

**UNIVERSIDAD MAYOR DE SAN ANDRÉS**  
**FACULTAD DE CIENCIAS PURAS Y NATURALES**  
**CARRERA DE INFORMATICA**



**PROYECTO DE GRADO**

**SISTEMA DE CONTROL DE APERTURAS -CIERRES  
Y EFECTOS SONOROS  
PARA EL MONITOREO DE ALARMAS ELECTRONICAS**

**PARA OPTAR AL TITULO DE LICENCIATURA EN INFORMATICA  
MENCION: INGENIERIA DE SISTEMAS INFORMATICOS**

**POSTULANTE: LUIS ANTONIO HUMEREZ CHIPANA  
TUTOR: LIC. NANCY ORIHUELA SEQUEIROS  
REVISOR: LIC. JOSE LUIS ZEBALLOS ABASTO**

**LA PAZ – BOLIVIA  
2009**

*Dedico el presente proyecto de grado  
a mis padres Lorenzo y Maxima  
por el apoyo incondicional que me dieron  
en todo momento para la exitosa  
culminación de este trabajo*

## **AGRADECIMIENTOS**

A Dios por darme la sabiduría y las fuerzas necesarias para la culminación de este trabajo.

A mi docente tutor Lic. Nancy Orihuela Sequeiros por las valiosas observaciones realizadas al presente proyecto de grado que ayudaron a concluir el mismo.

Un especial agradecimiento a mi docente revisor Lic. Jose Luis Zeballos Abasto que tan desinteresadamente colaboro a que el presente proyecto pueda concluirse satisfactoriamente.

Agradezco al Dr. Peter Chavarria Castro Gerente General de la Empresa CIT Seguridad por su valiosa cooperación y confianza para el desarrollo e implementación del presente proyecto de grado.

A los docentes de la Carrera de Informática por su aporte en mi formación universitaria, y a todos mis Amigos y compañeros de estudio .

**[luis\\_humerez23@hotmail.com](mailto:luis_humerez23@hotmail.com)**

## RESUMEN

En la ciudad de La Paz - Bolivia existen varias empresas dedicadas a brindar servicio de monitoreo de Paneles de Alarma o sistemas de seguridad electrónicos; este servicio consiste en la detección remota de eventos críticos en tiempo real que ocurren en un inmueble, controlados por los dispositivos de alarma instalados en un él, que permiten monitorear a distancia el funcionamiento del panel de alarma.

Coordinar, analizar y controlar sucesos del mundo conforme ocurren, es la definición que Roger S. Pressman nos ofrece para los sistemas de tiempo real y es precisamente en esta clasificación donde podemos ubicar nuestra vista para el presente proyecto de grado, que implementado se transforma en el Sistema de Monitoreo WOLF desarrollado para la empresa CIT Seguridad, con el objetivo de automatizar todos los procesos de control de aperturas y cierres de sus clientes monitoreados, generar diferentes efectos sonoros para cada evento que ingresa a su central de monitoreo y brindar la información necesaria para dar un servicio de monitoreo de alarmas electrónicas a los clientes de la empresa.

Sistemas de tiempo real, procesos, multiprogramación y principalmente la recepción de señales por el puerto serial y el manejo de hilos (threads), son algunos de los temas que involucran el presente estudio. En el análisis y diseño, se consideraron los conceptos, métodos y características de seguimiento y evaluación, utilizando la metodología RUP, el lenguaje de modelado UML y la herramienta UML Mast para modelar los aspectos específicos de los sistemas de tiempo real.

El sistema está desarrollado en la plataforma Windows, bajo el lenguaje Visual C# 2005 Express Edition y como servidor Microsoft SQL Server 2005 Express Edition, ambos sobre Visual Studio 2005 Express, que es una herramienta libre para su uso.

Como resultado, el sistema permite la percepción de los eventos por parte del operador y su respuesta ante un evento de alarma. Además, el sistema proporciona al operador la información necesaria para comunicarse con el cliente y la elaboración de reportes.

# INDICE

	Pag.
CAPITULO I	
MARCO REFERENCIAL.....	1
1.1. INTRODUCCION.....	1
1.2. ANTECEDENTES.....	2
1.3. PRESENTACION DE LA EMPRESA.....	3
1.4. SISTEMA LOGICO ACTUAL DE LA EMPRESA CIT SEGURIDAD.....	4
1.4.1. Tipos De Cuentas.....	5
1.5. PLANTEAMIENTO DEL PROBLEMA.....	6
1.6. OBJETIVOS.....	8
1.6.1. Objetivo General.....	8
1.6.2. Objetivos Específicos.....	8
1.7. JUSTIFICACION DEL TEMA.....	9
1.7.1. Justificación Económica.....	9
1.7.2. Justificación Social.....	9
1.7.3. Justificación Técnica.....	9
1.8. ALCANCES Y LIMITES.....	9
1.9. APORTE TEORICO PRACTICO.....	10
1.10. METODOS Y HERRAMIENTAS.....	10
1.10.1. Métodos.....	10
1.10.2. Herramientas.....	11
CAPITULO II	
MARCO TEORICO.....	12
2.1. MONITOREO DE ALARMAS.....	12
2.1.1. Generación, Envío Y Recepción De Eventos.....	14
2.2. EL RECEPTOR BOSCH D6600.....	14
2.2.1. Características.....	14
2.2.2. Diagrama General Del Sistema.....	15
2.2.2.1. PSTN configuración de comunicaciones.....	15
2.2.3. Comunicaciones PSTN.....	15
2.3. EL SOFTWARE DE ADMINISTRACION D6200.....	16

2.4. PUERTOS SERIE.....	16
2.4.1. Comunicación Serie.....	17
2.4.2. Componentes Del Puerto Serie.....	17
2.4.3. Conectores Del Puerto Serie.....	17
2.5. EL ESTANDAR RS-232.....	17
2.5.1. Conexiones (Desde el DTE).....	18
2.5.2. Construcción Física.....	18
2.5.3. Transmisión de Datos.....	19
2.5.3.1. Componentes.....	19
2.5.3.2. Tasa De Bits y Tasa De Baudios.....	20
2.6. SISTEMAS DE TIEMPO REAL.....	20
2.6.1. Entorno De Un Sistema De Tiempo Real.....	21
2.6.2. Requisitos Funcionales.....	21
2.6.2.1. Adquisición de Datos.....	21
2.6.2.2. Control Digital Directo.....	22
2.6.2.3. Interacción con personas.....	22
2.6.3. Requisitos Temporales.....	22
2.6.3.1. Origen de Requisitos Temporales.....	23
2.6.3.2. Control por Computador.....	23
2.6.3.3. Relación Entre Parámetros Temporales.....	24
2.6.4. Requisitos de Fiabilidad y Seguridad.....	24
2.6.5. Clases de Sistemas de Tiempo Real.....	24
2.6.5.1. Sistemas De Tiempo Real Críticos y Acríticos.....	25
2.6.5.2. Sistemas De Tiempo Real Con Parada Segura O Con Degradación Aceptable.....	25
2.6.5.3. Sistemas de Tiempo Real Respuesta Garantizada o lo Mejor Posible.....	25
2.6.5.4. Sistemas de Tiempo Real con Recursos Adecuados o Inadecuados.....	26
2.6.5.5. Sistemas Dirigidos por Tiempo y por Sucesos.....	26
2.6.5.6. Sistemas Convencionales y Empotrados.....	26
2.6.5.7. Sistemas centralizados y distribuidos.....	27
2.7. PROCESOS Y THREADS.....	27
2.7.1. Proceso.....	27
2.7.1.1. El Modelo de los Procesos Secuenciales.....	28
2.7.1.2. Creación de Procesos.....	28

2.7.1.3. Terminación de los Procesos .....	28
2.7.1.4. Estados de los Procesos .....	28
2.7.2. Threads .....	30
2.7.2.1. El Modelo de los Threads .....	30
2.7.2.2. Utilización de los Threads .....	34
2.7.3. Implementación de los Threads en el Espacio del Usuario .....	35
2.7.4. Implementación de los Threads en el Núcleo .....	36
2.7.7. Threads Emergentes .....	37
2.8. PLANIFICACIÓN .....	38
2.8.1. Planificación en Sistemas de Tiempo Real .....	39
2.9. PROCESO UNIFICADO RUP .....	40
2.9.1. Proceso Unificado Dirigido Por Casos De Uso .....	40
2.9.2. Proceso Unificado Centrado En La Arquitectura .....	41
2.9.3. Proceso Unificado Iterativo e Incremental .....	41
2.9.4. Ciclo De Vida De R.U.P. ....	42
2.9.5. Fases R.U.P. ....	43
2.9.5.1. Fase inicio: establecer viabilidad .....	43
2.9.5.2. Fase de elaboración: factibilidad .....	43
2.9.5.3. Fase de construcción .....	44
2.9.5.4. Fase de transición: en el entorno de usuario .....	45
2.9.6. Modelos y flujos de trabajo .....	45
2.9.6.1. Modelado Del Negocio .....	46
2.9.6.2. Requisitos .....	47
2.9.6.3. Análisis .....	49
2.9.6.4. Diseño .....	50
2.9.6.5. Implementación .....	52
2.10. UML MAST METAMODEL .....	53
2.11. HERRAMIENTAS DE PRUEBAS .....	57
2.11.1. Pruebas de caja Blanca .....	57
 CAPITULO III	
MODELO TEORICO .....	59
3.1. REQUERIMIENTOS DEL SISTEMA .....	59
3.2. IDENTIFICACION DE LOS CASOS DE USO .....	59

3.2.1. Descripción de Actores.....	59
3.2.2. Descripción De Casos De Uso.....	62
3.2.2.1. Percibir y Almacenar Eventos.....	62
3.2.2.2. Percepción de la Alerta Sonora por el Operador de Central.....	63
3.2.2.3. Percepción Visual del Evento por el Operador de Central.....	64
3.2.2.4. Despliegue de la Información de la Cuenta.....	65
3.2.2.5. Controlar Horarios de Apertura y Cierre.....	66
3.2.2.6. Elaboración de Reportes.....	67
3.2.2.7. Generar Reportes.....	68
3.3. MODELO DE ANALISIS.....	69
3.3.1. Casos de Uso Análisis.....	69
3.3.2. Diagramas de Colaboración.....	72
3.3.3. Diagrama de Secuencias.....	76
3.3.4. Diseño del sistema en Tiempo Real.....	80
3.3.5. Diagrama de Clases.....	84
3.3.6. Modelado de Base de Datos.....	84
3.3.7. ARQUITECTURA DEL SISTEMA.....	90
3.3.7.1. Entorno del Sistema.....	90
3.3.7.2. Requerimientos de Hardware.....	90
3.3.7.3. Requerimientos de Software.....	92
CAPITULO IV	
CONCRECION DEL MODELO.....	93
4.1. IMPLEMENTACION.....	93
4.2. INTERFACES DE USUARIO.....	93
4.3. PRUEBAS.....	102
4.3.1. Prueba De Caja Blanca.....	102
4.4. METRICAS DE CALIDAD.....	106
4.4.1. Punto Función.....	106
CAPITULO V	
CONCLUSIONES Y RECOMENDACIONES.....	109
5.1. CONCLUSIONES.....	109
5.2. RECOMENDACIONES.....	110



