 MB.
- Se puede descargar para la mayoría de los sistemas operativos existentes o se puede descargar su código fuente y recompilarlo.
- Se necesita del protocolo TCP/IP (incluido en todos los sistemas operativos) para el acceso al mismo desde la red.

Debido a que el presente trabajo continúa el desarrollo de una aplicación ya implementada y que el **SGBD** empleado en ella fue el PostgreSQL, sigue esta siendo la herramienta empleada, además de que los argumentos antes expuestos y el análisis de las características fundamentales de los **SGBD** que se pueden utilizar demuestra que la infraestructura de la aplicación implementada puede ser soportada por el mismo y que las ventajas que nos brindará su utilización son amplias.

## 1.4. Conclusiones

Al concluir este capítulo se ha podido apreciar el estudio realizado de los procesos que intervienen en la gestión de las interrupciones en los sistemas informáticos y las comunicaciones en la Zona Oriente Norte de la ECASA, logrando acercar al lector a las principales actividades del negocio para comprender mejor su funcionamiento.

Se realizó además un estudio importante del **Mantenimiento del Software**, sus técnicas, costos y las causas que provocan estos últimos. Los diferentes tipos de **MS** analizados nos guían en el desarrollo de los nuevos requerimientos funcionales que se añadieron al sistema informático.



Luego de haber analizado las tendencias y tecnologías actuales se reafirmó el uso de PHP y Ajax para el desarrollo de los nuevos requerimientos de la aplicación Web, y el PostgreSQL 8.0 como sistema gestor de base de datos. Se optó por utilizar la metodología RUP ya que nos brinda una serie de facilidades para efectuar el mantenimiento del software en cuestión.



## Capítulo 2. Descripción y elaboración de la propuesta solución

---

### 2.1. Introducción

---

En este capítulo se hace referencia a los requisitos funcionales surgidos de las necesidades del cliente que dieron lugar al mantenimiento del sistema **TI**, así como el diseño y la modelación de los mismos enmarcados en el mantenimiento perfectivo realizado, para ello se utiliza la metodología RUP y los artefactos que el lenguaje UML proporciona. Se describen también, los requisitos no funcionales y se valora la sostenibilidad del producto resultante en las dimensiones: ambiental, socio-humanista, administrativa y tecnológica.

### 2.2. Mantenimiento Correctivo

---

A continuación se describen las principales correcciones que se le realizaron al sistema **TI** y los errores detectados, como parte de las acciones propuestas por el mantenimiento correctivo; cuyo objetivo es cambiar el software para corregir los defectos.

- En las páginas donde se maneja la información concerniente a las contraseñas se validó para que estas tuvieran como mínimo 8 caracteres ya que los estándares de seguridad así lo reglamentan. Anteriormente podían tener cualquier longitud.
- Se incluyeron nuevos parámetros a configurar, los mismos no se tuvieron en cuenta y serían necesarios al generalizar la aplicación:
  - La dirección de la página inicial para acceder al sitio.
  - La dirección de correo electrónico predeterminada a través de la cual se envían los informes.
  - El porcentaje a comparar en los informes de Eficacia.
  - El directorio donde se guardará el log de trazas.

Estos parámetros son configurados por el administrador del sistema.



- Se corrigió la integridad referencial de las relaciones existentes entre las tablas de la base de datos del sistema **TI**, evitando que se borrara información importante para la toma de decisiones de los directivos.

## 2.3. Mantenimiento Adaptativo

---

Es muy probable que cambie el entorno original de un software con el paso del tiempo, por ejemplo el sistema operativo o las características externas de productos para el que se desarrolló el software.

Al extender la aplicación a la gestión de las interrupciones en los sistemas informáticos fue necesario realizar varias adaptaciones al sistema **TI**. A continuación se describen algunas de ellas:

- La necesidad de manipular información de las nuevas entidades: tipo y tipo informática en el sistema **TI**, conllevó a que se creara en la Base de Datos las tablas correspondientes, sus relaciones con las demás tablas, y se tuvieron que modificar aquellos casos de usos que estaban relacionados.
- Al surgir los nuevos requisitos funcionales: Crear informe de los reportes de Software y Redes (General y por Zona), fue necesario modificar la tabla de la base de datos que almacena los reportes, así como adaptar los casos de usos que permiten editar los reportes.
- Al módulo administrativo se le aplicó un prototipo de interfaz diferente a los otros módulos debido a la inclusión de un gran número de nuevas funcionalidades al mismo, los vínculos a estas últimas se agruparon para lograr una mejor visualización.

## 2.4. Mantenimiento Preventivo

---

La mantenibilidad de un producto informático no es más que la propiedad que lo hace más factible a la hora de aplicarle un proceso de mantenimiento posterior a su entrega, disminuyendo así el costo de este proceso.



En aras de lograr un producto software más mantenible se llevaron a cabo algunas de las actividades que propone el mantenimiento preventivo, las mismas se describen como sigue:

- Ayuda del sistema

El **TI** consta con una ayuda que describe como hacer uso de todas las funcionalidades que brinda el sistema informático, a esta se le agregaron las especificaciones de las nuevas funcionalidades introducidas como resultado del mantenimiento.

- Elementos de Diseño

En el diseño se respetaron los estilos que el **TI** consta desde sus inicios como: la interfaz de usuario y los colores que emplea, la letra usada en todos los textos, los estilos CSS, la librería JavaScript de componentes DOJO,...

- Codificación

Se estima que en el proceso de mantenimiento llevado a cabo se reutilizó el código en un 70%, se continuó el uso del desarrollo modular como patrón de diseño de programación, también se comentaron las nuevas funciones de mayor complejidad y se usaron los estándares de código utilizados en el sistema **TI** original:

- **textfield\_** prefijo para nombrar los componentes edits.
- **textarea\_** prefijo para nombrar los componentes textareas.
- **button\_** prefijo para nombrar los botones.
- **select\_** prefijo para nombrar los componentes list / menú
- **tb\_** prefijo para nombrar las tablas de las base de datos.
- **t\_** prefijo para nombrar los tipos de datos definidos en la base de datos.
- **pg\_** prefijo para nombrar las funciones creadas en PostgreSQL.

Todas las actividades antes expuestas hacen que la estructura interna del software en cuestión sea más fácil de entender, incluso luego de haber añadido un sinnúmero de cambios producto a la inclusión de las nuevas funciones.



## 2.5. Mantenimiento Perfectivo

“Entre más se utilice el software por los usuarios estos podrán descubrir funciones adicionales que producirán beneficios. El mantenimiento perfectivo lleva al software más allá de sus requisitos funcionales originales, es el conjunto de actividades para mejorar o añadir nuevas funcionalidades en el sistema” [ 13 ]

A continuación se describen y modelan los nuevos requerimientos que se incluyeron en el sistema informático **TI**.

### 2.5.1. Requerimientos funcionales

“La ingeniería de requisitos facilita el mecanismo apropiado para comprender lo que quiere el cliente , analizando necesidades , confirmando su viabilidad, negociando una solución razonable, especificando la solución sin ambigüedad, validado la especificación y gestionando los requisitos para que se trasformen en un sistema operacional.” [13 ]

Los nuevos requerimientos, también conocidos como requisitos, están agrupados por categorías y se organizaron en subconjuntos para lograr un mejor análisis, clasificándolos en base a las necesidades del cliente. Estos se exponen como sigue:

- **Administrativo:**

- Disponibilidad

- R1) Crear informe de Eficacia (disponibilidad) por Zona.
- R2) Crear informe de Eficacia (disponibilidad) General.
- R3) Crear gráfico de Eficacia (disponibilidad) por Zona.
- R4) Crear gráfico de Eficacia (disponibilidad) General.
- R5) Crear informe PDF de Eficacia (disponibilidad).
- R6) Enviar correo de informe de Eficacia (disponibilidad).
- R7) Crear gráfico de tendencia Eficacia (disponibilidad) por Zona.
- R8) Crear gráfico de tendencia Eficacia (disponibilidad) General.
- R9) Crear gráfico de tendencia disponibilidad por Zona.
- R10) Crear gráfico de tendencia disponibilidad General.



- Confiabilidad

- R11) Crear informe de Eficacia (confiabilidad) por Zona.
- R12) Crear informe de Eficacia (confiabilidad) General.
- R13) Crear gráfico de Eficacia (confiabilidad) por Zona.
- R14) Crear gráfico de Eficacia (confiabilidad) General.
- R15) Crear informe PDF de Eficacia (confiabilidad).
- R16) Enviar correo de informe de Eficacia (confiabilidad).
- R17) Crear gráfico de tendencia Eficacia (confiabilidad) por Zona.
- R18) Crear gráfico de tendencia Eficacia (confiabilidad) General.
- R19) Crear gráfico de tendencia confiabilidad por Zona.
- R20) Crear gráfico de tendencia confiabilidad General.

- Reportes

- R21) Crear informe PDF de reportes abiertos.
- R22) Crear informe PDF de reportes cerrados.
- R23) Crear informe de Inventario de Equipos Existentes por Zona.
- R24) Crear informe de Inventario de Equipos Existentes General.
- R25) Crear informe PDF de Inventario de Equipos Existentes.
- R26) Crear informe de las Quejas Inconsistentes General.
- R27) Crear informe de las Quejas Inconsistentes por Zona.
- R28) Crear informe PDF de las Quejas Inconsistentes.
- R29) Crear informe de los Tiempos Empleados en atender y solucionar las interrupciones General.
- R30) Crear informe de los Tiempos Empleados en atender y solucionar las interrupciones por Zona.
- R31) Crear informe PDF de los Tiempos Empleados en atender y solucionar las interrupciones.
- R32) Crear informe de los reportes de Software General.
- R33) Crear informe de los reportes de Software por Zona.
- R34) Crear informe PDF de los reportes de Software.
- R35) Enviar correo de informe de Software.
- R36) Crear informe de los reportes de Redes General.
- R37) Crear informe de los reportes de Redes por Zona.
- R38) Crear informe PDF de los reportes de Redes.
- R39) Enviar correo de informe de Redes.



- **Administración:**
  - R40) Actualizar parámetros de configuración del sistema.
- **Seguridad:**
  - R41) Insertar traza de operaciones en el fichero: log.txt.

---

## 2.5.2. Requerimientos no funcionales

---

“Son propiedades o cualidades que el producto debe tener. Estas propiedades son las características que hacen al producto atractivo, usable, rápido o confiable, En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto. Describen atributos del sistema o del ambiente del sistema.” [ 8 ]

A continuación se describen los requerimientos no funcionales que caracterizan al sistema informático **TI** tras la inclusión de los nuevos requisitos funcionales:

- **Apariencia o interfaz externa**

El diseño de aplicación se encuentra orientado a simular el entorno de trabajo del cliente para que el usuario se sienta cómodo e identificado con la aplicación.

Se ha propuesto crear una interfaz flexible y amigable para todos los usuarios, los mismos podrán acceder a la información sin entrenamiento previo.

- **Usabilidad**

El sistema debe estar disponible las 24 horas del día y debe ser accesible desde todos los puntos de la ECASA que cuenten con conexión a la red.

- **Rendimiento**

El rendimiento del sistema deberá tener un tiempo de respuesta corto en el procesamiento de la información para proporcionar a los usuarios la información en el momento oportuno, es decir, cuando ellos la requieran.

- **Portabilidad**

El sistema se encuentra implementado con el lenguaje de programación Web PHP y el sistema gestor de base de datos PostgreSQL, por lo que puede ser usado en diferentes plataformas, ya que dichos software son multiplataforma.

- **Seguridad**



Restringir el acceso de los usuarios a las funcionalidades de acuerdo a los roles jugados por los actores del sistema.

El sistema garantizará que los usuarios puedan acceder a la información que le es permitida de acuerdo a los roles definidos para cada uno, es decir el nivel de acceso debe ser restringido. La información debe ser actualizada por el personal autorizado.

Identificar el usuario antes de ejecutar cualquier acción y verificar que tiene permiso para llevarla a cabo.

El sistema debe contar con métodos de seguridad ante la pérdida de la información causada por ruptura del servidor u otros accidentes. Además de contar con datos históricos sobre las operaciones realizadas por los usuarios en el sistema.

El nuevo requisito funcional generado por los requisitos no funcionales de seguridad es **R41**).

- **Confiabilidad**

La información almacenada en el sistema deberá ser estrictamente confiable para evitar que la almacenada sobre las quejas formuladas, así como los reportes asociados posea errores, ya que esto puede alterar el cálculo de los indicadores técnicos de los sistemas de informática y las comunicaciones ocasionando serios problemas.

- **Ayuda y documentación en línea**

Se incluirá en la ayuda del sistema documentación acerca de cómo utilizar el sistema para facilitar su usabilidad.

- **Software**

- **Cliente:**

Se debe disponer de un navegador Web instalado en las máquinas clientes de la aplicación, con soporte para JavaScript y que implemente las características necesarias para el uso de la técnica AJAX, se recomienda el Internet Explorer, Mozilla Firefox o Opera.



Como una de las funcionalidades del sistema consiste en generar informes en formato PDF, se necesita tener instalado un visor de documentos portátiles (PDF) en la máquina cliente.

La resolución de pantalla debe ser 800x600.

- **Servidor:**

El servidor debe contar con un sistema gestor de base de datos PostgreSQL con la versión 8.0, servidor Web Apache con una versión mayor o igual a la 2.2.8, y una versión de PHP mayor o igual a la 5.2.0.

Se debe contar además de un servidor de correo para el envío de email.

▪ **Hardware**

Las computadoras clientes deben estar conectadas en red, y deben tener un procesador a 600 MHz o superior, con 128 MB de RAM como mínimo.

El servidor deberá tener un procesador a 2.0 GHz o superior, con 512 MB de memoria RAM como mínimo.

▪ **Restricciones del diseño y la implementación**

Para efectuar cambios en el software se debe utilizar el PostgreSQL como sistema gestor de base de datos y el PHP como lenguaje de programación, ya que estos son las herramientas que se emplearon inicialmente en el desarrollo del sistema TI.