REFERENCIAS BIBLIOGRAFICAS .....	111
ANEXOS	
Anexo A1: LISTA DE EVENTOS .....	i
Anexo A2: TABLA DE DEMORAS .....	v
Anexo A3: ANALISIS DE UNA TRAMA VALIDA .....	vi
Anexo A4: SIMULADOR D6600 .....	vii
Anexo A5: TABLA DE TRAMAS DE PRUEBA .....	viii

## INDICE DE FIGURAS

	Pag.
1.1 Organigrama de la empresa CIT Seguridad.....	4
1.2 Sistema Actual.....	6
2.1 EL Panel de Alarma y sus componentes.....	13
2.2 Computer Connection.....	15
2.3 Entorno de un sistema de tiempo real.....	21
2.4 Requisitos temporales.....	22
2.5 Origen de requisitos temporales.....	23
2.6 Control por computador.....	23
2.7 Sistemas centralizados y distribuidos .....	27
2.8 Estados y transiciones de un proceso .....	29
2.9 a) Tres procesos cada uno con un thread. (b) Un proceso con tres threads.....	31
2.10 Atributos propios de los procesos y threads .....	32
2.11 Cada thread tiene su propia pila .....	33
2.12 a) Un paquete de threads a nivel de usuario. b) Un paquete de threads gestionado por el núcleo.....	35
2.13 Creación de un nuevo thread cuando llega un mensaje .....	38
2.14 Proceso Iterativo e Incremental.....	42
2.15 Un ciclo con sus fases e iteraciones .....	43
2.16 Flujos de Trabajo.....	46
2.17 Modelos resultantes del flujo de trabajo modelado del negocio .....	47
2.18 Artefactos del grupo de trabajo Requisitos .....	48
2.19 Estereotipos Básicos.....	50

2.20	Subsistemas en capas de aplicación, intermedia y software del sistema...	52
2.21	Diagrama Mast RT View .....	55
3.1	Modelo de casos de uso .....	61
3.2	Modelo de Casos de Uso: Percibir y almacenar eventos .....	62
3.3	Modelo de Casos de Uso: Percepción de la alerta sonora por el Operador de Central.....	63
3.4	Modelo de Casos de Uso: Percepción visual .....	64
3.5	Modelo de Casos de Uso: Despliegue de la información de la cuenta .....	65
3.6	Modelo de Casos de Uso: Controlar horarios de apertura y cierre .....	66
3.7	Modelo de Casos de Uso: Elaboración de reportes .....	67
3.8	Modelo de Casos de Uso Generar Reportes .....	68
3.9	Diagrama de casos de uso-análisis: Percibir y almacenar eventos.....	69
3.10	Diagrama de casos de uso-análisis: Percepción de la alerta sonora por el Operador de Central .....	70
3.11	Diagrama de casos de uso-análisis: Percepción visual del evento por el Operador de Central.....	70
3.12	Diagrama de casos de uso-análisis: Despliegue de la información de la cuenta.....	71
3.13	Diagrama de casos de uso-análisis: Controlar horarios de apertura y cierre.....	71
3.14	Diagrama de casos de uso-análisis: Elaboración de reportes .....	72
3.15	Diagrama de casos de uso-análisis: Generar Reportes.....	72
3.16	Diagrama Colaboración: Percibir y almacenar eventos .....	73
3.17	Diagrama Colaboración: Percepción de la alerta sonora por el Operador de Central.....	73
3.18	Diagrama Colaboración: Percepción visual del evento por el Operador de Central.....	74
3.19	Diagrama Colaboración: Despliegue de la información de la cuenta .....	74
3.20	Diagrama Colaboración: Controlar horarios de apertura y cierre .....	75
3.21	Diagrama Colaboración: Elaboración de reportes.....	75
3.22	Diagrama Colaboración: Generar Reportes .....	76
3.23	Diagrama Secuencia: Percibir y almacenar eventos .....	76
3.24	Diagrama Secuencia: Percepción de la alerta sonora por el Operador de Central.....	77

3.25	Diagrama Secuencia: Percepción visual del evento por el Operador de Central.....	77
3.26	Diagrama Secuencia: Despliegue de la información de la cuenta.....	78
3.27	Diagrama Secuencia: Controlar horarios de apertura y cierre.....	78
3.28	Diagrama Secuencia: Elaboración de reportes.....	79
3.29	Diagrama Secuencia: Generar Reportes.....	79
3.30	Diagrama de clases y arquitectura del software en tiempo real.....	83
3.31	Diagrama de Clases.....	85
3.32	Modelo de Bases de Datos.....	86
3.33	Entorno del Sistema de Monitoreo WOLF.....	91
3.34	Descripción física de la comunicación por el puerto serial.....	92
4.1	Ventana de autenticación del Operador de la Central de Monitoreo.....	94
4.2	Ventana de Monitoreo.....	95
4.3	Ventana de Estados.....	96
4.4	Ventana Datos de la Cuenta.....	97
4.5	Ventana de autenticación del Administrador del Sistema.....	98
4.6	Ventana Principal de Administración del Sistema.....	98
4.7	Ventana Adicionar Nueva Cuenta.....	99
4.8	Ventana Registrar Zonas.....	100
4.9	Ventana Adicionar Usuario.....	100
4.10	Ventana Registrar, Modificar y Eliminar Operadores de Central.....	101
4.11	Ventana Generar Reportes.....	102
4.12	Grafo de Flujo.....	105

## INDICE DE TABLAS

	Pag.	
2.1	Señales RS-232.....	18
2.2	Formato de especificación de casos de uso.....	48
3.1	Requerimiento de Hardware.....	90
3.2	Requerimiento de Software.....	92
4.1	Calculo de punto función.....	107
4.2	Tabla de valores de ajuste de complejidad.....	108
4.3	Valores de ajuste de la complejidad.....	108

# CAPITULO I

## MARCO REFERENCIAL

### 1.1. INTRODUCCION

La seguridad en todos sus ámbitos es un tema que nos preocupa , esto se refleja en la necesidad de contratar los servicios de empresas dedicadas a prestar servicios de seguridad para nuestros hogares, negocios y nuestra propia integridad, lo que ha generado en los últimos años el surgimiento de empresas dedicadas a brindar estos servicios. Entre estas empresas podemos encontrar: Empresas de Recursos Humanos de Seguridad (Guardias), Asesores en Materias de Seguridad Privada, Capacitadores en Seguridad, Transporte de Valores, Suministro de Tecnologías y las de Monitoreo de Alarmas [Leemira,2007].

En el presente proyecto vamos a dirigir nuestra atención a las empresas que prestan el servicio de monitoreo de alarmas, que utilizan sistemas de Paneles de Alarma capaces de enviar señales a través de uno o más medios de comunicación (línea telefónica, radio frecuencias, Internet, etc.). Todas las empresas que desean prestar este servicio cuentan por lo menos con una receptora de señales, un equipo de computación y un software que hace la interfaz final entre el sistema de seguridad y el operador de central, patrullas y patrulleros para asistir al lugar del evento y personal competente para la instalación, configuración y mantenimiento de los paneles de alarma monitoreados.

Las empresa CIT Seguridad dedicada a brindar el servicio de monitoreo de sistemas de seguridad establecida en la ciudad de La Paz realiza este servicio utilizando para ello paneles de alarma capaces de comunicarse vía línea telefónica con una central de monitoreo.

El presente proyecto realiza una investigación aplicada hacia el monitoreo de alarmas electrónicas, para evaluar y corregir las falencias relacionadas con los procesos manuales y repetitivos dentro de este servicio.

## 1.2. ANTECEDENTES

Durante muchos años en muchos países se ha utilizado el vínculo telefónico análogo como la forma más sencilla y plural de monitorear alarmas a distancia. Hacia fines de los 90, esta forma de comunicación de alarmas alcanzo el 95 % del mercado de monitoreo como vinculo primario, quedando el 5 % restante conformado por el monitoreo radial VHF o UHF y el monitoreo por telefonía celular [Softguard,2007]. En Bolivia, mas específicamente en la ciudad de La Paz, las empresas de seguridad aun siguen utilizando la línea telefónica como medio principal de comunicación entre el Panel de Alarma y la central de monitoreo y en casos especiales cuando el cliente puede y quiere pagar se utiliza una comunicación vía celular como respaldo de comunicación.

Tomando en cuenta sistemas de control de personal o de suscriptores que son de propósito similar al monitoreo de alarmas se encuentra el “Sistema de Contabilidad y Control de Suscriptores” de la carrera de informática, que tiene como uno de sus objetivos reducir el tiempo de espera de los clientes en el sistema al momento de suscribirse y realizar reclamos, seguimiento de suscriptores y encargados y entrega de suscripciones, que fue desarrollado bajo la metodología SSADM, con la ayuda de herramientas ADO [Quisbert,2000].

Continuando con la rama de control de personal tenemos el “Sistema de Control de Asistencia Para Personal Caso: División de Acciones y control – UMSA” de la carrera de informática, que tiene como objetivo principal: Implantar un sistema de control de asistencia para el personal administrativo de la UMSA mediante dispositivos biométricos, que permita administrar desde las oficinas centrales de la división de acciones y control de información generada en los diferentes predios universitarios y así disponer de información oportuna y confiable, dicho proyecto fue desarrollado bajo la metodología RUP [Duran,2007].

Otro sistema que tiene un propósito similar es el “Sistema de Información Biométrico para el Control de Personal de la Empresa Control Total S.R.L.”, que se caracteriza por usar la tecnología de lectura de huellas dactilares para llevar a cabo un control adecuado del personal de la empresa, realizado con RUP y UML [Ormachea,2004].

Actualmente se ha desarrollado software para el monitoreo de sistemas electrónicos de seguridad como el siguiente:



**Winsamm.-** Es un sistema que posee procesamiento de eventos directamente del receptor y se puede enlazar con una o más tipos de receptora en forma simultanea. Admite puertos Com1 a Com8, múltiples formatos, tiene protección ante su uso no autorizado y una base de datos flexible, además de eso es multiusuario de comentarios pre definidos. A continuación se muestran algunas de sus características:

- Lista de códigos de alarma modificable por cuenta.
- Soporta respaldo diario automático.
- Tiempo de proceso de operador.
- Teclas de método abreviado redundantes con mouse.
- Cambio de operador sencillo.
- Sonidos seleccionables para Apertura/Cierre con período de gracias.
- Estado de Armado/Desarmado todo el tiempo disponible.
- Señales entrantes puestas en fila según esquemas de prioridades.
- Períodos de test flexibles: 1...999 en horas con opción de demora.
- Notificación de reporte de test no enviado.
- Importar/Ver hasta 12 imágenes en JPG para cada cuenta.
- Manejado por el motor postgresSQL [winsam,9548].

### **1.3. PRESENTACION DE LA EMPRESA**

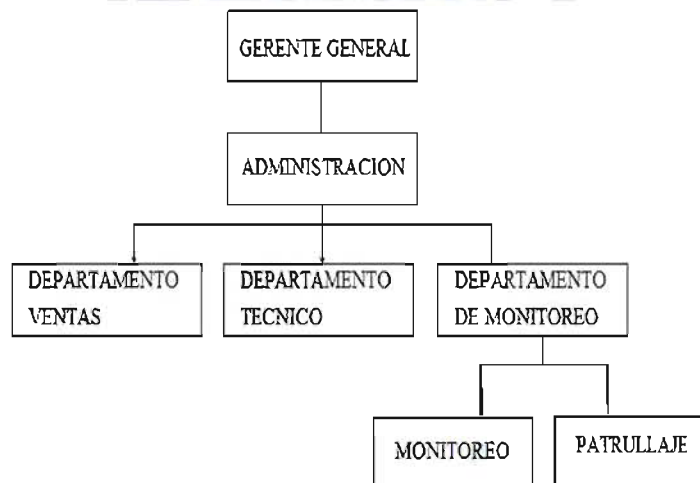
CIT Seguridad es una empresa de servicios de seguridad, su principal función es de brindar el servicio de monitoreo a sistemas electrónicos de seguridad instalados en inmuebles como ser casas, oficinas, entidades financieras, industrias, etc. CIT Seguridad también se dedica a instalar una variedad de sistemas de seguridad, entre los que podemos mencionar alarmas electrónicas, circuito cerrado de televisión (CCTV), cercas electrificadas, portones automatizados, porteros eléctricos, etc.

CIT Seguridad brinda un sistema de monitoreo que esta conectado a través de líneas telefónicas, cada evento de alarma o intrusión es comunicado a su central de monitoreo las 24 horas del día durante los 365 días del año.

Los servicios que ofrece CIT Seguridad son los siguientes:

- Servicio de respuesta inmediata las 24 horas con patrullas propias
- Línea de atención al cliente las 24 horas
- Monitoreo del sistema de alarma las 24 horas del día y 365 días del año
- Monitoreo permanente de corte de energía eléctrica
- Monitoreo permanente de estado de voltaje de la batería
- Monitoreo permanente de corte de línea telefónica
- Monitoreo de control de aperturas y cierres
- Monitoreo de control de usuarios que realizan cierres y aperturas
- Control de horarios de aperturas y cierres
- Hasta 10 usuarios con códigos individuales.

Para la mejor comprensión de la estructura orgánica de la empresa se presenta el siguiente organigrama:



**Fig. 1.1. Organigrama de la Empresa CIT Seguridad**  
Fuente: Elaboración propia

#### 1.4. SISTEMA LOGICO ACTUAL DE LA EMPRESA CIT SEGURIDAD

El servicio de monitoreo de la empresa CIT Seguridad depende de un equipo receptor de eventos (**Receptor D6600**), un Software de Administración para el Receptor D6600 (**D6200**) (detallados en el Capítulo II) y un Software Decodificador (**D6600**).

El Software **D6600** es un sistema que decodifica los eventos enviados por el Software de Administración **D6200**. El **D6600** identifica el número de cuenta, evento y estado, y luego, realiza una búsqueda en la base de datos de la cual extrae el nombre del cliente o razón social, posteriormente despliega en la pantalla todos los datos mencionados en una tabla bidimensional. Este sistema depende del anterior sistema. Si el **D6200** no funciona de forma correcta o no está en funcionamiento, el **D6600** no puede iniciarse.

Una vez que el D6600 muestra los datos del evento en pantalla el Operador debe realizar una acción de acuerdo con el tipo de evento desplegado en pantalla.

#### 1.4.1. Tipos De Cuentas

Dentro de la empresa están definidos tipos de cuentas, que se caracterizan por los servicios adicionales que reciben los clientes, dependiendo del uso que se le da al inmueble monitoreado. Los tipos de cuentas son tres y se detallan a continuación:

**Tipo 1.-** Son todos aquellos inmuebles que no tienen horario de trabajo y no necesitan que se les controle el horario en que realizan el armado y desarmado (apertura y cierre) <sup>1</sup> del panel de alarma. En este tipo de cuentas se encuentran los domicilios particulares (DM).

**Tipo 2.-** Son todos aquellos inmuebles que tienen un horario de trabajo irregular, pero deben estar protegidos la mayor parte del día, sobre todo en la noche. En este tipo de cuentas podemos encontrar: Depósitos y Hoteles.

**Tipo 3.-** Son todos aquellos inmuebles que tienen un horario de trabajo bien establecido. Entre estos podemos encontrar los siguientes: Tiendas (DMT), Oficinas (DMO), Bancos, Entidades Financieras, Laboratorios, Industrias, etc. Estos inmuebles deben tener un control estricto de sus horarios de apertura y cierre con un margen de error de 30 minutos en su horario de armado o cierre; pasados estos 30 minutos el operador deberá proceder con su procedimiento de respuesta para demoras en el cierre.

---

<sup>1</sup> En el ámbito de las empresas de monitoreo de sistemas de seguridad **apertura** se define como el inicio de la jornada laboral en el caso de las cuentas tipo 2 y 3 e Ingreso al domicilio en las cuentas del tipo 1 que llega a la central de monitoreo como desarmado del panel de alarma (Open). De la misma forma **cierre** se define como el final de la jornada laboral o la acción de salir del domicilio que llega a la central de monitoreo como armado del panel de alarma (Close).



## 1.5. PLANTEAMIENTO DEL PROBLEMA

El sistema que actualmente esta en funcionamiento dentro de la empresa CIT Seguridad no cuenta con documentación alguna de su diseño y sus características, tampoco cuenta con documentación acerca de la base de datos y su estructura, por lo tanto no es posible corregir, aumentar o integrar nuevos módulos, de tal manera que cumpla con las exigencias actuales de la empresa como ser el control de aperturas y cierres y la generación de alertas sonoras por cada tipo de evento.

Además se puede observar que, la información mostrada por cada evento no es suficiente para poder comunicarse con el cliente, si observamos el siguiente gráfico podemos observar en detalle los datos que visualiza el operador cuando un evento llega a la central de monitoreo.

Hora	Cliente	Cuenta	Evento	Zona	Lugar	Descripción
19:04:44	XXXXXXXXXXXXXXXX	7279	Open	002	No especificado	USUARIO
19:03:37	XXXXXXXXXXXXXXXX	7172	Close	000	No especificado	ARMADO DE BÓVEDA
19:00:45	XXXXXXXXXXXXXXXX	0164	Open	003	VM. (1er P. Post Pago)	USUARIO
18:55:11	XXXXXXXXXXXXXXXX	7188	Open	002	Piscina MV.	USUARIO
18:46:49	XXXXXXXXXXXXXXXX	7212	Open	001	Oficina	USUARIO
18:44:56	XXXXXXXXXXXXXXXX	7316	Open	000	No especificado	TEST PERIÓDICO
18:44:31	XXXXXXXXXXXXXXXX	7168	Close	001	Escritorio BP	USUARIO

**Fig. 1.2. Sistema Actual**  
**Fuente: Sistema D6600**

Los campos mostrados en la Fig. 1.2 se describen ha continuación:

**Hora.-** Representa la hora en la que el evento fue registrado por el sistema.

**Cliente.-** Contiene el nombre o razón social y en algunos casos también contiene la dirección del inmueble.

**Cuenta.-** Representa el número de cuenta único e inequívoco.

**Evento.-** Representa un adjetivo calificativo del evento (Open, Close o Previous).

- **Open** Indica que es un evento de cambio de estado inicial. Por ejemplo si llega se un evento de Usuario (ANEXO A1) con Open esto significa que la alarma ha sido desamada o desactivada.

- **Close** Indica que es un evento de restauración. Continuando al ejemplo anterior, la llegada de un evento de Usuario con Close, indica que el panel de alarma esta en un estado de armado.
- **Previous** no esta definido.

**Zona.-** Representa la zona en la que se produjo el evento de ROBO O TAMPER y en el caso de que el evento sea USUARIO O ARMADO INTERNO este campo representa el número de usuario que realizo la acción de armar o desarmar el sistema.

**Lugar.-** En este campo se puede visualizar la zona donde se produjo la alarma y los sensores instalados en la zona.

**Descripción.-** Este campo muestra el tipo de evento (ANEXO A1).

La información faltante por cada evento desplegado en la figura 1.1 es la siguiente:

**Usuario.-** nombre del usuario que realizo la acción de APERTURA y CIERRE .

**Teléfonos.-** En caso de un evento de ROBO se debería desplegar una lista con todos los nombres y teléfonos de las personas autorizadas o elegidas para hacer uso del sistema, colaborar con el patrullero al momento de llegar al inmueble y que pueden colaborarnos en caso de la ausencia del cliente o encargado.

**Estado.-** El estado actual de todas las cuentas monitoreadas.

Por lo tanto podemos decir que los reportes generados por el D6600 no cuentan con los datos necesarios o apropiados, como ser nombre del usuario, números telefónicos y estado actual de cuentas, para que el operador pueda realizar su trabajo sin dejar de consultar documentos en Excel, carpetas de filiación y Tablas de demoras impresas (control de apertura y cierre).

El error mas frecuente esta relacionado con el control de apertura y cierre de las cuentas de tipo 3, que corresponden a: Tiendas, Entidades Financieras e Industrias, este control lo realiza el operador de forma manual marcando el número de cuenta del cliente en una tabla impresa el horario de ingreso o salida y en la tabla de demoras (ANEXO A2), cuidando que estos dos eventos no se realizaron fuera de horario; pero el operador puede olvidar marcar una cuenta o marcar equivocadamente en la tabla de control.

Otro error frecuente y el más perjudicial es cuando el operador deja pasar una señal de ROBO, Pánico del teclado, Falla de batería, etc. Este error generalmente se da durante los turnos de madrugada, tiempo en el cual el operador puede estar dormido o dormitando. Este error se puede asociar al sistema actual pues el sistema emite un solo tipo de sonido para cualquier evento entrante; entonces el operador puede llegar a confundir un evento de ROBO con cualquier otro evento que no necesita la intervención del patrullero o algún servicio de emergencia.