---

### **2.5.3. Actores del sistema a automatizar**

---

“Los actores del sistema, son aquellos que pueden intercambiar información con él, pueden ser recipiente pasivo de información y pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado.” [ 9 ]

En la siguiente tabla se muestran los actores del sistema que se relacionan con las actividades de mantenimiento efectuadas y el papel que desempeñan en la utilización de los nuevos requisitos funcionales:

**Tabla 5. Definición de actores del sistema a automatizar**

Actores del sistema	Justificación
Administrador	Representa al encargado de actualizar la información concerniente al sistema. <b>R40)</b>
Administrativo por Zona	Representa al encargado de controlar y visualizar toda la información de la zona a la que pertenece referente al cálculo de disponibilidad, confiabilidad, y eficacia; así como generación de informes. Es una especialización del actor Cliente. <b>R1) R3) R5) R6) R7) R9) R11) R13) R15) R16) R17) R19) R21) R22) R23) R25) R27) R28) R29) R31) R33) R34) R35) R37) R38) R39).</b>
Administrativo Central	Representa al encargado de controlar y visualizar toda la información de las distintas zonas del país, así como el cálculo de disponibilidad, confiabilidad, eficacia y generación de informes. Es una especialización del actor Cliente. <b>R2) R4) R5) R6) R8) R10) R12) R14) R15) R16) R18) R20) R21) R22) R24) R25) R26) R28) R30) R31) R32) R34) R35) R36) R38) R39).</b>

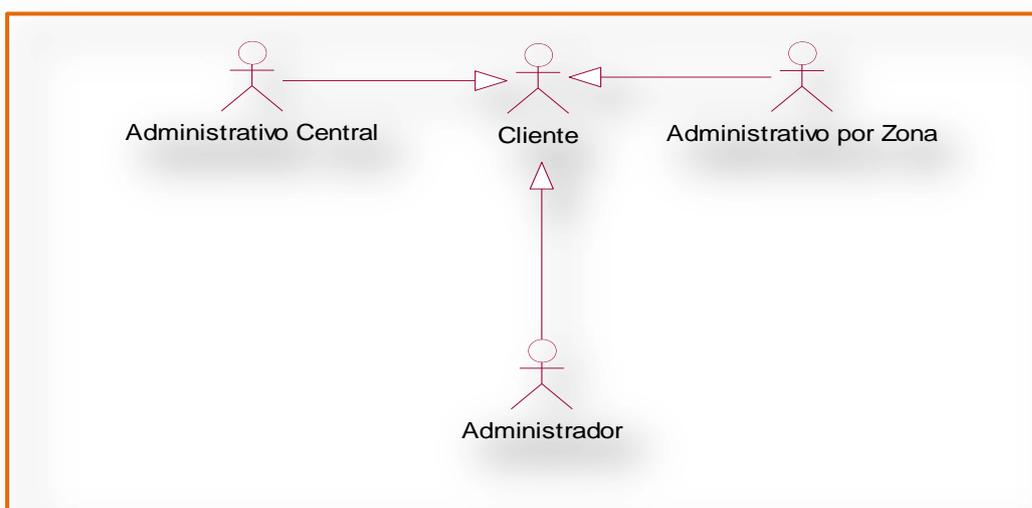


Figura 6. Actores del Sistema

### 2.5.4. Diagrama de Casos de Uso del Sistema a automatizar

En los diagramas que se relacionan a continuación aparecen casos de usos pintados de color azul, estos hacen referencias a casos de usos existentes en el sistema TI.

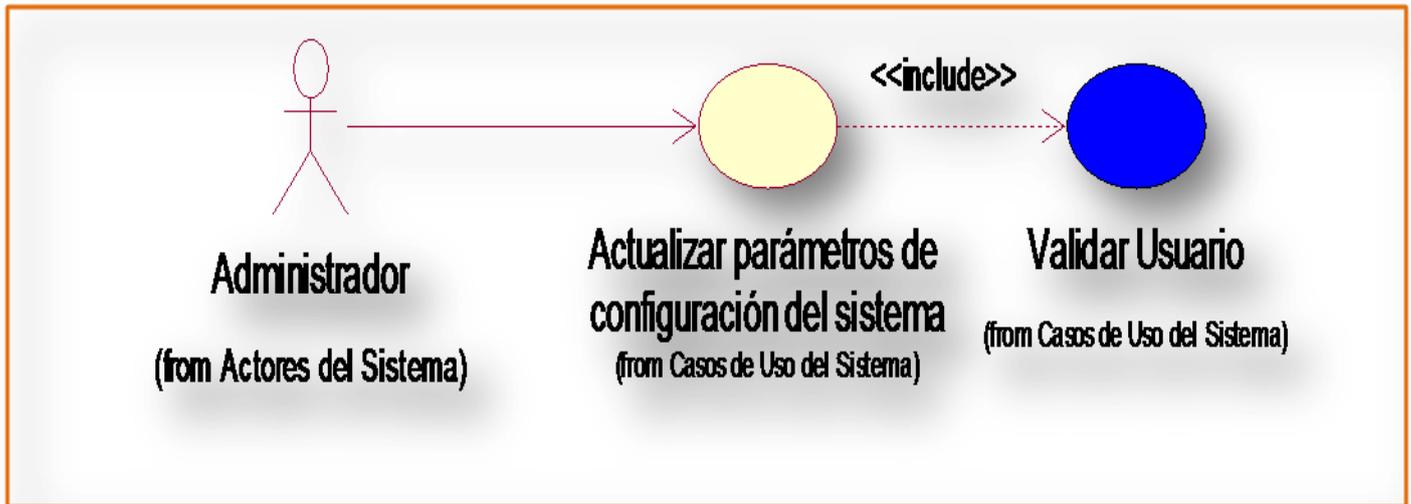


Figura 7. Diagrama de los casos de uso del sistema, paquete Administración.

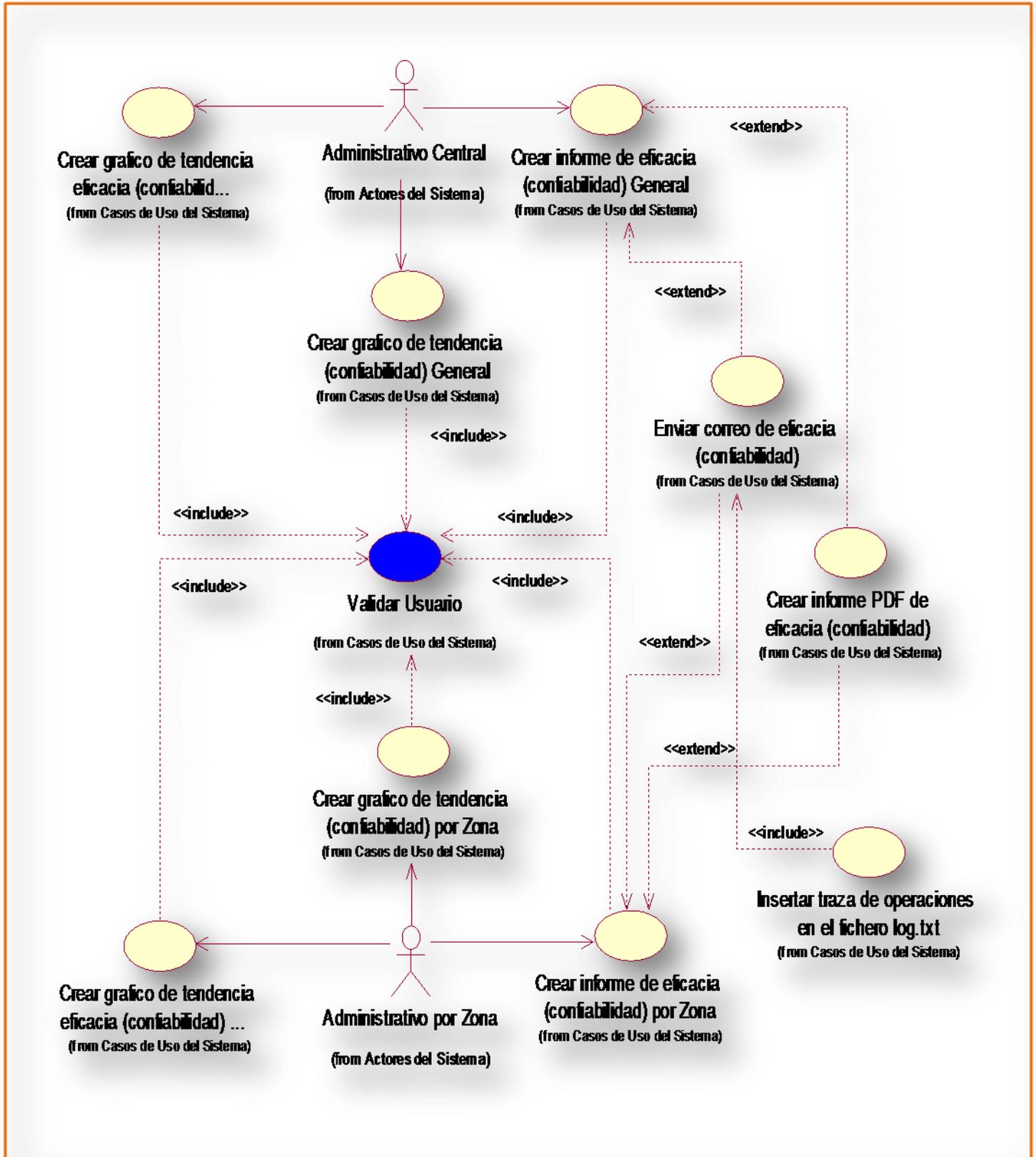


Figura 8. Diagrama de los casos de uso del sistema, paquete Administrativo -> Confiabilidad.

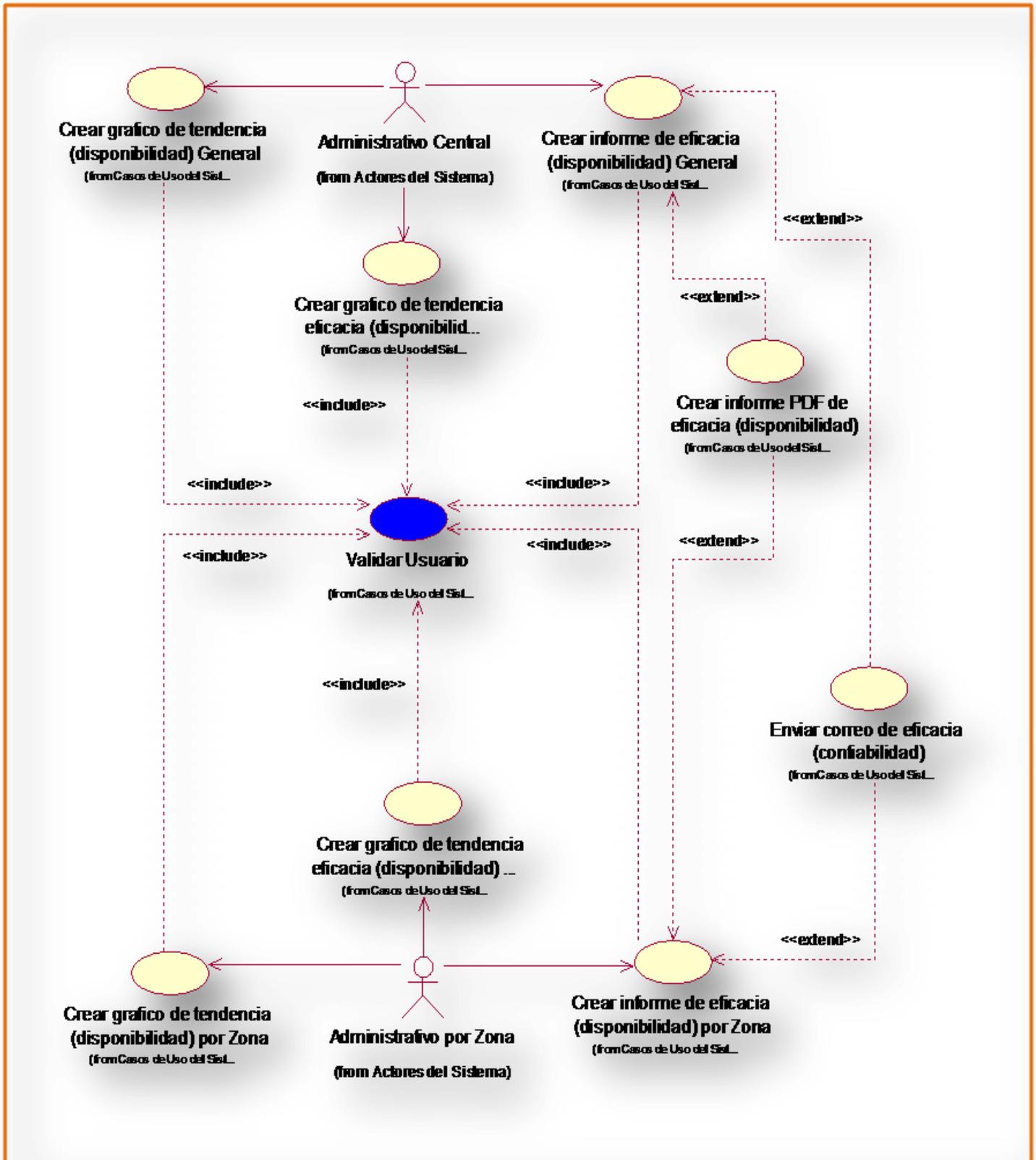


Figura 9. Diagrama de los casos de uso del sistema, paquete Administrativo -> Disponibilidad.



## 2.5.5. Descripción de los Casos de Uso

A continuación se muestran algunas de las descripciones textuales de los casos de uso del sistema, el resto puede ser consultada en la documentación digital adjunta.