Teniendo en cuenta todo lo anterior podemos indicar, que el operador puede cometer un error debido a que los procesos que se realizan como respuesta a eventos que están fuera de su rango de llegada (eventos 400, 401 y 441 en cuentas del tipo 2 y 3), eventos de ROBO e intrusiones al sistemas de seguridad (eventos 130, 120, 137, 121 y 123 (ANEXO A1)), o alertas de la alarma por la falta de uno o mas recursos del sistema de seguridad electrónico (eventos 301,302 y 351 (ANEXO A1)). Son propensos a fallas humanas como ser: descuido al realizar procesos repetitivos, descuido por cansancio y negligencia del operador.

## **1.6. OBJETIVOS**

### **1.6.1. Objetivo General**

Implementar un nuevo sistema de monitoreo (de alarmas) para automatizar todos los procesos de control de apertura y cierre de las cuentas del tipo 2 y 3, generar diferentes efectos sonoros para cada evento que ingresa a la central de monitoreo y brindar al operador la información necesaria para dar un servicio de monitoreo de alarmas electrónicas a los clientes de la empresa CIT Seguridad.

### **1.6.2. Objetivos Específicos**

- Desplegar toda la información necesaria para comunicarse con el cliente.
- Generar alertas sonoras únicas para cada tipo de evento entrante.
- Realizar el control automático de apertura y cierre de las cuentas.
- Facilitar la búsqueda de información de cada cliente.

## **1.7. JUSTIFICACION DEL TEMA**

### **1. 7.1. Justificación Económica**

En la empresa CIT Seguridad con el sistema actual se produjeron pérdidas de clientes debidos principalmente a que no se atendieron de forma correcta y a tiempo eventos de ROBO y control de aperturas y cierres. El proyecto pretende disminuir estos errores relacionados con la información que muestra el sistema y la emisión de tonos por cada evento que se registra. De esta manera la empresa se beneficiará en forma económica puesto que se disminuirán las pérdidas por dichos eventos que no fueron atendidos de forma correcta y oportuna.

### **1. 7. 2. Justificación Social**

Debido al crecimiento del número de clientes, los procedimientos manuales se hacen tediosos y están propensos a errores de parte del personal de monitoreo; con el proyecto se proporcionara una mejor atención a todos ellos y también se automatizara la función de control de aperturas y cierres que actualmente el operador realiza manualmente; y así ellos tendrán mas tiempo para realizar sus reportes e informes.

### **1. 7. 3. Justificación Técnica**

La empresa cuenta con recursos tecnológicos, materiales y equipos computacionales los cuales serán aprovechados para la elaboración del sistema; así como para su implementación.

## **1. 8. ALCANCES Y LIMITES**

Los alcances del presente proyecto están delimitados por la función que se ha definido para la central de monitoreo, la cual es el de brindar el servicio de monitoreo de alarmas electrónicas para los clientes de la empresa CIT Seguridad. El sistema cubrirá todas cuentas monitoreadas ubicadas en la ciudad de La Paz que sean del tipo 1, 2 y 3, desplegará toda la

información necesaria para comunicarse con el cliente , capturaré los eventos detallados en el Anexo A1 para generar alertas sonoras únicas para cada tipo de evento. Además de realizar el control automático de apertura y cierre de las cuentas.

El sistema estará limitado por los siguientes puntos:

- El sistema no esta orientado a realizar los reportes que se envían semanalmente a cada cliente con relación a los eventos registrados de cada cuenta.
- El sistema no se comunicara con el cliente solo desplegara la información necesaria para realizar dicha acción.
- El sistema trabajara únicamente con los pulsos 4+2 del Contact ID.
- El sistema será un complemento al sistema D6200 . No lo reemplazará.

## **1. 9. APORTE TEORICO PRACTICO**

En cuanto al producto software, el proyecto ofrece a la empresa un sistema que contribuye al desempeño de la atención que brinda la empresa a sus clientes, permitiendo de esta manera contar con información fiable y oportuna y las alertas correspondientes para cada evento que percibe la central de monitoreo.

El aporte más significativo será el estudio de sistemas de tiempo real que coordinan/analizan/controlan sucesos del mundo conforme ocurren. Por lo cual para la recepción de los eventos se utilizarán hilos (threads). Al mismo tiempo, para desplegar el evento en pantalla, se generarán alertas sonoras y visuales llamativas por cada evento que necesite la ejecución de un procedimiento de respuesta por parte del operador de la central de monitoreo.

## **1.10. METODOS Y HERRAMIENTAS**

### **1.10.1. Métodos**

- Método científico para el proceso de investigación: El cual permite hacer una investigación explorativa y descriptiva de la institución, indispensable para realizar la ingeniería de requerimientos.

- Metodología de desarrollo de software, Proceso Unificado RUP basado en UML: Metodología que permite hacer un proceso de desarrollo iterativo, evolutivo e incremental.

### 1.10.2. Herramientas

- Herramientas para el desarrollo de sistemas: Microsoft.net Express Edition que es una plataforma de desarrollo y ejecución de aplicaciones . En el presente proyecto se trabajara bajo el lenguaje C# y Microsoft SQL Server 2005 Express Edition.
- Herramienta CASE para el desarrollo de software en el proceso de diseño: Poseidon que es una herramienta CASE que trabaja con UML .





## CAPITULO II

### MARCO TEORICO

#### 2.1. MONITOREO DE ALARMAS

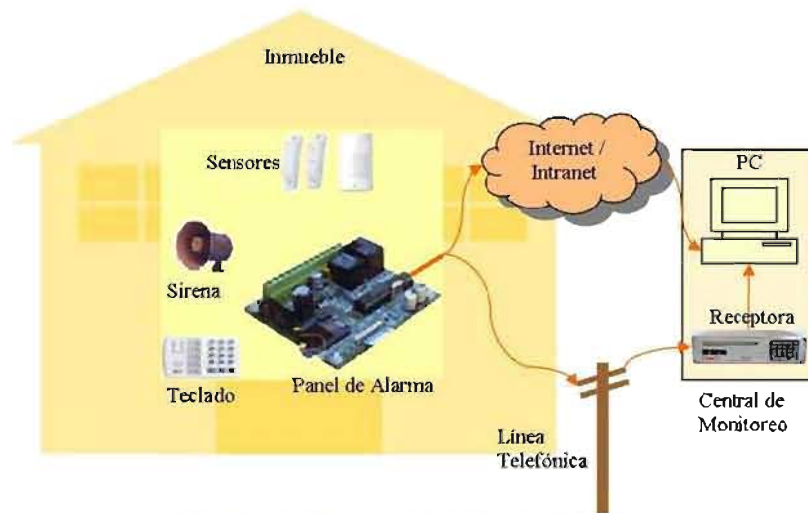
La palabra Monitor se define como Dispositivo visual o acústico o bien conjunto de instrumentos que permiten controlar a distancia el funcionamiento de un aparato o instalación y que le da la alarma cuando alguna de las magnitudes medidas rebasan los límites inferiores o superiores previamente fijados. Y Alarma se define como los aparatos (timbres, sirenas, señales luminosas, emisiones hertzianas, etc.) que pueden ser de accionamiento manual o automático. Contra el robo y la intrusión de personas en lugares prohibidos [Larouse,1987], por tanto podemos definir Monitoreo de Alarmas como la detección remota de eventos críticos, generados por los dispositivos de alarma, que permiten controlar a distancia el funcionamiento de un aparato o instalación.

Este servicio consiste en la colocación de un dispositivo electrónico de alarma en el inmueble, propiedad del usuario o razón social, y en el control remoto de la actividad de dicha alarma, desde, Una Central de Monitoreo [Dinarte,2004]; con la capacidad de filtrar, analizar y responder a dichos eventos.

**El dispositivo electrónico** o Panel de alarma consiste de un panel de alarma y sus respectivos componentes (teclado, batería de respaldo, sirena) y sensores, estos sistemas permiten la comunicación entre el sitio protegido y la Central de monitoreo, a través de varias modalidades de comunicación como ser: línea telefónica, radio frecuencia, GSM/GPRS e IP (ver Fig. 2.1.).

**La Central de Monitoreo**, se entiende por central de monitoreo al establecimiento y los equipos que permitan a la empresa que brinda el servicio, el control remoto del estado del Panel de Alarma y sus componentes y sensores; la recepción y registro de cualquier señal producida por el accionar de dichos instrumentos, así como los instrumentos necesarios para trasladar el llamado de auxilio electrónico [Dinarte,2004] a quienes el cliente determine, ya sean: policía, servicios de emergencia, familiares, empleados, etc.; para que puedan colaborar con los procedimientos de repuesta ante verificación de una falsa alarma y acción

en caso de intrusiones confirmadas, además, de la posibilidad de enviar una patrulla de la empresa para verificar el evento.



**Fig. 2.1. EL Panel de Alarma y sus componentes**  
Fuente: Elaboración propia

Las empresas dedicadas a brindar el servicio de Monitoreo de Sistemas de Seguridad Electrónicos, además de contar con personal administrativo, cuentan con el siguiente personal especializado:

- 1) **Operadores De Central:** Son los receptores “directos” del sistema de monitoreo quienes tienen la responsabilidad de ejecutar los procedimientos de comunicación con los patrulleros, clientes, servicios de emergencia; según el tipo de alarma activada.
- 2) **Los Instaladores De Dispositivos:** Son los responsables de realizar la instalación y configuración de los Paneles de Alarma en los inmuebles que desean contar con el servicio de monitoreo.
- 3) **Patrulleros:** Son los encargados de realizar la verificación de eventos e informar al Operador de Central el estado externo y/o interno del inmueble y todos sus dispositivos de seguridad.



### **2.1.1. Generación, Envío Y Recepción De Eventos**

Los eventos son generados por los Paneles de Alarma cuando estos perciben un cambio de estado en sus sensores, un cambio de estado de partición o cuando deja de percibir un recurso vital para su buen funcionamiento (La lista de todos los eventos generados por un Panel de Alarma esta detallada en el Anexo A1). Se envían los eventos desde el lugar protegido al Receptor bajo la topología de la conmutación de circuitos, uniendo origen y destino de la llamada en una comunicación punto a punto de audio bidireccional. Sobre ese canal de audio, se trafican los pulsos (3+1, 4+2, otros), los tonos multifrecuentes (DTMF) típicos del Contact Id o las tramas de MODEM FSK típicas de los formatos SIA, MODEM IIe y otros [Softguard,2007]. El Receptor recibe el evento y lo envía a un Software de Administración instalado en la PC de la Central de Monitoreo.

### **2.2. EL RECEPTOR BOSCH D6600**

El receptor de comunicaciones D6600 con componentes de hardware NetCom funciona en las siguientes configuraciones:

- Comunicaciones de línea telefónica con PSTN.
- Comunicaciones en internet o intranet vía LAN o WAN utilizando protocolos estándar de usuario y protocolo de internet (UDP/IP).
- PSTN simultánea y comunicaciones vía LAN o WAN. [0412 sp].