**Tabla 6. Descripción del caso de uso del sistema: Crear informe de Eficacia (disponibilidad) General**

<b>Nombre del caso de uso</b>	Crear informe de Eficacia (disponibilidad) General
<b>Actores</b>	Administrativo Central (inicia)
<b>Propósito</b>	Le permite al Administrativo Central obtener el informe que calcula la eficacia de la disponibilidad de una instalación, equipo o sistema en un periodo de tiempo determinado.
<b>Resumen</b>	El caso de uso inicia cuando el Administrativo Central escoge la opción crear informe de Eficacia (disp), pudiendo visualizar la información concerniente a los sistemas, subsistemas y aeropuertos de cualquier zona, terminando el caso de uso cuando el Administrativo Central escoge otra opción o cierra la sesión.
<b>Referencias</b>	<b>R2).</b> Casos de usos asociados: Crear informe PDF de Eficacia (disponibilidad) (extendido) y Enviar correo de informe de Eficacia (disponibilidad) (extendido).
<b>Precondiciones</b>	El Usuario debe estar autenticado satisfactoriamente como Administrativo Central. Deberán existir los datos necesarios en la <b>BD</b> , para realizar el cálculo y la confección del informe.
<b>Poscondiciones</b>	Queda generado el informe de Eficacia (disponibilidad).
<b>Requisitos especiales</b>	-



**Tabla 7. Descripción del caso de uso del sistema: Crear gráfico de tendencia Eficacia (confiabilidad) por Zona**

<b>Nombre del caso de uso</b>	Crear gráfico de tendencia Eficacia (confiabilidad) por Zona
<b>Actores</b>	Administrativo por zona (inicia)
<b>Propósito</b>	Le permite al Administrativo por zona visualizar el gráfico que muestra la línea de tendencia(calculada por el método de los mínimos cuadrados) de los valores de la eficacia de la confiabilidad de una instalación, equipo o sistema en un periodo de tiempo determinado.
<b>Resumen</b>	El caso de uso inicia cuando el Administrativo por zona escoge la opción crear gráfico de tendencia Eficacia (conf), pudiendo seleccionar el tipo de gráfico que desee , si es por zona, por aeropuerto, por sistemas o subsistemas, luego observa los gráficos de la zona a la que pertenece, terminando el caso de uso cuando el Administrativo por zona escoge otra opción o cierra la sesión.
<b>Referencias</b>	<b>R17).</b>
<b>Precondiciones</b>	El Usuario debe estar autenticado satisfactoriamente como Administrativo por zona. Deberán existir los datos necesarios en la <b>BD</b> , para realizar el cálculo y la confección del gráfico
<b>Poscondiciones</b>	Queda generado el gráfico de tendencia Eficacia (confiabilidad).
<b>Requisitos especiales</b>	-



**Tabla 8. Descripción del caso de uso del sistema: Crear informe de Inventario de Equipos Existentes por Zona**

<b>Nombre del caso de uso</b>	Crear informe de Inventario de Equipos Existentes por Zona
<b>Actores</b>	Administrativo por Zona (inicia)
<b>Propósito</b>	Le permite al Administrativo por Zona obtener el informe que muestra los nombres de las instalaciones, equipos o sistemas existentes en la zona a la cual pertenece, así como la cantidad de estos.
<b>Resumen</b>	El caso de uso inicia cuando el Administrativo por Zona escoge la opción crear informe de Inventario de Equipos Existentes, pudiendo visualizar la información concerniente a los sistemas, subsistemas, equipos y aeropuertos de la zona a la cual pertenece, terminando el caso de uso cuando el Administrativo por Zona escoge otra opción o cierra la sesión.
<b>Referencias</b>	<b>R23).</b> Casos de usos asociados: Crear informe PDF de Inventario de Equipos Existentes (extendido).
<b>Precondiciones</b>	El Usuario debe estar autenticado satisfactoriamente como Administrativo por Zona. Deberán existir los datos necesarios en la <b>BD</b> , para la confección del informe.
<b>Poscondiciones</b>	Queda generado el informe de Inventario de Equipos Existentes.
<b>Requisitos especiales</b>	-



**Tabla 9. Descripción del caso de uso del sistema: Crear informe de las Quejas Inconsistentes por Zona**

<b>Nombre del caso de uso</b>	Crear informe de las Quejas Inconsistentes por Zona
<b>Actores</b>	Administrativo por zona (inicia)
<b>Propósito</b>	Le permite al Administrativo por Zona obtener el informe que muestra las quejas inconsistentes de las instalaciones, equipos o sistemas existentes en la zona a la cual pertenece, en un periodo de tiempo determinado.
<b>Resumen</b>	El caso de uso inicia cuando el Administrativo Central escoge la opción crear informe de las quejas inconsistentes, pudiendo filtrar la información por diversos criterios: supervisor, aeropuerto, sistema, subsistemas, entre otros; creando el informe de quejas inconsistentes de los sistemas de informática y las comunicaciones de la zona a la cual pertenece. El caso de uso termina cuando el Administrativo Central escoge otra opción o termina la sesión.
<b>Referencias</b>	<b>R27).</b> Casos de usos asociados: Crear informe PDF de las Quejas Inconsistentes (extendido).
<b>Precondiciones</b>	El Usuario debe estar autenticado satisfactoriamente como Administrativo por zona. Deberán existir los datos necesarios en la <b>BD</b> , para realizar la confección del listado
<b>Poscondiciones</b>	Queda generado el listado de Quejas Inconsistentes.
<b>Requisitos especiales</b>	-

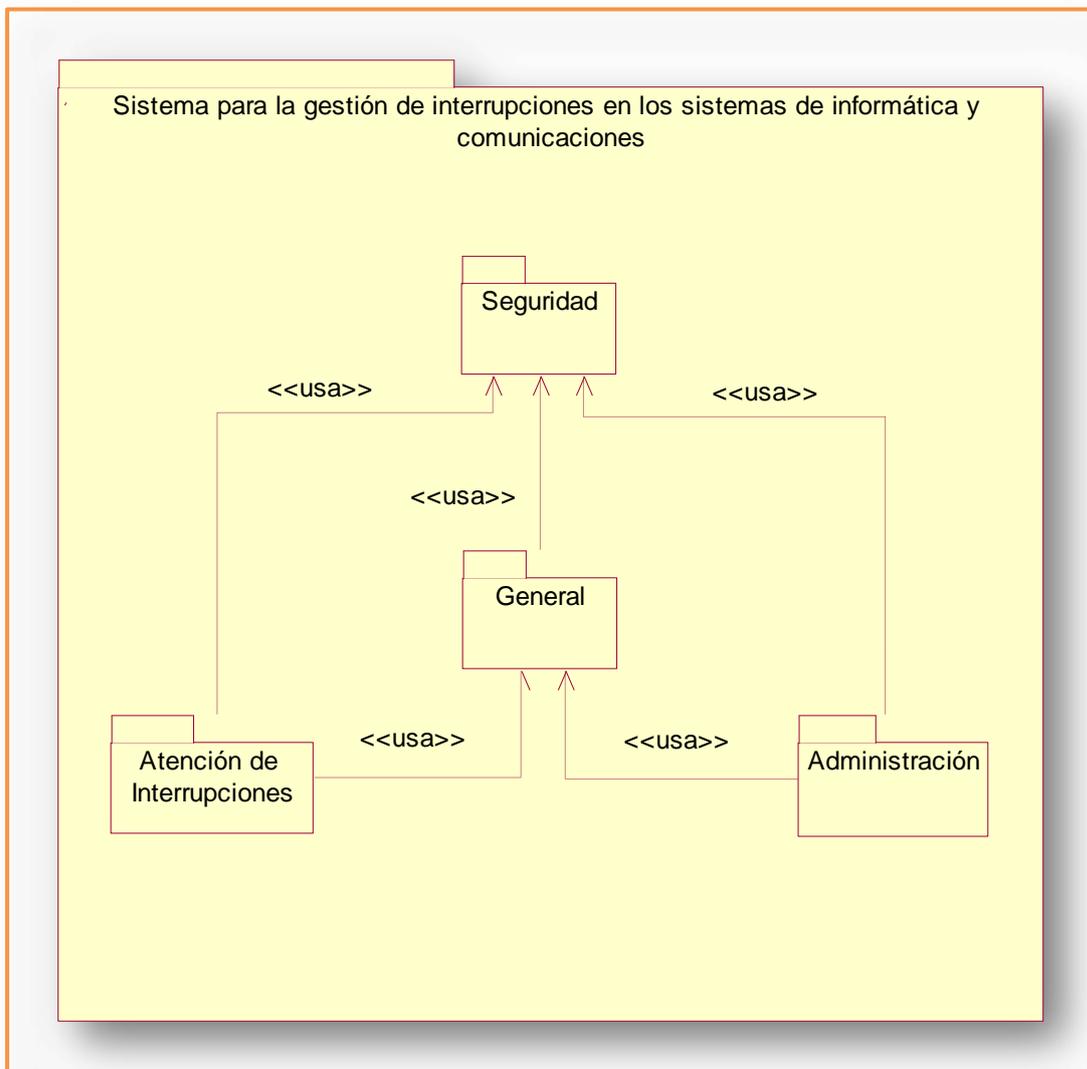


**Tabla 10. Descripción del caso de uso del sistema: Crear informe de los Tiempos Empleados en atender y solucionar las interrupciones por Zona**

<b>Nombre del caso de uso</b>	Crear informe de los Tiempos Empleados en atender y solucionar las interrupciones por Zona
<b>Actores</b>	Administrativo por zona (inicia)
<b>Propósito</b>	Le permite al Administrativo por Zona obtener el informe que muestra los tiempos empleados en atender y solucionar las interrupciones de las instalaciones, equipos o sistemas existentes en la zona a la cual pertenece.
<b>Resumen</b>	El caso de uso inicia cuando el Administrativo Central escoge la opción crear informe de los tiempos empleados, pudiendo filtrar la información por diversos criterios: técnico que inicia o termina, supervisor que cierra, aeropuerto, sistema, subsistemas, entre otros; creando el informe de los tiempos empleados en atender y solucionar las interrupciones de los sistemas de informática y las comunicaciones de la zona a la cual pertenece. El caso de uso termina cuando el Administrativo Central escoge otra opción o termina la sesión.
<b>Referencias</b>	<b>R30).</b> Casos de usos asociados: Crear informe PDF de los Tiempos Empleados en atender y solucionar las interrupciones (extendido).
<b>Precondiciones</b>	El Usuario debe estar autenticado satisfactoriamente como Administrativo por zona. Deberán existir los datos necesarios en la <b>BD</b> , para realizar la confección del informe.
<b>Poscondiciones</b>	Queda generado el informe de los Tiempos Empleados en atender y solucionar las interrupciones.
<b>Requisitos especiales</b>	-

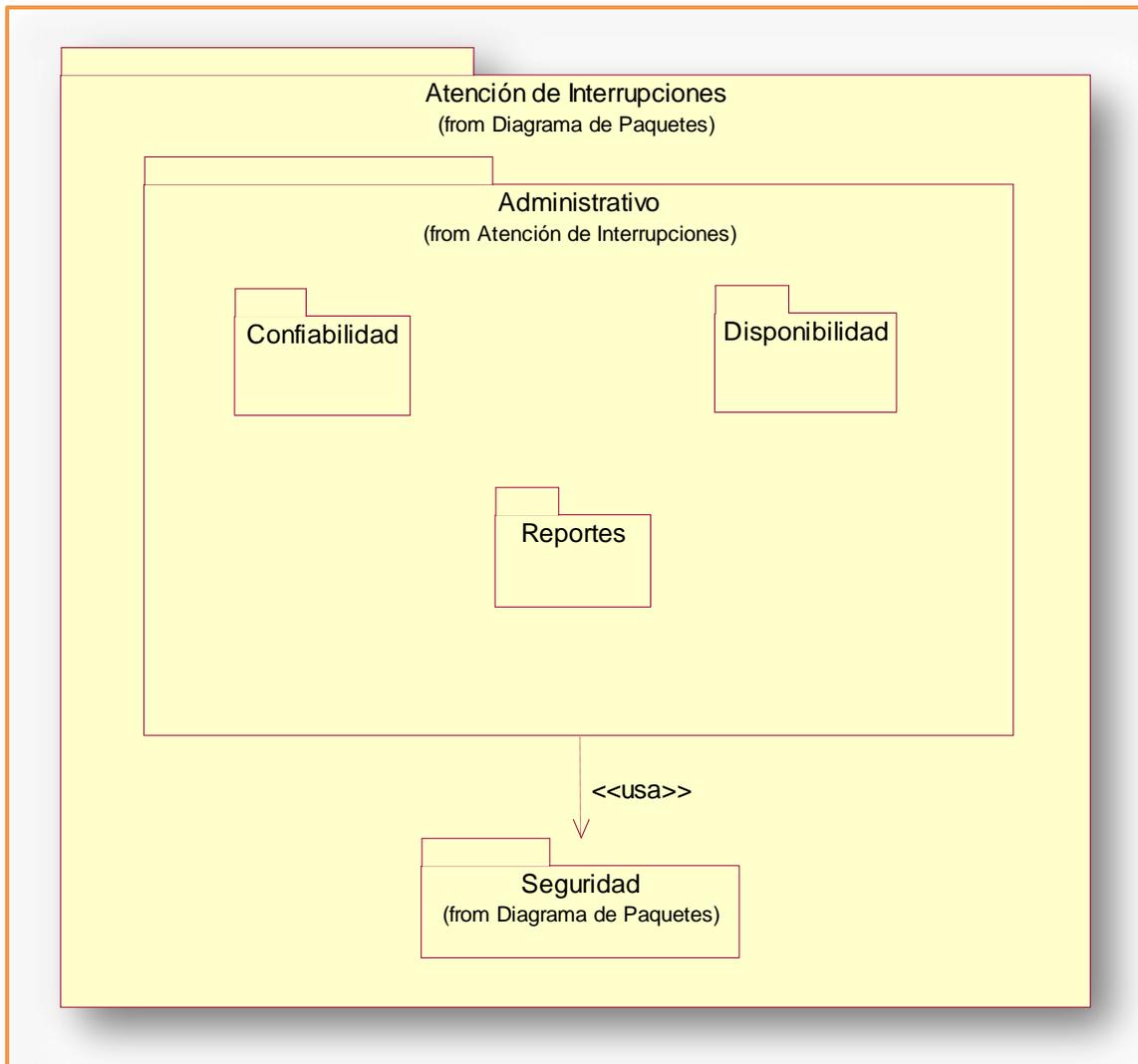
## 2.5.6. Paquetes y sus relaciones

Para una mejor comprensión, el sistema se encuentra dividido en subsistemas o paquetes donde se agrupan las principales funcionalidades para facilitar el desarrollo del mismo. Los nuevos requisitos funcionales se encuentran dentro de estos paquetes.



**Figura 10. Diagrama de Paquetes del Sistema.**

A continuación se muestra el paquete “Atención de Interrupciones” donde se encuentran los requisitos funcionales relacionados con el procesamiento de la información obtenida del cálculo de la Eficacia de los indicadores técnicos disponibilidad y confiabilidad, y los nuevos reportes incluidos en el sistema:



**Figura 11. Diagrama del Paquete “Atención de Interrupciones”.**

### **2.5.7. Flujo de diseño**

“Según la metodología RUP el flujo de trabajo del diseño tiene la responsabilidad de modelar el sistema y encontrar la forma para que este soporte todos los requisitos necesarios para cumplir su objetivo. Este flujo describe la realización física de los casos de uso usando artefactos como diagramas de clases” [ 8 ]

En el modelo del diseño, los casos de uso son realizados por las clases del diseño y sus objetos. Esto se representa por colaboraciones en el modelo de diseño y denota la realización de casos de uso del diseño.