#### **2.2.1. Características**

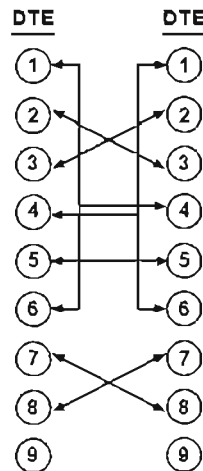
- Comunicaciones vía red y telefonía pública.
- Recibe hasta 32 líneas telefónicas simultáneamente.
- Hasta 3200 cuentas para comunicaciones de red de área local (LAN) o red de área extendida (WAN).
- Placas de protección de cuatro líneas telefónicas para aislación y supresión transitoria.
- Acceso a placa de CPU y placas de línea.
- Placa de líneas intercambiables.
- Two-way Audio (doble sentido/ transmisor/ receptor) [0412 sp].

## 2.2.2. Diagrama General Del Sistema

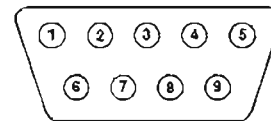
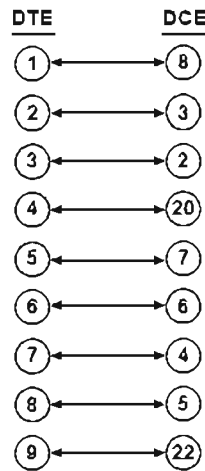
### 2.2.2.1. PSTN Configuración De Comunicaciones

Esta configuración utiliza líneas telefónicas existentes. Una señal del panel transmite por una línea telefónica analógica PSTN al receptor de comunicaciones D6600. El D6600 traduce la señal a un formato de datos común y envía la señal a través de RS-232 al sistema de automatización de la central (ver Fig. 2.2.) [0412 sp].

When connecting data terminal equipment together (i.e. D6600 COM port to PC serial port), a null modem cable **must** be used. One possible configuration for a null modem cable is:



When connecting data terminal equipment to data communication equipment (i.e. D6600 COM port to modem), a modem cable should be used (i.e. straight-through cable):



1 = DCD  
2 = RX  
3 = TX  
4 = DTR  
5 = SIG GND  
6 = DSR  
7 = RTS  
8 = CTS  
9 = RI

Fig. 2.2. D6600 – Computer connection  
Fuente [4998122703C]

### 2.2.3. Comunicaciones PSTN

- El sistema de Identificación Numérica Automática (ANI) identifica los números telefónicos entrantes.
- El Servicio de Identificación de Número Discado (DNIS) regula los ajustes de tarjeta de línea de acuerdo al número telefónico discado por el panel.
- Cada placa de línea tiene 4 líneas telefónicas.
- Recibe señales de hasta 32 líneas simultáneamente.
- Ajustes configurables e independientes en el formato de comunicación para cada línea telefónica.

- El procesamiento de alarma de Identificación de Llamadas ( Caller ID) almacena los formatos handshake correspondientes. El procesamiento automáticamente rellama el formato cuando la misma identificación de llamada es procesada.
- Indicadores LED's para condiciones de estado en línea y falla en línea.
- Requiere la placa de línea telefónica D6640 o D6641 [0412 sp].

### **2.3. EL SOFTWARE DE ADMINISTRACION D6200**

El Software de Administración D6200 es una aplicación basada en computadoras personales para Microsoft® Windows® 98, Windows 98 Segunda Edición, Windows Me, Windows NT®, Windows 2000, y Windows XP que el usuario debe ver, cambiar, mover y descargar todo de los parámetros de programación del D6600 vía conexión de Red o el puerto serial RS-232. Dentro del Software D6200 el usuario puede configurar la CPU y las líneas de parámetros de las tarjetas, viendo el estado de todos los abonados o clientes en las Bases de Datos, adicionar/editar/eliminar abonados o clientes, y configurar las operaciones de Red [4998154991C D6200 O&I Guide].

### **2.4. PUERTOS SERIE**

La interfaz serie acepta cualquier dispositivo que necesite una conexión bidireccional (quiere decir que los datos viajan desde y hacia el dispositivo periférico) y asíncrona (quiere decir que no existe una señal de tiempo u otra disposición predeterminada para las comunicaciones entre el equipo y el dispositivo) con el PC; tales como las impresoras, los dispositivos de señalamiento e incluso otras computadoras. Tanto la computadora como el periférico pueden transmitir datos siempre que lo necesiten.

El termino puerto serie quiere decir que el equipo y el dispositivo conectado se comunican transmitiendo un bit cada vez, a diferencia de otras interfaces, tales como el puerto paralelo y el bus del sistema, que pueden transmitir 8, 16 o 32 bits a la vez. Las comunicaciones serie se dividen en bytes, que para los PC son de 8 bits. Sin embargo, en algunos casos, los PC pueden comunicarse con otros tipos de computadoras que utilizan bytes de 7 bits [Zacker-Rourke,2001].

### **2.4.1. Comunicación Serie**

Para que el equipo y el periférico se comuniquen asincrónicamente a través de la conexión serie, cada uno tiene que poder determinar cuándo ha comenzado a transmitir el otro un byte. Es posible que los dispositivos interpreten erróneamente el ruido en las conexiones como datos, por lo que para indicar el inicio de un byte el dispositivo que transmite genera un bit de inicio, que es un bit de un valor. Luego el dispositivo transmite el byte seguido de un bit final. El dispositivo receptor trata como dato las señales recibidas entre el bit de inicio y el bit de final e ignora todo lo que esté comprendido entre el bit final y el próximo bit de inicio.

Este mecanismo permite que tanto el equipo como el dispositivo conectado puedan transmitir datos en cualquier momento, pero también reduce la eficacia global de la interfaz. Dos bits de cada diez (suponiendo los bytes de 8 bits) están dedicados al control de tráfico, lo que significa un 20 por 100 del tráfico total [Zacker-Rourke, 2001].

### **2.4.2. Componentes Del Puerto Serie**

Un PC típico tiene dos puertos serie (denominados también puertos COM) integrados en el súper circuito controlador de entrada/salida de la placa base. Se trata del mismo circuito que contiene el controlador para la unidad de disquetes y para las interfaces del puerto paralelo.

### **2.4.3. Conectores Del Puerto Serie**

Para poder conectar los dispositivos al puerto serie, la placa base del equipo dispone de dos conectores que se sitúan en los puertos físicos montados en la parte trasera del equipo. Los PC actuales utilizan un conector tipo D de 9 pines, denominado DB-9 y un conector DB-25. El DB-25 utiliza el mismo tipo de capsula que el DB-9, excepto que es algo mayor y que tiene 25 pines.

## **2.5. EL ESTANDAR RS-232**

Electronic Industries Alliance RS-232C es una interfaz que designa una norma para el intercambio serie de datos binarios entre un DTE (Equipo Terminal de Datos) y un DCE

(Equipo de Comunicación de datos), aunque existen otras situaciones en las que también se utiliza la interfaz RS-232.

En particular, existen ocasiones en que interesa conectar otro tipo de equipamientos, como pueden ser computadores. Evidentemente, en el caso de interconexión entre los mismos, se requerirá la conexión de un DTE (*Data Terminal Equipment*) con otro DTE.

El RS-232 consiste en un conector tipo DB-25 (de 25 pines), aunque es normal encontrar la versión de 9 pines (DE-9), más barato e incluso más extendido para cierto tipo de periféricos (como el ratón serie del PC) [Zacker-Rourke,2001].

### 2.5.1 Conexiones (Desde el DTE)

En la siguiente tabla se muestran las señales RS-232 más comunes según los pines asignados:

Señal		DB-9 (TIA-574)
Common Ground	G	5
Transmitted Data	TD	3
Received Data	RD	2
Data Terminal Ready	DTR	4
Data Set Ready	DSR	6
Request To Send	RTS	7
Clear To Send	CTS	8
Carrier Detect	DCD	1
Ring Indicator	RI	9

**Tabla 1.1. Señales RS-232**  
**Fuente: [Zacker-Rourke,2001]**

### 2.5.2. Construcción Física

La interfaz RS-232 está diseñada para distancias cortas, de unos 15 metros o menos, y para velocidades de comunicación bajas, de no más de 20 Kb. La interfaz puede trabajar en comunicación asíncrona o síncrona y tipos de canal simplex, half duplex o full duplex. En un canal simplex los datos siempre viajarán en una dirección, por ejemplo desde DCE a DTE. En un canal half duplex, los datos pueden viajar en una u otra dirección, pero sólo durante un determinado periodo de tiempo; luego la línea debe ser conmutada antes que los datos

puedan viajar en la otra dirección. En un canal full duplex, los datos pueden viajar en ambos sentidos simultáneamente. Las líneas de *handshaking* de la RS-232 se usan para resolver los problemas asociados con este modo de operación, tal como en qué dirección los datos deben viajar en un instante determinado.

Si un dispositivo de los que están conectados a una interfaz RS-232 procesa los datos a una velocidad menor de la que los recibe deben de conectarse las líneas *handshaking* que permiten realizar un control de flujo tal que al dispositivo más lento le de tiempo de procesar la información. Las líneas de "*hand shaking*" que permiten hacer este control de flujo son las líneas RTS y CTS. Los diseñadores del estándar no concibieron estas líneas para que funcionen de este modo, pero dada su utilidad en cada interfaz posterior se incluye este modo de uso [Zacker-Rourke,2001].

### 2.5.3. Transmisión De Datos

La transmisión de datos (en forma de ceros y unos) entre los dispositivos a través de alguna forma de medio de transmisión (como un cable). La transmisión de datos se considera local si los dispositivos de comunicación están en el mismo edificio o en un área geográfica restringida y se considera remota si los dispositivos están separados por una distancia considerable [Forovzan,2002].

#### 2.5.3.1. Componentes

Un sistema de transmisión de datos está formado por cinco componentes.

1. **Mensaje:** Es la información (datos) a comunicar. Puede estar formado por texto, números, gráficos, sonido o video o cualquier combinación de los anteriores .
2. **Emisor:** Es el dispositivo que envía los datos del mensaje. Puede ser una computadora, una estación de trabajo, un teléfono y muchas otras.
3. **Receptor:** Es el dispositivo que recibe el mensaje. Puede ser una computadora, una estación de trabajo, un teléfono y muchas otras.
4. **Medio:** El medio de transmisión es el camino físico por el cual viaja el mensaje del emisor al receptor. Puede estar formado por un cable de par trenzado, un cable coaxial, un cable de fibra óptica, etc.



5. **Protocolo:** Es un conjunto de reglas que gobiernan la transmisión de datos. Representa un acuerdo entre dos dispositivos que se comunican [Forovzan,2002].