Los elementos más comunes que conforman la tecnología Web son las páginas y los formularios. El lenguaje de modelado UML brinda varios estereotipos para extensiones Web, entre los más comunes se encuentran:

- `<<Server Page>>` Representa la página Web que radica en el servidor, la cual contiene código que se ejecuta en este. .
- `<<Client Page>>` Representa a la página Web que son entregadas al navegador, con formato HTML y a veces con código que puede ser ejecutado por el navegador.
- `<<Html Form>>` Colección de elementos de entrada que son parte de un página cliente.

Para ayudar a identificar las clases se le añadió a los nombres de las clases clientes, servidoras y formularios los prefijos *cl*, *sr* y *fr* respectivamente.

En los diagramas de clases del diseño que se proponen se incluyeron funciones dentro de las clases clientes que se utilizan para validar entrada de datos (manipular fechas, números enteros, etc.), y para la comunicación entre el navegador y el servidor a través de la técnica AJAX.

La implementación de los nuevos requisitos se diseñaron siguiendo el empleo del patrón de diseño: desarrollo modular, en la modelación reflejada en los diagramas de clases del diseño se tuvo en cuenta dicho patrón. El desarrollo modular permite modularizar las aplicaciones Web, de forma que toda la aplicación se ejecute sobre una misma página.

La clase servidora *sr index* modela a la página encargada de incluir todas las otras páginas del sistema, por lo cual será el único punto de entrada. Esto representa una gran ventaja a la hora de programar, ya que se colocan todas las tareas repetitivas, inicialización de variables, conexión a la Base de Datos y lectura de otros archivos de funciones o configuración en un solo fichero.

Los diagramas de los escenarios de los casos de uso: Enviar correo de reportes de software y Crear informe PDF de equipos existentes, se muestran en el **Anexo 3**, el resto de los diagramas pueden ser consultados en el fichero digital adjunto.

### 2.5.8. Flujo de implementación

“En la implementación empezamos con el resultado del diseño e implementamos el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares” [ 8 ]

“El modelo de implementación describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados” [ 8 ]

Un componente se define como “el empaquetamiento físico de los elementos de un modelo, como son las clases en el modelo de diseño. “

Los componentes tienen las siguientes características:

- Tienen relaciones de traza con los elementos del modelo de diseño que implementan.
- Puede que un componente implemente varios elementos de diseño.

El lenguaje de modelado UML consta de varios estereotipos para clasificar los componentes de implementación:

1. executable: representa a un programa que puede ser ejecutado
2. file: representa a un fichero que contiene código fuente o datos
3. library: representa a una librería estática o dinámica
4. table: representa a una tabla de la base de datos
5. document : representa a un documento

En la modelación de los casos de uso en el modelo de implementación se utilizó el estereotipo file, debido a que el lenguaje PHP se caracteriza por ser ejecutado por un Servidor Web. Además se representó la distribución que se realiza de los componentes al utilizar el patrón de diseño desarrollo modular.

En el **Anexo 4** se muestran los diagramas de componentes de implementación de los casos de usos Crear gráfico de tendencia (conf) por Zona y Crear informe PDF de eficacia (disp), el resto de los diagramas implementación hacia las clases del diseño se pueden consultar en la documentación digital adjunta.



## 2.6. Principios de diseño

---

El sistema informático **TI** desde sus inicios consta con una interfaz agradable al usuario para facilitar la gestión de interrupciones. Se utilizan colores con tonalidades tenues: gris, mandarina, blanco y negro.

En el sistema es escaso el uso de imágenes, favoreciendo al proceso de descarga de las páginas y disminuyendo el tiempo de espera por parte de los usuarios. La letra utilizada en todos los textos es verdana, la cual facilita la lectura fácil y sugerente.

Todo el diseño se encuentra soportado por estilos CSS lo que permite realizarle cambios sin afectar la programación. En el mantenimiento se crearon nuevos estilos que mejoran el diseño visual del sistema. Se utilizó la librería JavaScript de componentes DOJO, favoreciendo la usabilidad del software.

En el proceso de mantenimiento se respetaron los estándares de código utilizados en el sistema y se utilizó el desarrollo modular como patrón de diseño de programación.

---

### 2.6.1. Interfaz de usuario

---

Luego del mantenimiento, las interfaces visuales del sistema **TI**, resultan de fácil uso y comprensión por parte de los usuarios, brindándole facilidades a estos como el filtrado y paginación de la información.

A continuación se muestran una serie de esquemas donde se representan algunas pantallas del sistema **TI**. En todas estas pantallas se implementó el tratamiento de errores llevado a cabo en todo el sistema, además se validaron que los datos de entrada estuvieran correctamente establecidos, por ejemplo: las fechas, direcciones de correos, entre otros, y que se entraran todos los datos necesarios o indispensables para la correcta elaboración de los informes. En caso de que se escojan datos, para la realización de los informes y gráficos, que estén mal establecidos se les muestran a los usuarios mensajes aclarando qué deben hacer en cada caso.

En la **Pantalla 1** a la izquierda se muestran las funcionalidades con que cuenta el módulo administrativo relacionados con la disponibilidad de los sistemas. A su

derecha, en la región editable, se muestran múltiples opciones que puede elegir el usuario para crear el informe de Eficacia (disponibilidad), siendo de forma muy similar la página de la Eficacia (confiabilidad).

**Pantalla 1. Permite al usuario crear el informe de Eficacia (disponibilidad).**

The screenshot shows a web application interface for creating an availability report. The interface is titled "Tramitación de las Interrupciones en los Sistemas de Informática y Comunicaciones" and includes a user profile indicator "<Usuario:leoadmin>". The main content area is titled "Disponibilidad Promedio" and contains the following fields and options:

- Fecha Inicial: 2008-05-27 00:22 ...
- Fecha Final: 2009-05-27 00:22 ...
- Zona: Zona Oriente Norte (dropdown menu)
- Aeropuerto: Aeropuerto Internacional Frank País García (dropdown menu)
- Sistema: Computadoras (Clientes) (dropdown menu)
- Sub-Sistema: Aerovaradero (dropdown menu)
- Buttons: Aceptar, Cancelar

The interface also features a navigation menu on the left with categories: Administrativo, Disponibilidad, Confiabilidad, Reportes, Informes, Cliente, and Otros. The footer includes the text: "[Intranet HOG] [Aeronav] [Nivel Central ECASA] [IACC] UT Servicios Aeronauticos. Zona Oriente Norte. ECASA, SA. Todos los derechos reservados ©" and the ECASA logo.

**Figura 12. Crear el informe de Eficacia (disponibilidad).**

Si el usuario desea visualizar el informe de Tiempos Empleados en atender y solucionar las interrupciones, solo deberá escoger la opción Informes de Tiempos Empleados dentro del grupo Informes donde se le mostrará la **Pantalla 2**; en la cual el usuario puede filtrar los elementos por diversos criterios, como se muestra a continuación.

**Pantalla 2. Permite al usuario crear el informe de Tiempos Empleados.**

**Tramitación de las Interrupciones en los Sistemas de Informática y Comunicaciones** <Usuario:leoadmin>  
Inicio | Desconectarse | Contáctenos | Ayuda

**Administrativo**  
Disponibilidad  
Confiabilidad  
Reportes  
Informes

Inventario de Equipos Existentes  
Quejas Inconsistentes  
Informe de Tiempos Empleados  
Parte Diario  
Partes Enviados  
Ver registros de incidencias  
Libro Incidencias Redes  
Libro Incidencias Software

**Cliente**  
Otros

**Fecha a seleccionar**  
Fecha Inicial: 2008-05-27 00:37 ... Fecha Final: 2008-06-27 00:37 ...  
Mostrar

**Legenda:**  
T1: Tiempo entre la formulación de la queja y la apertura del reporte.  
T2: Tiempo entre la formulación de la queja y atención del reporte por 1ra vez.  
T3: Duración del reporte.

**Informe de Tiempos Empleados en atender y solucionar las Interrupciones**

**Filtrar Reportes**

Tipo de Reporte: Todos  
Técnico Inicia: Todos  
Técnico Termina: Todos  
Supervisor Cierra: Todos  
Grupo: Radio Aeropuerto  
Zona: Zona Oriente Norte  
Aeropuerto: Aeropuerto Internacional Sierra Maestra  
Sistema: Todos  
Subsistema: Todos

No.	Sistema	Subsistema	Equipo	Aeropuerto	T1	T2	T3	Detalles
124	Telefonía	Telefonos E&M	Telef E&M Hog/Mzo	MZO	0:13	2811:19	3502:25	[Ver]

<< Primero < Anterior [ 1 al 1 de 1 ] Siguiente > Ultimo >>

**Figura 13. Crear el informe de Tiempos Empleados en atender y solucionar las interrupciones.**

Aunque el sistema cuenta con un diseño muy sencillo y de fácil manejo, la ayuda le permitirá al usuario consultar qué debe hacer en cada momento, en dependencia de lo que desee, para facilitar su labor. Además para una mejor comprensión de cómo navegar a través de las nuevas funcionalidades se pueden consultar los mapas de navegación en el **Anexo 5**.

Para los informes mostrados anteriormente, la aplicación cuenta con la opción de presentarlos en varios formatos de salida, los cuales se estarán abordando como sigue.

## 2.6.2. Formato de salida de los informes

Los diferentes informes se muestran en HTML, pero se le brinda la opción al usuario de imprimir estos en formato PDF, para ello se empleó la clase escrita en PHP conocida como FPDF.

FPDF permite generar documentos PDF directamente desde PHP, es decir, sin usar la biblioteca PDFlib. La ventaja es que mientras PDFlib es de pago para usos comerciales la F (Free, en idioma Inglés) de FPDF significa gratis y libre: puede usted usarla para cualquier propósito y modificarla a su gusto para satisfacer sus necesidades.

Los reportes en PDF son hoy una excelente alternativa a los reportes HTML tradicionales. Manejarlos tiene significativas ventajas:

- Un documento PDF encapsula las fuentes que necesite para que su documento se vea tal y como lo diseñó.
- Las márgenes, tamaños y tipos de papel son más fáciles de manipular a la hora de realizar la impresión.
- Ideal cuando queremos que los reportes generados no sean modificados por los usuarios de las aplicaciones.

### **Pantalla 3. Permite al usuario visualizar en formato PDF los informes.**

Aeropuerto Orestes Acosta	Cantidad de Instalaciones	Por encima de 97.5 %	%	Confiabilidad
Meteorológico	4	4	100.00%	confiable
NDB	1	1	100.00%	confiable
Respaldo de Energia	2	2	100.00%	confiable
UPS Sistemas RadioAyuda	1	1	100.00%	confiable
<b>Total</b>	<b>4</b>	<b>4</b>	<b>100.00%</b>	<b>confiable</b>

**Figura 14. Informe PDF de Eficacia (confiabilidad).**



Otro formato de salida de los informes de gran aporte visual lo constituyen los gráficos, muy empleados en la actualidad en los sistemas para el apoyo a la toma de decisiones.

Es por ello que en el mantenimiento del sistema **TI** se empleó la librería JGraph para graficar las líneas de tendencia de los datos de interés para el cliente.

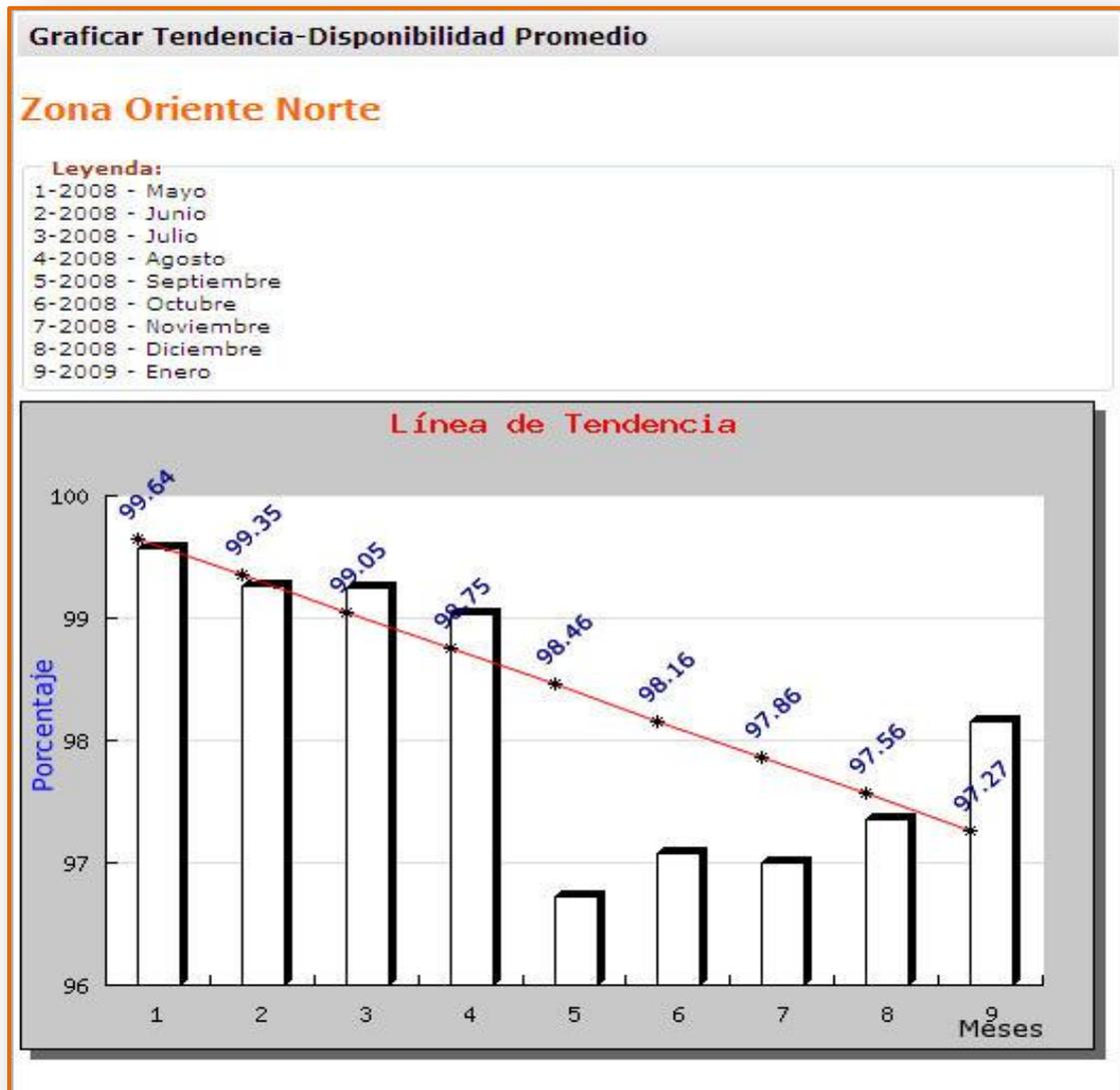
“JGraph es una librería que incluye una serie de clases-código orientado a objetos que sirven para crear imágenes con todo tipo de gráficas, dinámicamente desde PHP. La librería está muy depurada y soporta multitud de funcionalidades. Además, la mayoría de las configuraciones de las gráficas vienen con opciones por defecto, así que resulta bastante sencillo obtener resultados rápidamente.”[ 1 ]

Algunas de las características de la librería son:

- Reducido peso en bytes de las imágenes resultado. Habitualmente unas pocas KB.
- Uso de la Interpolación matemática para obtener curvas a partir unos pocos valores.
- Diversos tipos de gráficas 2D o 3D, como de puntos, líneas, tartas, barras y cajas.
- Escalas flexibles tanto en el eje X como el Y, que se ajustan al juego de datos que se tenga que representar.
- Soporte para generar gráficas con varios juegos de valores a la vez.
- Configurable con distintos tipos de colores, leyendas, tipografías e imágenes de fondo.

A continuación se muestra un ejemplo de gráfico de tendencia lineal, calculada por el método matemático de los Mínimos Cuadrados. Este ejemplo fusiona gráficos de barras y de líneas.

**Pantalla 4. Permite al usuario visualizar líneas de tendencia lineal.**



**Figura 15. Gráfico de Tendencia Eficacia (disponibilidad) por Zona.**

## 2.7. Diseño de la Base de Datos

El diseño de la **BD** incluye la definición de tablas y las relaciones entre estas, procedimientos almacenados y otras estructuras necesarias para almacenar, recuperar y eliminar la información persistente. Con esto se hace posible que la información sea conservada y protegida íntegramente.

Las clases persistentes son aquellas capaces de mantener su valor en el espacio y el tiempo. Son de vital importancia en los diagramas de clases del

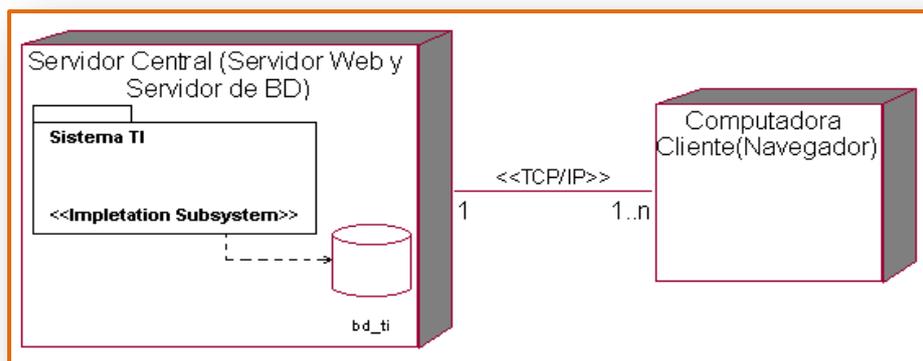
diseño pues en su interacción con las clases servidoras definen en la aplicación todos los accesos físicos que se realizarán a la **BD**. [ 8 ]

Debido a la longitud de los diagramas y para que pudiera lograrse una mejor comprensión, el modelo lógico de datos y el diagrama de clases persistentes no fueron incluido en el trabajo, se encuentran detallados en la documentación digital adjunta.

## 2.8. Diagrama de Despliegue

“El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo...En sí mismo representa una correspondencia entre la arquitectura software y la arquitectura del sistema (el hardware).” [ 8 ]

A continuación se muestra la distribución física del sistema informático **TI**:



**Figura 16. Diagrama de Despliegue.**

El nodo a la izquierda representa al Servidor de la aplicación, en el están ubicados el Servidor Web y el Servidor de Base de datos, a la derecha se encuentran simbolizadas bajo el nombre de Computadora Cliente, todas aquellas computadoras desde las cuales se accederá al sistema para formular las quejas, visualizar la información, etc.

## 2.9. Valoración de Sostenibilidad

La implantación de un sistema informático trae consigo consecuencias positivas y/o negativas en el entorno de aplicación, es por eso que se procederá a la

valoración de sostenibilidad del sistema como producto informático a continuación, para contribuir así con el desarrollo sostenible del mismo.

La valoración de sostenibilidad de un producto informático, según Rita Concepción es: “el proceso de evaluación de impactos ambientales, socio-humanista, administrativos y tecnológicos de un producto informático, previsible desde el diseño del proyecto, que favorece su autorregulación, para la satisfacción de la necesidad que resuelve, con un uso racional de recursos y la toma de decisiones adecuadas a las condiciones del contexto y el cliente.” [ 6 ]

### Dimensión Administrativa

Para el análisis de la factibilidad económica del proceso de mantenimiento del software llevado a cabo en el sistema **TI**, se utilizó la extensión propuesta por el Modelo Constructivo de Costos 2.0 (COCOMO, por sus siglas en inglés) para el proceso de **MS**; en la cual existen algunas diferencias con los modelos de diseño inicial y post-arquitectura.

“El COCOMO tiene como objetivo la estimación del costo asociado al desarrollo de un software. Utiliza los puntos de función para calcular el tamaño de un producto de software y el esfuerzo asociado a su desarrollo. El empleo de COCOMO permite realizar una estimación de la cantidad de personas que se requieren utilizar para el desarrollo de un sistema, así como el tiempo de desarrollo aproximado.” [ 7 ]

En el modelo para el mantenimiento el esfuerzo asociado a este se le denomina

$PM_M$ . Para calcular  $PM_M$  se utilizan los multiplicadores de esfuerzo, los factores de escala y una serie de valores calibrados (A, B, C, D).

$$PM_M = A * (Size_M)^E \prod_{i=1}^{15} EM_i$$

El esfuerzo asociado al mantenimiento utiliza todos los multiplicadores de esfuerzo (17 en total), excepto el SCED y RUSE.



El  $Size_M$  representa el tamaño estimado en miles de instrucciones fuentes asociado a la actividad de mantenimiento del software, y se calcula por la siguiente ecuación:

$$Size_M = (Size_{Added} + Size_{Modified}) * MAF$$

Siendo el  $Size_{Added}$  el tamaño estimado en miles de instrucciones fuentes asociado a los nuevos requisitos funcionales incorporados al sistema, y el  $Size_{Modified}$  el tamaño estimado en miles de instrucciones fuentes modificadas.

El  $MAF$  representa el factor de ajuste del mantenimiento, y es igual a:

$$MAF = 1 + \left( \frac{SU}{100} * UNFM \right)$$

Donde SU es una constante que representa la comprensión del software y el UNFM es la no familiaridad de los programadores.

Las tablas con las entradas externas, consultas externas y ficheros lógicos internos, asociadas a los nuevos requisitos funcionales incorporados al TI y utilizadas para el cálculo de los puntos de función, pueden ser consultadas en el documentación digital adjunta.

Los elementos que se tienen en cuenta para calcular el total de puntos de función desajustados resultantes de los nuevos requisitos funcionales se encuentra en la siguiente tabla:

**Tabla 11. Puntos de función desajustados.**

Elementos	Simplex	X Peso	Medios	X Peso	Complejos	X Peso	Subtotal de puntos de función
<b>Ficheros lógicos internos</b>	32	7	0	10	0	15	224
<b>Ficheros de interfaces externas</b>	0	5	0	7	0	10	0
<b>Entradas externas</b>	4	3	0	4	0	6	12
<b>Salidas externas</b>	0	4	0	5	0	7	0
<b>Consultas Externas</b>	1	3	1	4	33	6	205
<b>Total de puntos de función desajustados (UFP)</b>							<b>441</b>



El cálculo de las instrucciones fuentes, según COCOMO 2.0, se basa en la cantidad de instrucciones fuentes por punto de función que genera el lenguaje de programación empleado.

En el desarrollo del sistema en general se utilizaron los lenguajes de programación PHP y Javascript. Se estimó que el uso del PHP en la realización de los nuevos requisitos funcionales fue aproximadamente el 85 % del total de código utilizado y el del Javascript solamente un 15 %, ya que este generalmente se utilizó para realizar validaciones de los formularios de entrada de datos, así como para implementar las características de la técnica AJAX .

**Tabla 12. Instrucciones fuentes por lenguaje utilizado.**

Lenguajes	% uso	UFP del leng.	Ratio	UFP(leng)*Ratio
<b>PHP</b>	85	85 % de 441 = 374.85	69	25864.65
<b>Javascript</b>	15	15 % de 441 = 66.15	56	3704.4

$$\text{SLOC} = \text{UFP (PHP)} * \text{ratio (PHP)} + \text{UFP (Javascript)} * \text{ratio (Javascript)}$$

$$= 29569.05$$

**Tabla 13. Instrucciones fuentes.**

Características	Valor
<b>Puntos de función desajustados</b>	441
<b>Instrucciones fuentes</b>	29569.05
<b>Miles de instrucciones fuentes (MI)</b>	29.6

El tamaño estimado en miles de instrucciones fuentes asociado a los nuevos requisitos funcionales  $Size_{Added}$  es igual a 29.6

Teniendo en cuenta que la herramienta seleccionada para el desarrollo de los requisitos contribuye con técnicas visuales al desarrollo de la aplicación Web y considerando que esto facilita el trabajo, se aplicó una reducción del 20 % de las instrucciones fuentes, obteniéndose:

$$Size_{Added} = 23.66$$



Al desarrollarse el mantenimiento del sistema, esto implicó que se modificara aproximadamente el 5 % de las instrucciones fuentes originales, quedando el

$$Size_{Modified} = 1.07$$

Para calcular el  $Size_{Modified}$  se tuvo en cuenta los datos arrojados por el COCOMO en la investigación realizada por el Ing. Oscar Gabriel Reyes Pupo durante el desarrollo del sistema informático **TI**, donde SLOC es igual a 21 724.2 instrucciones fuentes.

Para un valor de 0.4 de la constante SU y UNFM con un valor de 1.0 el  $MAF$  es igual a 1.4, quedando:

$$Size_M = (Size_{Added} + Size_{Modified}) * MAF = 34.62$$

Los valores de los multiplicadores de esfuerzo asociados al mantenimiento, así como los factores de escala, pueden ser consultados en el **Anexo 7**

$$PM_M = A * (Size_M)^E \prod_{i=1}^{15} EM_i$$

E depende de los factores de escala (SF):

$$E = B + 0.01 * \sum_{j=1}^5 SF_j$$

El esfuerzo asociado al mantenimiento del sistema es de 45,8 hombres mes.

$$TDEV = C * PM_M^F$$

$$\text{Donde } F = D + 0.2 * (E - B)$$

$$TDEV = 12.65$$

$$CH = PM_M / TDEV = 4$$

El desarrollo del proceso de mantenimiento del software requiere aproximadamente 13 meses para realizarlo con 4 personas.



Como se cuenta con 1 desarrollador y el apoyo de 3 expertos el CHR es igual a 4, el mantenimiento del sistema se desarrollaría en aproximadamente 13 meses.

$$TDEV = PM/CHR = 13$$

El salario medio de los expertos se considera que es \$385 y el del desarrollador \$155. Se denomina CHM al costo por Hombre Mes. El costo total del mantenimiento del sistema se determina por:

$$CHM = \$ 1310$$

$$\text{Costo} = CHM * TDEV = \$ 17 030$$

El costo total del proyecto asciende a \$ 17 030

La implantación del Producto Informático (en lo adelante **PI**) no incurre en gastos adicionales por compra de equipos, pues la ECASA consta del necesario. Además la implantación del **PI** no incurre en pagos de licencias por el uso de software, ya que todo el software utilizado en el desarrollo del **PI** es libre, así como el sistema operativo sobre el cual se ejecuta el sistema.

Con el sistema **TI**, se ahorran materiales de impresión y papel en más de un 90%, se ahorra el tiempo en la realización de las actividades relacionadas con la gestión de las interrupciones, así como en el cálculo de la confiabilidad, disponibilidad y eficacia de los sistemas de informática y las comunicaciones. Además disminuye la tarifa telefónica, al ser tramitadas las interrupciones en su totalidad por el sistema informático.

Con el **PI** se gestionan de una manera rápida y eficiente las interrupciones generadas por los sistemas de informática y las comunicaciones, logrando una mayor calidad y rapidez.

Con la inclusión de nuevas funcionalidades el sistema contribuye a la toma de decisiones, además permite visualizar de forma más específica y estadística los informes y gráficos de tendencia de la disponibilidad y confiabilidad de los sistemas, así como la eficacia de los mismos. El **PI** no genera ingresos indirectos o directos monetarios.



Por todo lo antes expuesto se concluye que el **PI** es sostenible en la dimensión Administrativa.

### **Dimensión Socio- Humanista**

El sistema **TI** favorece al proceso de gestión de las interrupciones de los sistemas de informática y las comunicaciones, propiciando una mayor eficiencia, control y calidad en las actividades. El producto ha sido desarrollado teniendo en cuenta las necesidades del personal que ha de utilizarlo. Disminuye el esfuerzo físico de los trabajadores y las probabilidades de errores a la hora de manipular la información de las interrupciones.

El **PI** utiliza el PostgreSQL como sistema gestor de base de datos, Apache como servidor Web, y PHP como lenguaje de programación. Todos estos softwares son multiplataformas, por lo que el costo de emigración del sistema hacia otros sistemas operativos será mínimo.

Como el sistema **TI** se basa en el Procedimiento Específico PE-2806 Revisión 1.0 basado en la Norma ISO 9001.2000, procedimiento utilizado en todos los aeropuertos del país para tramitar las interrupciones, es aplicable a todas las instalaciones y áreas de comunicaciones de las diferentes zonas y aeropuertos.

El **PI** aumenta la cultura profesional y general, ya que introduce al usuario en el mundo de la Web, siendo hoy en día uno de las plataformas de intercambio de información más popular. Desarrolla en los trabajadores la responsabilidad, haciéndole entender la necesidad de utilizar correctamente el sistema e informar de cualquier fallo que este presente. Con la implantación de **PI** no se disminuyen las capacidades de empleo, ni se generan.

Por los beneficios que trae consigo el uso del **PI** no ha existido hasta el momento rechazo por parte de los trabajadores, aunque es recomendable capacitarlos para usarlo, y así explicarle las ventajas que trae la implantación del software y las facilidades que brinda.

Por todo lo antes expuesto se concluye que el **PI** es sostenible en la dimensión Socio-Humanista.