### 2.5.3.2. Tasa de bits y tasa de baudios

Dos términos que se usan frecuentemente en la transmisión de datos son la *tasa de bits* y la *tasa de baudios*. La **tasa de bits** es el número de bits transmitidos durante un segundo. La **tasa de baudios** indica el número de unidades de señal por segundo necesarias para representar estos bits. Cuando se habla de la eficiencia de las computadoras, la tasa de bits es lo más importante; se quiere saber cuanto cuesta procesar cada pieza de la información. Sin embargo, en la transmisión de datos es más importante conocer la eficiencia con que se pueden mover los datos de un lado para otro, tanto en piezas como en bloques. Cuantas menos unidades de señal sean necesarias, más eficiente será el sistema y menor será el ancho de banda para transmitir más bits; por tanto es más importante lo concerniente a la tasa de baudios. La tasa de baudios determina el ancho de banda necesario para enviar la señal.

La tasa de bits es igual a la tasa de baudios por el número de bits representados para cada unidad de señal. La tasa de baudios es igual a la tasa de bits dividida por el número de bits representados por cada desplazamiento de la señal. La tasa de bits siempre es mayor o igual a la tasa de baudios. Entonces tenemos las siguientes formulas:

Tasa de bits = Tasa de baudios x Número de bits por elementos de señal [Forovzan,2002].

## 2.6. SISTEMAS DE TIEMPO REAL

Un sistema real de tiempo real es un sistema informático que interacciona repetidamente con su entorno físico. Responde a los estímulos que recibe del mismo dentro de un plazo de tiempo determinado (para que el funcionamiento del sistema sea correcto no basta con que las acciones sean correctas, sino que tienen que ejecutarse dentro del intervalo de tiempo especificado). El tiempo en que se ejecutan las acciones del sistema es significativo.

Ejemplos de aplicación: Aviónica, Control de tráfico aéreo, Control de trenes, Control de automóviles, Telecomunicaciones, Producción y distribución de energía eléctrica, Electrónica de consumo, Sistemas multimedia.

### 2.6.1. Entorno De Un Sistema De Tiempo Real

Muchos sistemas de tiempo real son componentes de otros sistemas, en los que realizan funciones de control, en este caso se dice que se trata de sistemas empotrados.

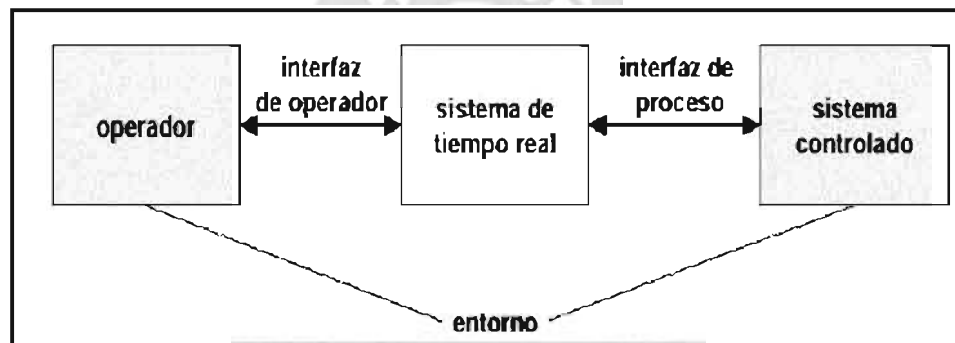


Fig. 2.3. Entorno de un sistema de tiempo real  
Fuente: [Carretero,2009]

Ejemplos: automóviles, electrónica de consumo (teléfonos, radios, televisores), Electrodomésticos, periféricos del computador. Características: A menudo, el computador no es visible desde fuera y los recursos con los que se cuenta son limitados.

### 2.6.2. Requisitos Funcionales

#### 2.6.2.1. Adquisición de Datos

- Medida de variables
  - Cada variable pertenece a la esfera de control de un subsistema.
  - Fuera de su esfera de control se pueden observar, pero no cambiar.
  - La imagen de una variable tiene un intervalo de validez.
  - Las medidas pueden estar dirigidas por tiempo o por sucesos.
- Acondicionamiento de señales
  - Los datos brutos se filtran y convierten a unidades de ingeniería.
  - Luego se analizan para obtener datos validados



- Supervisión
  - Cuando algún dato tiene valores incorrectos se generan alarmas.
  - Es importante identificar el suceso original de alarmas en cadenasas.

### 2.6.2.2. Control Digital Directo

- El sistema de tiempo real puede actuar sobre el sistema controlado para conseguir que tenga un comportamiento determinado.
- Normalmente se usa esquema con realimentación
  - La acción de control es función de la desviación entre los valores de referencia y los valores medios de las variables.
  - El diseño de los algoritmos de control es un problema importante.
  - El comportamiento es muy regular (periódico)

### 2.6.2.3. Interacción con personas

- Aspectos típicos:
  - Presentación de datos.
  - Presentación de alarmas.
  - Presentación de tendencias.
  - Registro de datos.
  - Generación de informes.

### 2.6.3. Requisitos Temporales

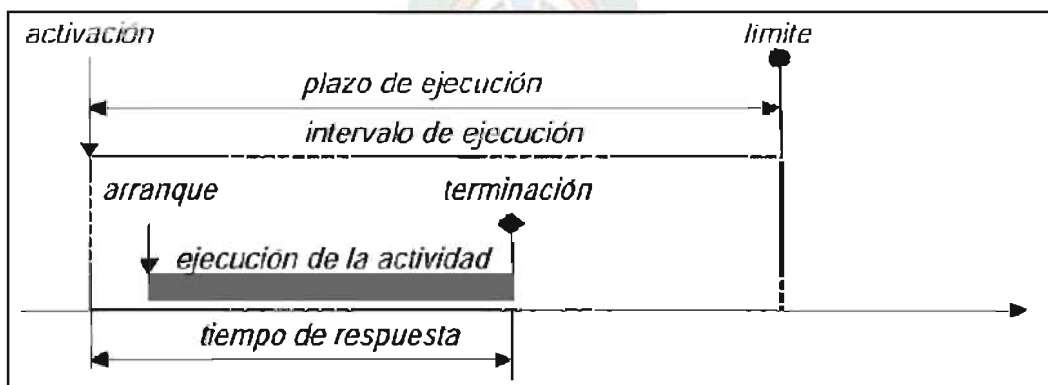
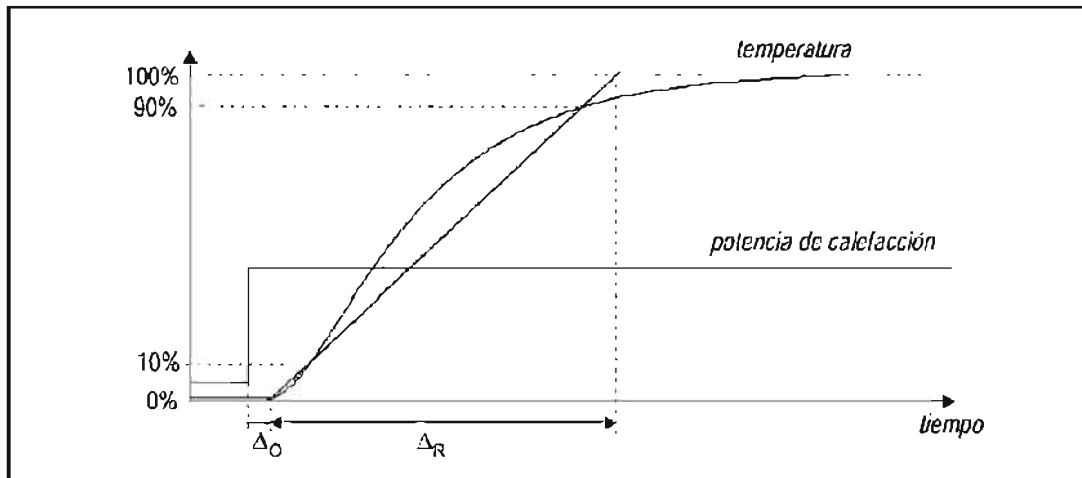


Fig. 2.4. Requisitos temporales  
Fuente: [Carretero,2009]

Las acciones de los sistemas de tiempo real se ejecutan repetidamente, dentro de intervalos de tiempo determinados.

### 2.6.3.1. Origen de Requisitos Temporales

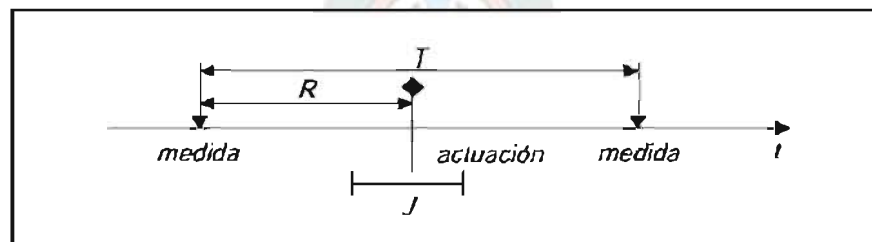
Los más críticos aparecen en los bucles de control



**Fig. 2.5. Origen de requisitos temporales**  
Fuente: [Carretero,2009]

### 2.6.3.2. Control por Computador

El computador muestrea periódicamente la variable controlada, la compara con el valor de referencia y calcula el valor de la variable de control mediante un algoritmo de control, el comportamiento tarda un cierto tiempo de respuesta  $R$  en calcular la acción de control:



**Fig. 2.6. Control por computador**  
Fuente: [Carretero,2009]

$J$  es la variación (jitter) en el tiempo de respuesta,  $J = R_{max} - R_{min}$

### 2.6.3.3. Relación Entre Parámetros Temporales

Un periodo de muestreo de Tiempo Real excesivo puede falsear la imagen de las variables del sistema controlado. D es un plazo de respuesta menor o igual que el periodo. Una variación significativa en R puede comprometer el correcto funcionamiento del algoritmo de control.

### 2.6.4. Requisitos de Fiabilidad y Seguridad

- Fiabilidad
  - Probabilidad de proporcionar el servicio especificado. Para una tasa de fallos constante de  $\lambda$  averías/hora se tiene  $R(t) = e^{-\lambda(t-t_0)}$
  - Tiempo medio entre averías: los sistemas con  $MTBF > 10^9$
- Seguridad
  - Tipos de averías: malignas y benignas. Una avería maligna tiene un coste superior a la utilidad del sistema.
  - Sistemas críticos
    - Ultra fiables respecto a las averías malignas.
    - En muchos casos se exige una certificación de seguridad efectuada por un organismo independiente.
- Mantenibilidad
  - Medida del tiempo necesario para recuperar una avería benigna
  - Tiempo medio de reparación:  $MTTR = 1/\mu$  (Problema: Fácil de mantener > menos fiable)
- Disponibilidad
  - Fracción de tiempo durante el cual el sistema está disponible, depende del MTBF y el MTBR.