## Dimensión Ambiental

Con el sistema **TI** se disminuyen considerablemente los materiales de oficina a utilizar para la gestión de las interrupciones, el diseño de las interfaces del **PI** son lo más amigable posible y de fácil uso por parte de los usuarios.

Se utilizan colores adecuados y un tamaño de letra que no daña ni cansa la vista, logrando una mayor identificación del usuario con la aplicación.

La solución propuesta en el presente trabajo no tendrá un impacto directo sobre el medio ambiente.

Todos los software que se utilizaron en el desarrollo del **PI** son libres, esto contribuye a que se puedan reutilizar códigos existentes en el mundo y que sirvan para disminuir el tiempo de desarrollo. Así como el mismo código generado podría ser utilizado para la reutilización con vista a la implementación de otros módulos o nuevos requerimientos del sistema.

La implantación del **PI** aumenta el uso de la computadora pudiéndole causar enfermedades profesionales a los usuarios. Por lo que se recomienda una correcta postura de los usuarios en las sillas, así como tener el monitor a la altura de los ojos y utilizar protector de pantalla.

Por todo lo antes expuesto se concluye que el **PI** es sostenible en la dimensión Ambiental.

## Dimensión Tecnológica

En sentido general los usuarios del sistema cuentan con una buena preparación profesional en cuanto al uso de la computación, lo que le sirve como base para el empleo del **PI** y del sistema en sí. En caso de que existan usuarios que no estén acostumbrados o no se adapten al sistema, se recomienda capacitarlos con cursos y deberán utilizar la ayuda adjunta al sistema.

La organización cuenta con una infraestructura que le permite la implantación y correcta explotación del **PI**. No es necesario incurrir en gastos adicionales en la compra de equipamiento para la implantación del **PI** y del sistema en general.

Entre los factores tecnológicos que pueden afectar la utilización del **PI** se encuentra la no disponibilidad en momentos críticos de los servicios de red en la



organización, así como una caída del servidor donde se encuentra la información.

Por todo lo antes expuesto se concluye que el **PI** es sostenible en la dimensión Tecnológica.

En sentido general se puede concluir que luego de realizar una valoración desde el punto de vista administrativo, socio-humanista, tecnológica y ambiental, el **PI** presenta características que lo hacen ser factible y perdurable en el tiempo, sobre todo si centra en la necesidad que provocó el surgimiento de esta investigación.

## 2.10. Valoración de la propuesta

---

Una vez implantado el sistema **TI** resultante del proceso de mantenimiento del software, se realizaron entrevistas al cliente, con el objetivo de evaluar la satisfacción de los mismos respecto al producto informático.

### 2.10.1. Entrevista

---

Se realizaron entrevistas **Anexo 6** con el objetivo de evaluar la satisfacción del cliente respecto a los resultados obtenidos del proceso de mantenimiento del sistema **TI**. Los entrevistados, de manera general, coincidieron en que están satisfechos con los resultados obtenidos valorando que:

- Se logró aumentar la seguridad del sistema.
- El indicador técnico: Eficacia cumple con los requerimientos requeridos.
- Las gráficas de tendencia ayudan significativamente a la toma de decisiones.
- Las nuevas funcionalidades perfeccionan la calidad del sistema.

## 2.11. Conclusiones

---

En este capítulo se han descrito las distintas actividades del proceso de mantenimiento del software llevadas a cabo en el sistema **TI**.



Mediante el mantenimiento perfectivo se logró incorporar los nuevos requisitos funcionales al sistema, a través de los flujos de trabajos propuestos por la metodología RUP.

Se logró la mantenibilidad del producto software aplicando las actividades propuestas por el mantenimiento preventivo.

Con el mantenimiento correctivo se alcanzó una mayor confiabilidad e integridad, y mediante el mantenimiento adaptativo se logró ajustar al sistema **TI** a las nuevas condiciones.

El producto informático resultante del proceso de **MS** teniendo en cuenta sus características, queda clasificado como sostenible, además de considerarse factible la solución empleada.

La satisfacción de los usuarios y el cliente fue percibida a través de las entrevistas, que arrojaron que el sistema cumple con las expectativas de estos, y resuelve el problema definido en esta investigación.



## Conclusiones Generales

---

Con el desarrollo del proceso de mantenimiento del software en el sistema informático **TI**, dedicado a favorecer el proceso de Gestión de Interrupciones de los sistemas de informática y las comunicaciones en la Zona Oriente Norte de la ECASA, se ha dado cumplimiento al objetivo de esta investigación, dando como resultado una nueva versión del producto informático que se caracteriza por ser más seguro, abarcador, fiable y informativo; además de constar con nuevas funcionalidades que favorecen al proceso de gestión de interrupciones.

A través del estudio realizado para lograr el objetivo de esta investigación se puede concluir que:

- La nueva versión del sistema **TI** se valora como sostenible en la dimensión socio-humanista, ambiental, administrativa y tecnológica.
- El proceso de mantenimiento resulta una de las etapas más complejas e importantes dentro del ciclo de vida del software, ya que permite ampliar, perfeccionar y adaptar las funcionalidades existentes y las nuevas a las necesidades reales del usuario.

La hipótesis enunciada al inicio de esta investigación queda demostrada a través de los argumentos expuestos en el presente documento, los que permiten arribar a la conclusión que con la nueva versión del sistema se logra favorecer el proceso de gestión de interrupciones de los sistemas de informática y las comunicaciones en la Zona Oriente Norte de la ECASA.



## Recomendaciones

---

A partir de la investigación realizada y para continuar el desarrollo de este trabajo se recomienda:

- Constatar una regulación o procedimiento de calidad que rijan el cálculo del indicador técnico Eficacia, ya que este es tan imprescindible para la correcta toma de decisiones de los directivos de los aeropuertos.
- Hacer un estudio estadístico más detallado de los tipos de gráficos de tendencia y pronóstico que son más factibles para predecir el comportamiento de los datos de interés del cliente.
- Generalizar el uso de la aplicación al resto de las Zonas Aeroportuarias del país.
- Confeccionar un plan de capacitación para los usuarios que usarán la herramienta para que aprovechen al máximo las facilidades que brinda el sistema informático **TI**.



## Bibliografía y Referencias Bibliográficas

---

1. (2006). "JPGraph." from <http://www.aditus.nu/jpgraph/index.php>
2. (2007). "¿Qué es cada tecnología? ." Desarrollo de Web: Manual, from <http://www.desarrolloweb.com/>
3. Alain April, J. H. H. (2004). Software Maintenance Maturity Model. The software maintenance process model.
4. Carrasquel, Y. (2006). Sistema integral para el control de las operaciones
5. Cedeño, G. G. (2007). Sistema de Tramitación de Interrupciones en los Sistemas de Comunicaciones Zona Oriente Norte, Ecasa S.A. Trabajo de Diploma para optar por el título de Ingeniero Informático. Facultad de Informática y Matemática. Holguín, Oscar Lucero Moya.
6. Concepción, R. (2006). Procedimiento para la valoración de sostenibilidad de un Producto Informático.
7. González, F. R. (1999). COCOMO v2. Modelo de Estimación de Costes para proyectos software. Universidad de Castilla -La Mancha. Campus de Ciudad Real.
8. Jacobson, I. (2000). El Proceso Unificado de Desarrollo de Software. Addison Wesley
9. Jacobson, I. (2002). "El Lenguaje Unificado de Modelado".
10. Matos, R. M. (2006). "Diseño de Bases de Datos." from <ftp://serverinfo/Docs/Persistencia/Teoría/Bases de Datos/Maria-Diseño de base de datos.rar>.
11. Michelena, S. (2007). Tramitación de interrupciones. Cálculo de la disponibilidad y confiabilidad.
12. Press, I. C. S. (1998). Standard for Software Maintenance. Los Alamitos, CA.
13. Pressman (2002). Ingeniería del Software. Un enfoque práctico, McGraw-Hill/Interamericana de España.
14. Pupo, O. G. R. (2008). Mantenimiento del sistema de tramitación de interrupciones de los sistemas de comunicaciones Facultad de Informática-Matemática, Universidad de Holguín "Oscar Lucero Moya".
15. Rodríguez, B. (2007). "¿Qué es una aplicación Web? ." from <http://www.soho.com.mx>.



16. Ruiz, F. (2006). "Curso: Mantenimiento del Software." from <http://alarcos.inf-cr.uclm.es/per/fruiz/cur/mso/mso.htm>.
17. Wikipedia. (2009). "Sistemas Gestores de Bases de Datos." from <http://es.wikipedia.org>.
18. Zervaas, Q. (2008). Practical Web 2.0 Applications with PHP.



## Glosario de términos

---

**Aeropuerto Internacional:** Todo aeropuerto designado por el Estado contratante en cuyo territorio esté situado, como puesto de entrada o de salida para el tráfico aéreo internacional, donde se llevan a cabo los trámites de Aduanas, Inmigración, Sanidad pública, Reglamentación veterinaria, fitosanitaria, y procedimientos similares.

**Aeronave:** Toda máquina que puede sustentarse en la atmósfera por reacciones del aire que no sean las reacciones del mismo contra la superficie de la tierra.

**BD:** Base de Datos.

**C:** Lenguaje de programación de alto nivel.

**CSS:** Las hojas de estilo en cascada (Cascading Style Sheets, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML.

**ECASA:** Empresa Cubana de Aeropuertos y Servicios Aeronáuticos.

**Hardware:** Conjunto de elementos materiales que componen un ordenador. En dicho conjunto se incluyen los dispositivos electrónicos y electromecánicos, circuitos, cables, tarjetas, armarios o cajas, periféricos de todo tipo y otros elementos físicos.

**Hipertexto:** Texto que incorpora nexos o relaciones a otros documentos.

**Hipervínculo:** Enlace, elemento que al ser pinchado conduce a otro sitio en Internet, una Web, una dirección de correo, etc.

**HTML:** Es el Lenguaje de Marcado de Hipertexto (HyperText Markup Language), que se utiliza para crear los documentos a los que se accede a través de navegadores WWW.

**HTTP:** Es el Protocolo de Transferencia de Hipertexto (HyperText Transfer Protocol), un protocolo Web que controla las peticiones y servicios de documentos HTML.



**Interfaz:** Parte de un programa informático que permite a éste comunicarse con el usuario o con otras aplicaciones permitiendo el flujo de información.

**Internet:** Red de redes a escala mundial de millones de computadoras interconectadas con el conjunto de protocolos TCP/IP.

**Intranet:** Red local que utiliza herramientas de Internet. Se puede considerar como una internet privada que funciona dentro de una organización.

**JAVA:** Lenguaje de programación de alto nivel.

**Multimedia:** Término empleado para describir diversos tipos de medios (media) que se utilizan para transportar información (texto, audio, gráficos, animación, video, e interactividad).

**Página Web:** Es un documento escrito en formato HTML que es visible en un navegador.

**Protocolo:** Es una serie de reglas que utilizan dos computadoras para comunicarse entre sí. Cualquier producto que utilice un protocolo dado puede funcionar con otros productos que utilicen el mismo protocolo.

**Script:** Instrucciones internas de una aplicación.

**Sistema Operativo:** Conjunto de software que controla los distintos recursos del ordenador.

**Software:** Conjunto de programas que puede ejecutar el hardware para la realización de las tareas de computación a las que se destina. Es el conjunto de instrucciones que permite la utilización del equipo.

**SIMA:** Sistema internacional de mensajería aeronáutica.

**SQL:** Lenguaje de Consultas Estructurado (Structured Query Language). Lenguaje de consultas y programación de Base de Datos. Se utiliza ampliamente para tener acceso a datos, consultar, actualizar y administrar sistemas de bases de datos relacionales.



**TCP/IP:** Protocolo de Control de Transmisión/Protocolo de Internet (Transmission Control Protocol/Internet Protocol), es el conjunto de protocolos que rigen las comunicaciones entre las computadoras en Internet.

**TI:** Sistema de tramitación de las interrupciones de los sistemas de informática y las comunicaciones.

**UEB:** Unidad empresarial de base.

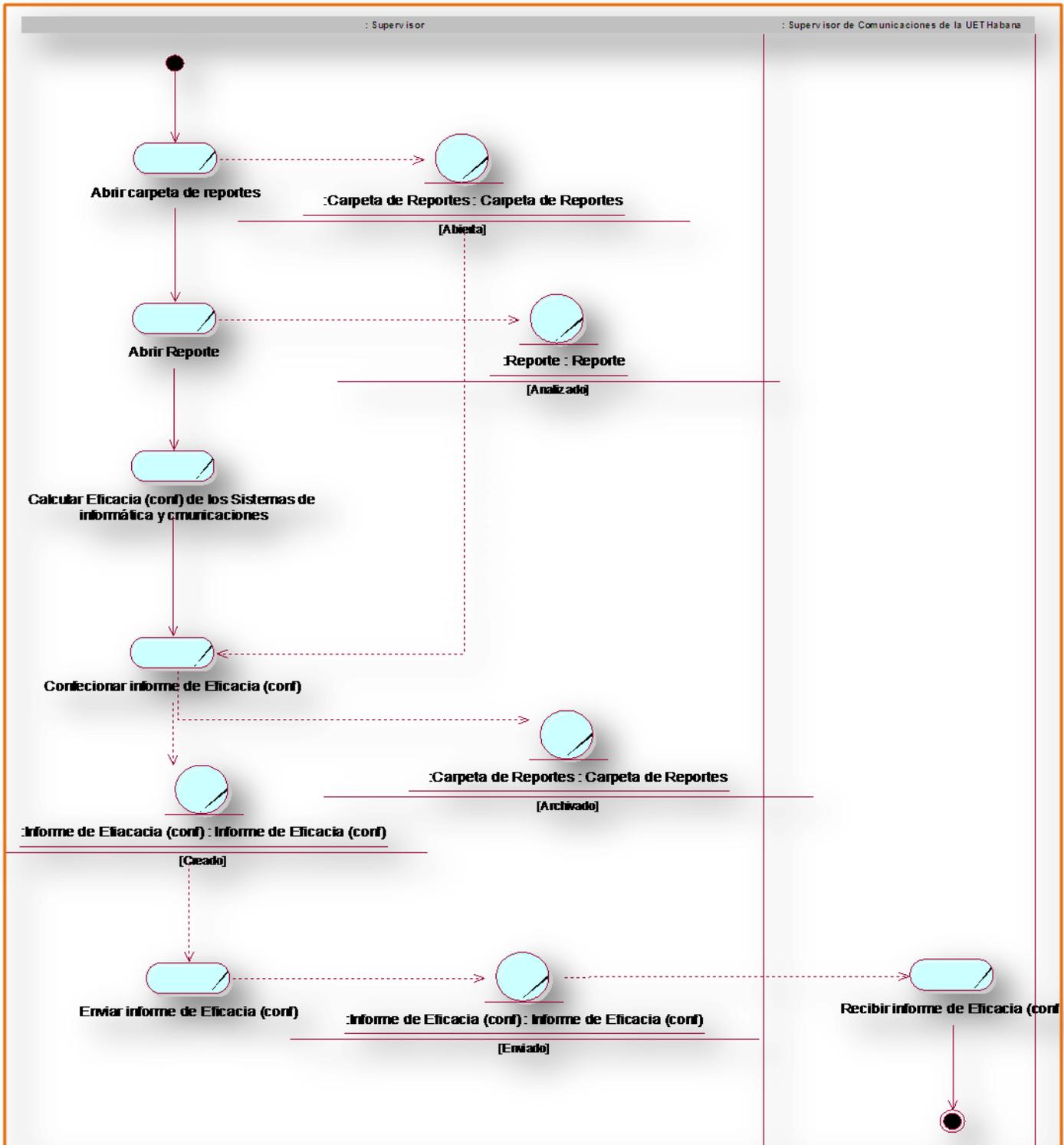
**UET:** Unidad empresarial territorial.