### 2.6.5. Clases de Sistemas de Tiempo Real

- Según la prioridades del sistema controlado
  - Sistemas críticos y sistemas acríticos
  - Sistemas con parada segura y sistemas con degradación aceptable

- **Según las propiedades del sistema de tiempo real**
  - Sistemas con tiempo de respuesta garantizado y sistemas que hacen lo que pueden
  - Sistemas con recursos adecuados y sistemas con recursos inadecuados
  - Sistemas dirigidos por tiempo y sistemas dirigidos por eventos
- **Según la instalación del sistema de tiempo real**
  - Sistemas convencionales y empotrados
  - Sistemas centralizados y distribuidos

### 2.6.5.1. Sistemas De Tiempo Real Críticos y Acríticos

Se distinguen por sus requisitos temporales y de fiabilidad

<b>Sistemas críticos (hard real-time systems)</b>	<b>Sistemas acríticos (soft real-time systems)</b>
<ul style="list-style-type: none"> <li>• Plazo de respuesta estricto</li> <li>• Comportamiento temporal determinado por el entorno</li> <li>• Comportamiento en sobrecargas predecible</li> <li>• Requisitos de seguridad críticos</li> <li>• Redundancia activa</li> <li>• Volumen de datos reducido</li> </ul>	<ul style="list-style-type: none"> <li>• Plazo de respuesta flexible</li> <li>• Comportamiento temporal determinado por el computador</li> <li>• Comportamiento en sobrecargas degradado</li> <li>• Requisitos de seguridad acríticos</li> <li>• Recuperación de fallos</li> <li>• Gran volumen de datos</li> </ul>

### 2.6.5.2. Sistemas De Tiempo Real Con Parada Segura O Con Degradación Aceptable

Se distinguen por su comportamiento en caso de avería.

<b>Sistemas con parada segura (fail safe)</b>	<b>Sistemas con degradación aceptable (fail soft)</b>
<ul style="list-style-type: none"> <li>• Detención en estado seguro</li> <li>• Probabilidad de detección de fallos elevada</li> <li>• Funcionamiento con pérdida parcial de funcionalidad</li> </ul>	<ul style="list-style-type: none"> <li>• Funcionamiento con pérdida parcial de funcionalidad o prestaciones</li> <li>• También hay sistemas con tolerancia de fallas completa (fail operational)</li> </ul>

### 2.6.5.3. Sistemas de Tiempo Real Con Respuesta Garantizada o lo Mejor Posible

Se definen por su grado de determinismo temporal.

<b>Sistemas con respuesta garantizada (guaranteed response systems)</b>	<b>Sistemas que hacen lo que pueden (best-effort systems)</b>
<ul style="list-style-type: none"> <li>• Comportamiento temporal garantizado analíticamente</li> <li>• Hace falta caracterizar con precisión la carga máxima y los posibles fallos</li> </ul>	<ul style="list-style-type: none"> <li>• Comportamiento temporal de tipo “lo mejor que se pueda”</li> <li>• No se hace un caracterización precisa de carga y fallos</li> </ul>

#### 2.6.5.4. Sistemas de Tiempo Real con Recursos Adecuados o Inadecuados

Se distinguen por la cantidad de recursos disponibles

<b>Sistemas con recursos adecuados (resource-adequate systems)</b>	<b>Sistemas con recursos inadecuados (resource-inadequate systems)</b>
<ul style="list-style-type: none"> <li>• Diseño con suficientes recursos para garantizar el comportamiento temporal con máxima carga y en cualquier caso</li> </ul>	<ul style="list-style-type: none"> <li>• Diseño con recursos “razonables” desde un punto de vista económico</li> <li>• Sólo sirve para sistemas acríticos</li> </ul>

#### 2.6.5.5. Sistemas Dirigidos por Tiempo y por Sucesos

Se distinguen por la forma de arrancar la ejecución de sus actividades

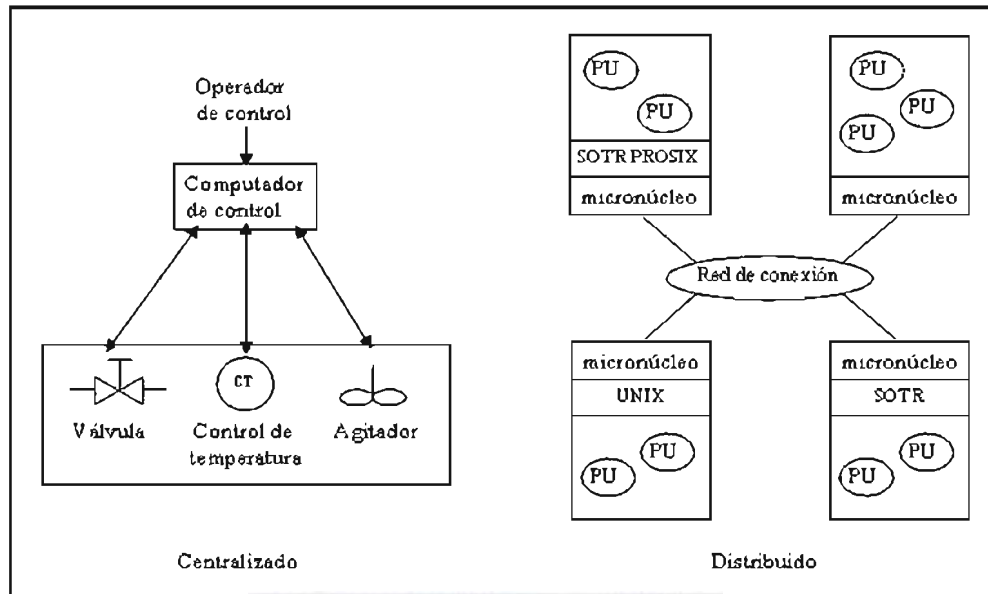
<b>Sistemas dirigidos por sucesos (event-triggered systems)</b>	<b>Sistemas dirigidos por tiempo (time-triggered systems)</b>
<ul style="list-style-type: none"> <li>• Arrancan cuando se produce un suceso de cambio de estado</li> <li>• Mecanismo básico: interrupciones</li> </ul>	<ul style="list-style-type: none"> <li>• Arrancan en instantes de tiempo predeterminados</li> <li>• Mecanismo básico: reloj</li> </ul>

#### 2.6.5.6. Sistemas Convencionales y Empotrados

Se distinguen por la forma de integrar e instalar el sistema de tiempo real.

<b>Sistemas convencionales</b>	<b>Sistemas empotrados</b>
<ul style="list-style-type: none"> <li>• Se instalan en computadores convencionales</li> <li>• Se dispone de todos los mecanismos del computador (IU, almacenamiento, etc.)</li> </ul>	<ul style="list-style-type: none"> <li>• El computador es sólo una parte de un sistema de ingeniería mayor y no parece un computador</li> <li>• Un computador empotrado: <ol style="list-style-type: none"> <li>1. No es visible al usuario</li> <li>2. No es reprogramable por el usuario</li> <li>3. Posee recursos limitados</li> <li>4. Tiene una interfaz de usuario específica de la aplicación</li> </ol> </li> </ul>

### 2.6.5.7. Sistemas centralizados y distribuidos



**Fig. 2.7. Sistemas centralizados y distribuidos**  
Fuente: [Carretero,2009]

## 2.7. PROCESOS Y THREADS

### 2.7.1 Proceso

Un proceso es una actividad de algún tipo. Tiene un programa, entrada, salida y un estado. Un único procesador puede compartirse entre varios procesos utilizando un algoritmo de planificación que determine cuándo hay que detener el trabajo sobre un proceso y pasar a atender a otro diferente.

Todos los ordenadores modernos pueden hacer varias cosas a la vez. Mientras un ordenador está ejecutando un programa de usuario puede perfectamente también estar leyendo de un disco e imprimiendo texto en una pantalla o una impresora. En un sistema multiprogramado la CPU también conmuta de unos programas a otros, ejecutando cada uno de ellos durante decenas o cientos de milisegundos. Aunque, estrictamente hablando, en cualquier instante de tiempo la CPU sólo está ejecutando un programa, en el transcurso de 1 segundo ha podido estar trabajando sobre varios programas, dando entonces a los usuarios la impresión de un cierto paralelismo [Tanenbaum,2001].

### **2.7.1.1 El Modelo de los Procesos Secuenciales**

En este modelo, todo el software ejecutable en el ordenador, incluyendo a veces al propio sistema operativo, se organiza en un número de procesos secuenciales, o simplemente procesos para acortar. Un proceso es justamente un programa en ejecución, incluyendo los valores actuales del contador de programa, registros y variables. Conceptualmente cada proceso tiene su propia CPU virtual. En realidad, por supuesto, la CPU real conmuta sucesivamente de un proceso a otro, esta rápida conmutación de un proceso a otro en algún orden se denomina **multiprogramación** [Tanenbaum,2001].

### **2.7.1.2 Creación de Procesos**

En sistemas de propósito general, los cuatro principales sucesos que provocan la creación de nuevos procesos son:

1. La inicialización del sistema
2. La ejecución por parte de un proceso (en ejecución) de una llamada al sistema de creación de un nuevo proceso.
3. La petición por parte del usuario de la creación de un nuevo proceso.
4. El inicio de un trabajo en batch [Tanenbaum,2001].

### **2.7.1.3 Terminación de los Procesos**

Los procesos terminan usualmente debido a una de las siguientes causas:

1. El proceso completa su trabajo y termina (voluntariamente).
2. El proceso detecta un error y termina (voluntariamente).
3. El sistema detecta un error fatal del proceso y fuerza su terminación.
4. Otro proceso fuerza la terminación del proceso

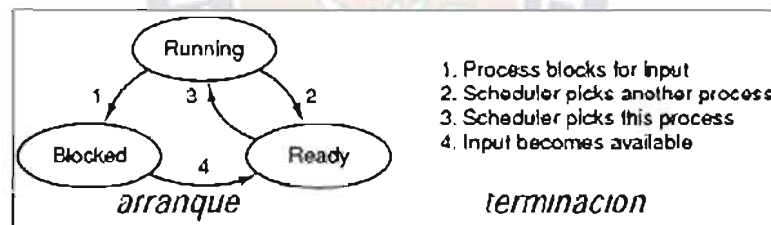
### **2.7.1.4. Estados de los Procesos**

Aunque cada proceso es una entidad independiente, con su propio contador de programa y estado interno, los procesos necesitan a menudo interactuar con otros procesos. Un proceso



puede generar los datos de salida que otro proceso utiliza como entrada. En ese caso el proceso que ejecuta el *grep* debe bloquearse hasta que esté disponible alguna entrada. Cuando un proceso se bloquea, lo hace porque desde un punto de vista lógico no puede continuar, normalmente debido a que está esperando por datos de entrada que aún no están disponibles. También es posible para un proceso que esté conceptualmente preparado y sea capaz de ejecutarse, que esté parado debido a que el sistema operativo ha decidido temporalmente asignar la CPU a otro proceso. Estas dos situaciones son completamente diferentes. En el primer caso, la suspensión es inherente a la situación (no puede procesarse el comando del usuario hasta que no halla terminado de teclearse). En el segundo caso, se trata simplemente de una cuestión técnica del sistema (no hay suficientes CPUs para dar a cada proceso su propio procesador privado). En la Figura 2.8 podemos ver un diagrama de estados mostrando los tres estados en los que puede estar un proceso:

1. En ejecución (utilizando realmente la CPU en ese instante).
2. Preparado (ejecutable; detenido temporalmente para permitir que otro proceso se ejecute).
3. Bloqueado (incapaz de ejecutarse hasta que tenga lugar algún suceso externo).