**UT:** Unidad territorial.

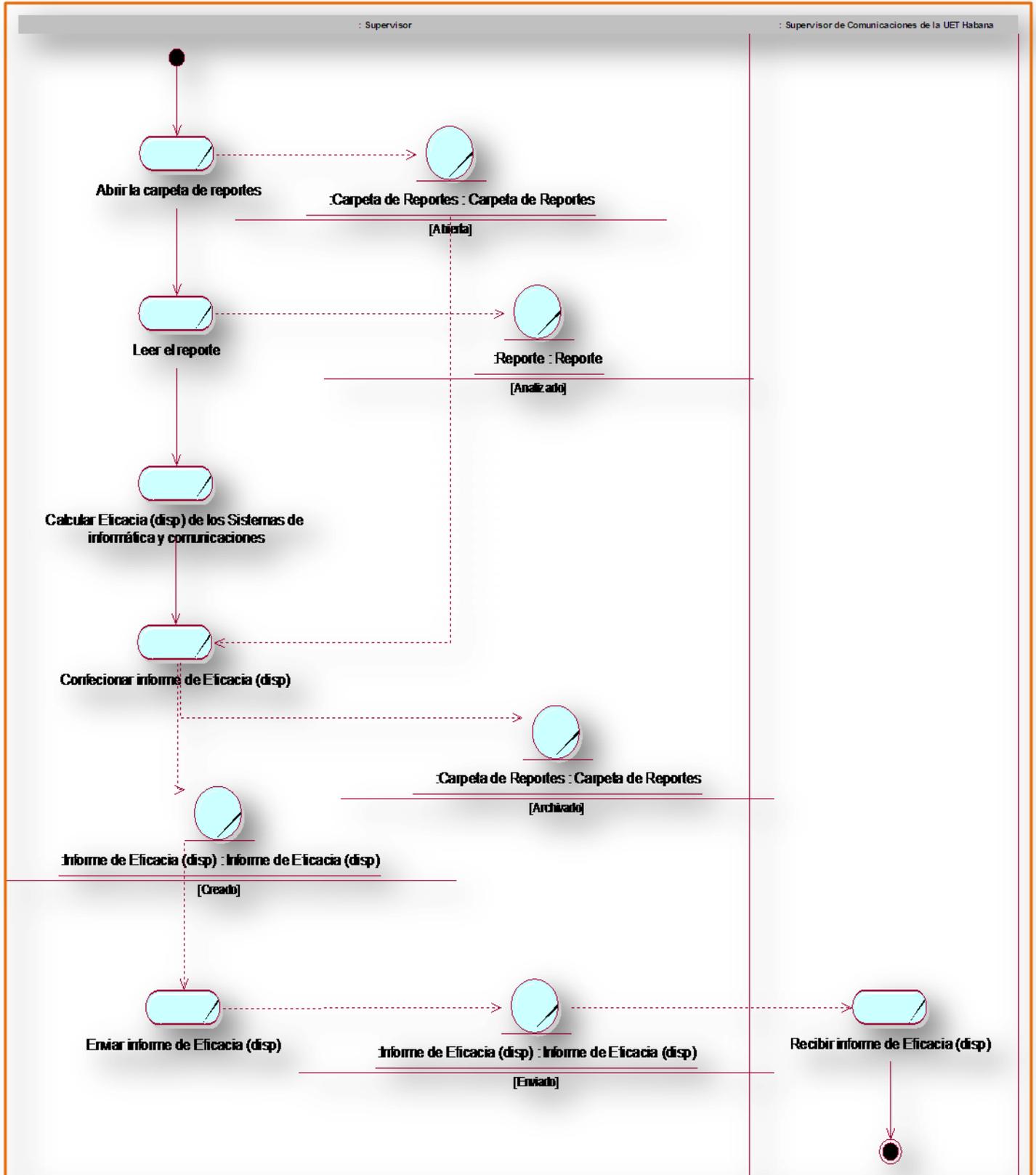
## Anexos

### ANEXO 1. Diagrama de actividades.

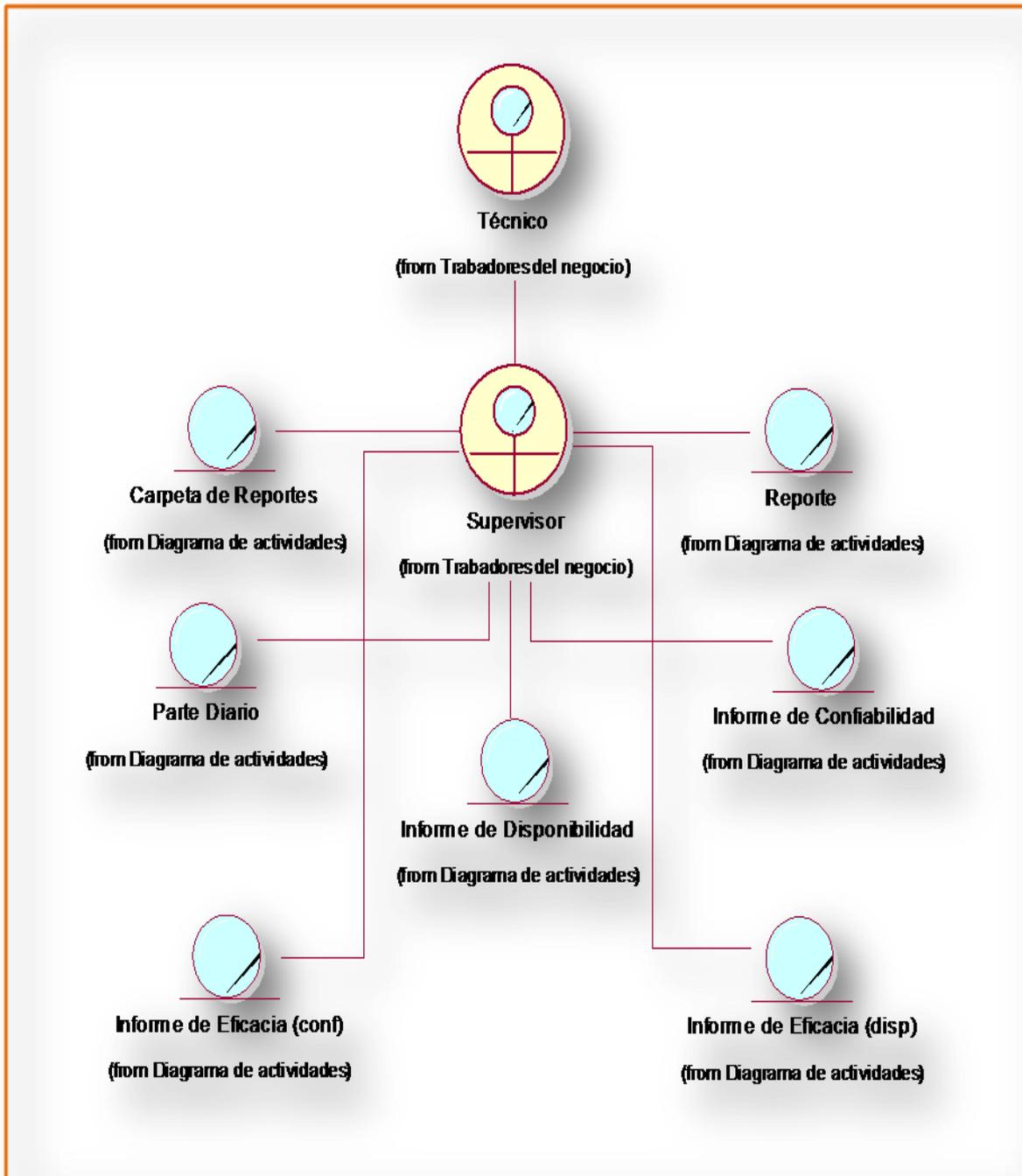
#### Caso de uso Calcular Eficacia (conf) de los sistemas de informática y las comunicaciones



### Caso de uso Calcular Eficacia (disp) de los sistemas de informática y las comunicaciones

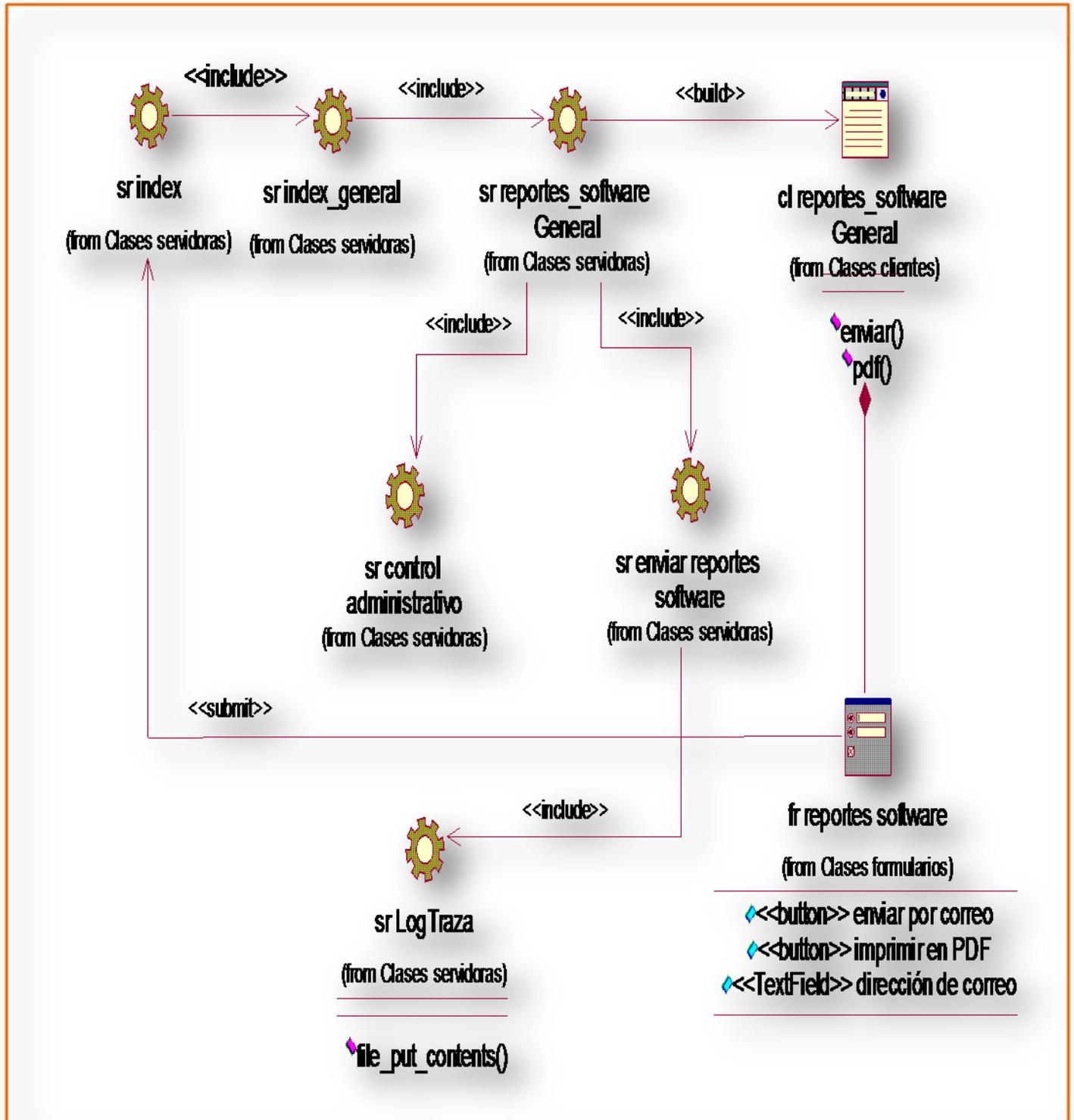


## ANEXO 2. Diagrama de clases del modelo objeto

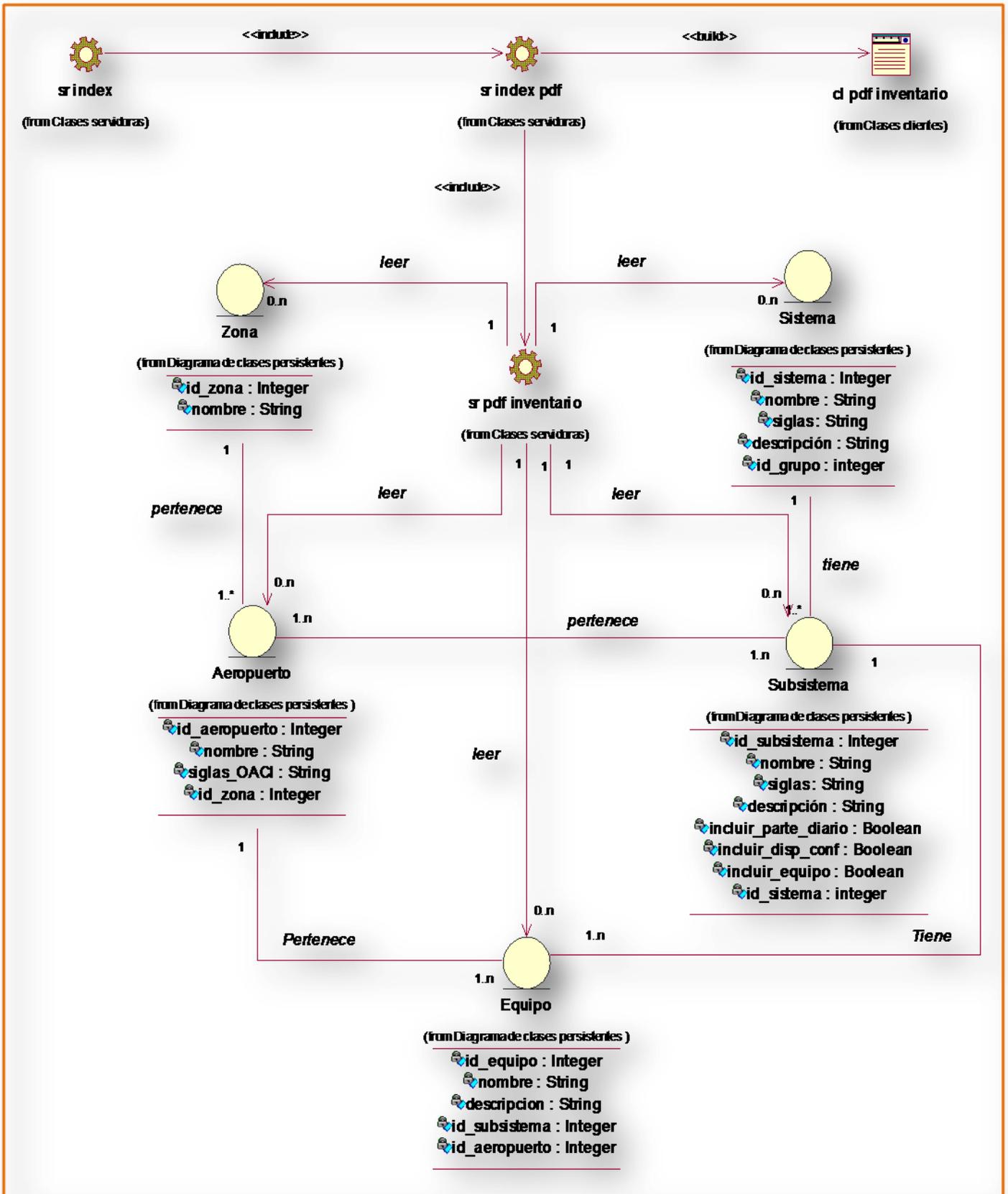


### ANEXO 3. Diagramas de clases del diseño

#### Caso de uso Enviar correo de reportes de software

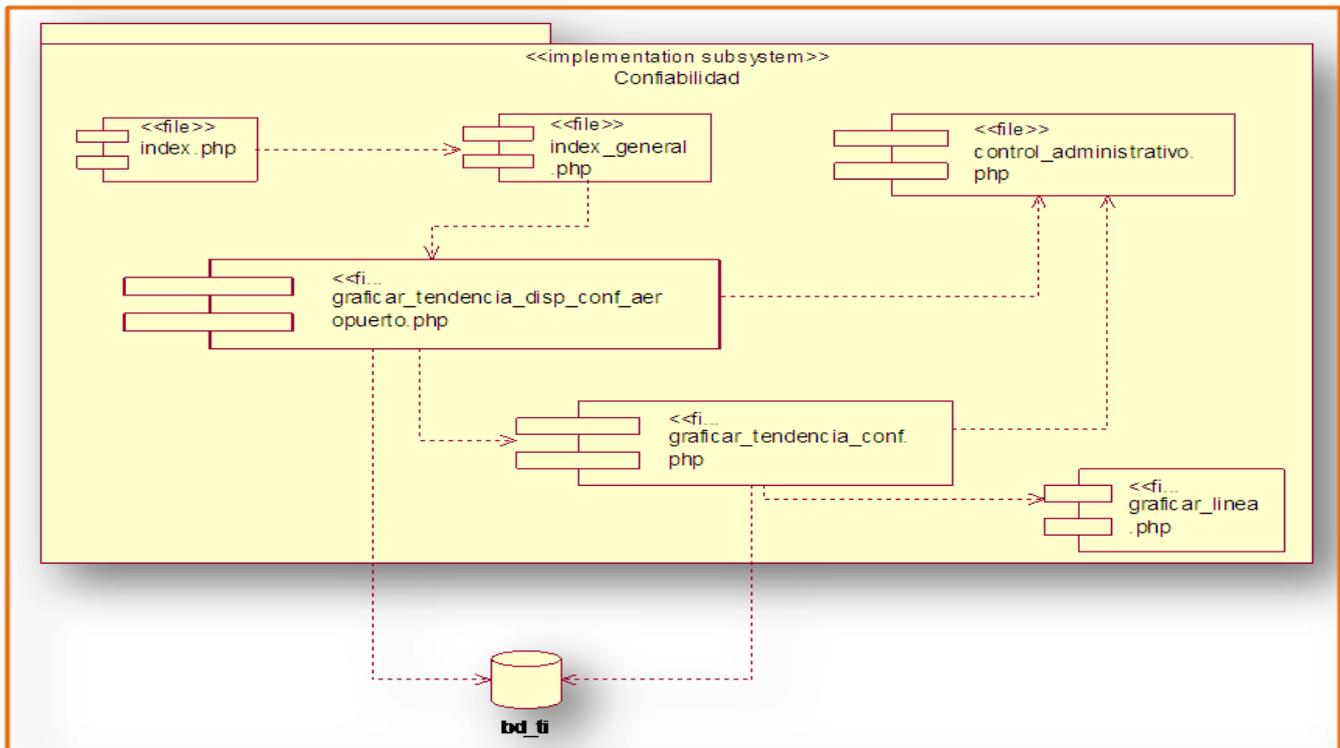


### Caso de uso Crear informe PDF de equipos existentes

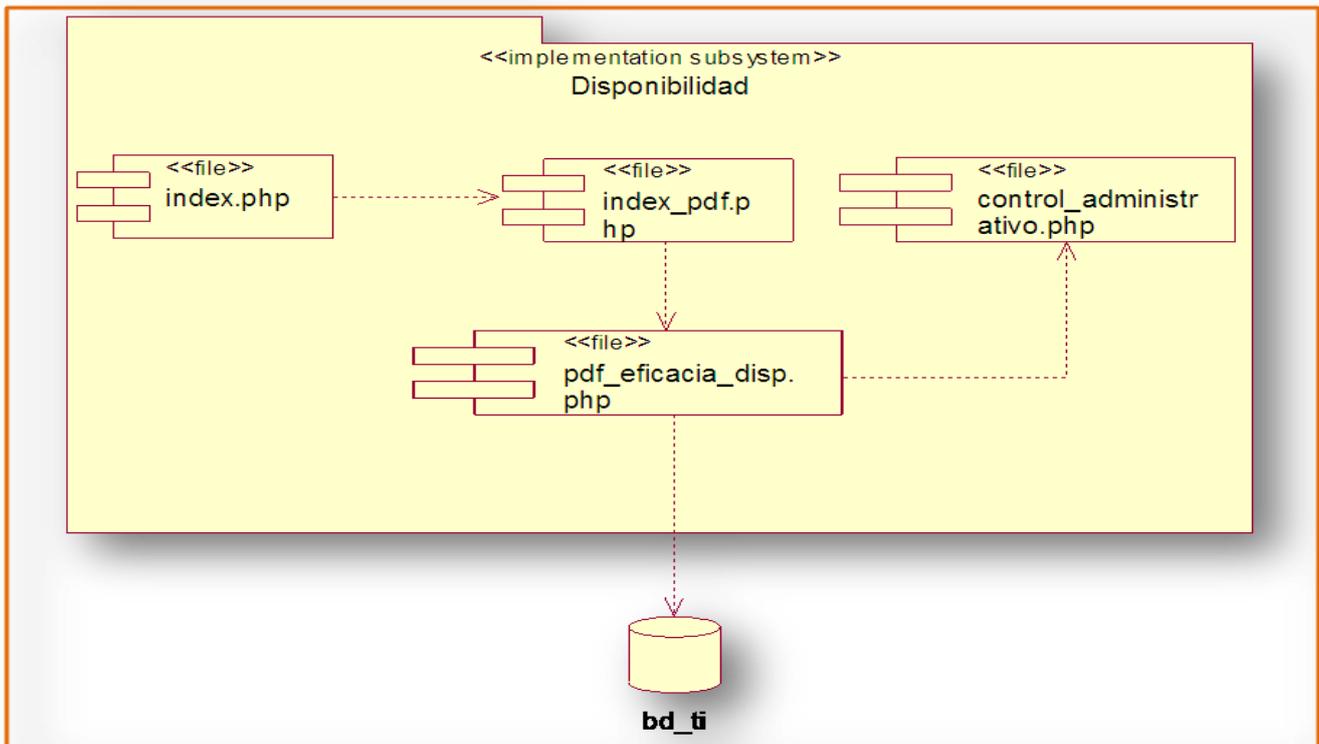


### ANEXO 4. Diagramas de componentes de implementación.

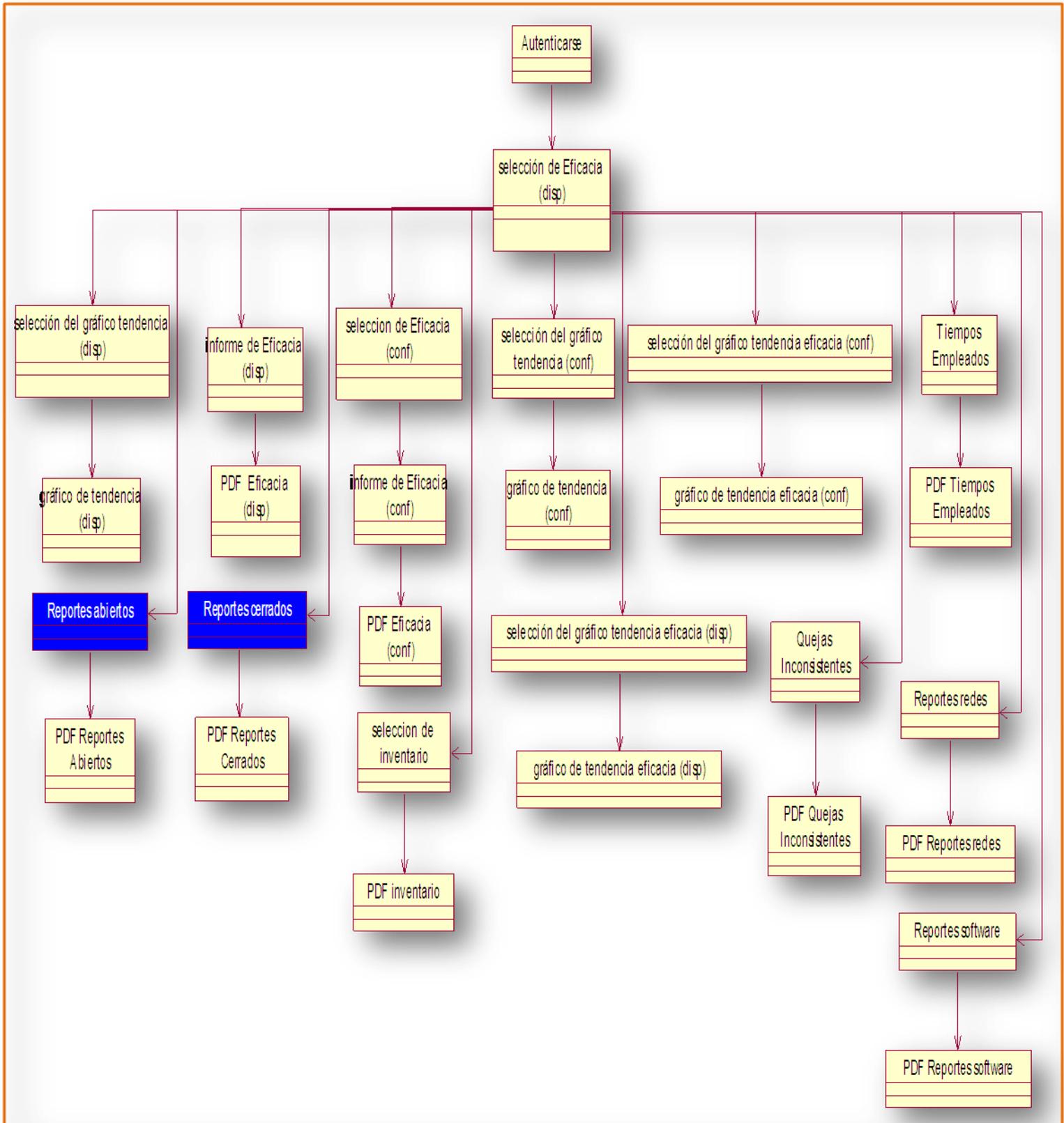
#### Caso de uso Crear gráfico de tendencia (conf) por Zona.



#### Caso de uso Crear informe PDF de eficacia (disp).



### ANEXO 5. Mapas de navegación. Administrativo:





## ANEXO 6. Entrevista realizada al cliente.

### Entrevista al Cliente

Fecha: \_\_/\_\_/\_\_\_\_

Nombre: \_\_\_\_\_ Cargo: \_\_\_\_\_

#### Puntos a analizar:

- ¿Considera que se logró aumentar la seguridad del sistema?
- ¿Considera que la nueva versión del sistema **TI** cumple todas sus expectativas?
- ¿En qué medida usted piensa que los gráficos de tendencia favorecen a la toma de decisiones?
- ¿Qué importancia le concede al indicador técnico Eficacia?

#### Respuestas del cliente:

---

---

---

---

---

---



## ANEXO 7. Cocomo II

**Multiplicadores de esfuerzos** que utiliza el COCOMO para la actividad de mantenimiento.

Multiplicador	Valor
DATA	0.90
CPLX	1
DOCU	0.91
TIME	1.0
STOR	1
PVOL	0.87
ACAP	0.85
PCAP	0.88
APEX	1
PLEX	0.91
LTEX	0.91
TOOL	0.90
PCON	0.90
SITE	0.80
RELY	1

### Factores de Escala.

Factores	Descripción	Valor
PREC	Muy diferente	4.96
FLEX	Cierta flexibilidad	3.04
RESL	Se identifican algunos de los riesgos críticos y se establecen hitos para resolverlos. Pueden presentarse algunos riesgos.	4.24
TEAM	Interacciones básicas cooperativas, objetivos y culturas de accionistas básicamente consistentes.	3.29
PMAT	Relación con el proceso de madurez del software. Nivel 1 superior.	6.24

### Valores Calibrados.

Constante	Valor
A	2.94
B	0.91
C	3.67
D	0.28