**Figura 2.8. Estados y transiciones de un proceso**  
**Fuente: [Tanenbaum,2001]**

Desde un punto de vista lógico, los dos primeros estados son similares. En ambos casos el proceso está dispuesto a ejecutarse, sólo que en el segundo caso temporalmente no existe ninguna CPU disponible para él. El tercer estado es diferente de los dos primeros ya que el proceso no puede ejecutarse, ni siquiera en el caso de que la CPU no tuviera ninguna otra cosa que hacer.

Como puede verse son posibles cuatro transiciones entre esos tres estados. La transición 1 tiene lugar cuando un proceso descubre que no puede continuar. En algunos sistemas el proceso debe ejecutar una llamada al sistema, tal como *block* o *pause*, para pasar al estado bloqueado. Las transiciones 2 y 3 se deben a la actuación del planificador de procesos (una



parte del sistema operativo) sin que el proceso tenga conocimiento de ellas. La transición 2 tiene lugar cuando el planificador (*scheduler*) decide que el proceso en ejecución se ha ejecutado durante un tiempo suficientemente largo, y es hora de dejar a otro proceso que reciba algún tiempo de CPU. La transición 3 tiene lugar cuando todos los demás procesos han recibido ya su justa parte del tiempo de CPU, siendo hora ya de que el primer proceso consiga la CPU para ejecutarse de nuevo.

La transición 4 tiene lugar cuando por fin se produce el suceso externo por el cual estaba esperando un proceso (tal como la llegada de algún nuevo dato de entrada). Si no está ejecutándose ningún otro proceso en ese instante, tiene lugar la transición 3 y el proceso prosigue con su ejecución donde había quedado bloqueado. En otro caso el proceso tiene que esperar en el estado *preparado* durante algún tiempo hasta que esté disponible la CPU y le toque el turno.

## **2.7.2. Threads**

### **2.7.2.1. El Modelo de los Threads**

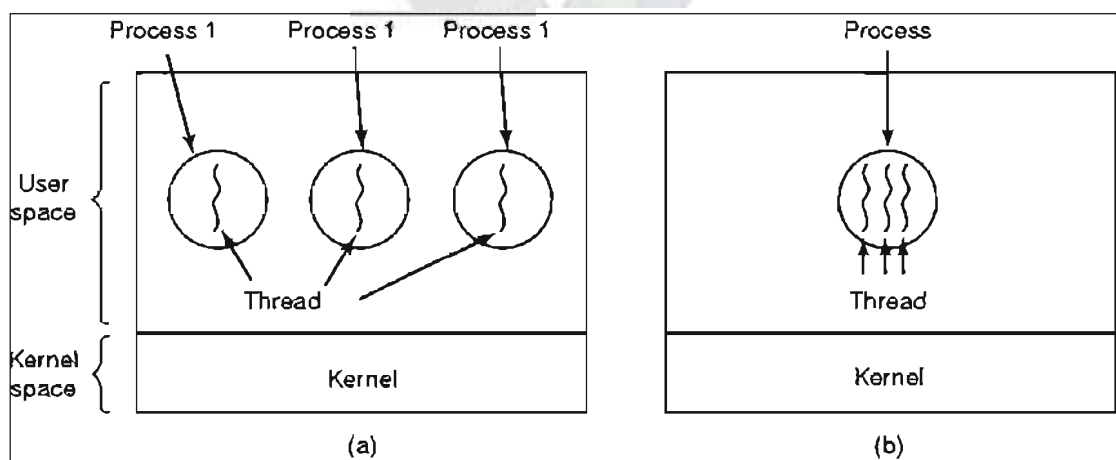
Una forma de ver un proceso es que es una manera de agrupar juntos recursos relacionados. Un proceso tiene un espacio de direcciones conteniendo código y datos del programa, así como otros recursos. Estos recursos pueden incluir ficheros abiertos, procesos hijos, alarmas pendientes, controladores de señales, información de contabilidad, etc. Poniendo juntos todos esos recursos en la forma de un proceso, es posible gestionarlos más fácilmente.

El otro concepto que incluye un proceso es el de hilo de ejecución, usualmente denominado un thread. El thread tiene un contador de programa que indica cuál es la siguiente instrucción a ejecutar. Tiene además registros que contienen sus variables de trabajo actuales. Tiene una pila, que contiene una especie de historia de su ejecución, con una trama por cada procedimiento al que se ha llamado pero del que no se ha retornado todavía. Aunque un thread debe ejecutarse en algún proceso, el thread y su proceso son conceptos diferentes y pueden tratarse de forma separada. Los procesos se utilizan para agrupar recursos juntos; los threads son las entidades planificadas para su ejecución en la CPU.

Lo que los threads añaden al modelo de los procesos es que permiten que haya múltiples ejecuciones en un mismo entorno determinado por un proceso, y esto con un alto grado de independencia de esas ejecuciones. Tener múltiples threads ejecutándose en paralelo dentro de un proceso es análogo a tener múltiples procesos ejecutándose en paralelo dentro de un ordenador. En el primer caso, los threads comparten el espacio de direcciones, los ficheros abiertos y otros recursos. En el segundo caso, los procesos comparten la memoria física, los discos, las impresoras y otros recursos. También se utiliza el término de **multihilo** (*multithreaded*) para describir la situación en la cual se permite que haya múltiples threads en el mismo proceso.

En la Figura 2.9(a) se representan tres procesos tradicionales. Cada proceso tiene su propio espacio de direcciones y un único thread de control. En contraste, en la Figura 2.9(b) se representa un único proceso con tres threads de control. Aunque en ambos casos tenemos tres threads, en la Figura 2.9(a) cada uno de los threads opera en un espacio de direcciones diferente, mientras que en la Figura 2.9(b) todos los threads comparten el mismo espacio de direcciones.

Cuando un proceso multihilo se ejecuta sobre un sistema con una única CPU, los threads deber hacer turnos para ejecutarse. La CPU va conmutándose rápidamente de unos threads a otros proporcionando la ilusión de que los threads se están ejecutando en paralelo, aunque sobre una CPU virtual más lenta que la real.



**Figura 2.9. (a) Tres procesos cada uno con un thread. (b) Un proceso con tres threads.**  
**Fuente: [Tanenbaum,2001]**

Los diferentes threads de un proceso no son tan independientes como si fueran diferentes procesos. Todos los threads tienen exactamente el mismo espacio de direcciones, lo que significa en particular que comparten las mismas variables globales. Ya que cualquier thread puede acceder a cualquier dirección de memoria dentro del espacio de direcciones del proceso, un thread puede leer, escribir o incluso borrar completamente la pila de cualquier otro thread. No existe ninguna protección entre los threads debido a que (1) es imposible establecer ninguna medida de protección, y (2) es innecesario que haya protección. Entonces la organización de la Figura 2.9(a) puede utilizarse cuando los tres procesos están esencialmente no relacionados, mientras que la Figura 2.9(b) puede ser apropiada cuando los tres threads son realmente parte del mismo trabajo y cooperan activa y estrechamente uno con otro.

Los elementos de la primera columna de la Figura 2.10 corresponden a propiedades de los procesos y no a propiedades de los threads. Por ejemplo si un thread abre un fichero, ese fichero es visible por todos los otros threads del proceso, que pueden ponerse inmediatamente a leer y escribir en él. Esto es lógico ya que el proceso es la unidad de gestión de recursos, y no el thread. Si cada thread tuviera su propio espacio de direcciones, ficheros abiertos, alarmas pendientes, etc. se trataría de un proceso separado. Lo que estamos tratando de conseguir con el concepto de thread es la capacidad de que múltiples hilos de ejecución compartan un conjunto de recursos de manera que puedan trabajar estrechamente juntos para realizar alguna tarea.

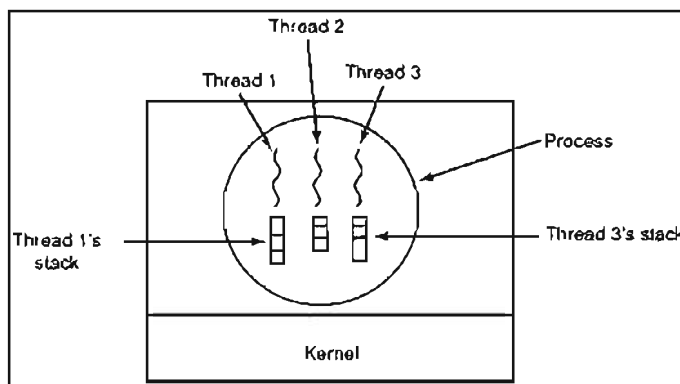
Atributos propios de los procesos	Atributos propios de los threads
Espacio de direcciones	Contador de programa
Variables globales	Registros
Ficheros abiertos	Pila
Procesos hijos	Estado
Alarmas pendientes	
Señales y controladores de señales	
Información de contabilidad	

**Figura 2.10. Atributos propios de los procesos y threads**

**Fuente: [Tanenbaum,2001]**

Igual que un proceso tradicional (es decir un proceso con un único thread), un thread puede estar en cualquiera de los estados: en ejecución, bloqueado, preparado o terminado. Un thread en ejecución tiene actualmente la CPU y está activo. Un thread bloqueado está esperando a que algún suceso lo desbloquee. Un thread puede bloquearse esperando a que

tenga lugar algún suceso externo o a que algún otro thread lo desbloquee. Un thread preparado está planificado para ejecutarse y lo hace tan pronto como le llega su turno. Las transiciones entre los estados de un thread son las mismas que las transiciones entre los estados de un proceso y se ilustran en la Figura 2.8. Es importante que nos demos cuenta de que cada thread tiene su propia pila, como se muestra en la Figura 2.11 La pila de cada thread contiene una trama (o registro de activación) por cada procedimiento al que se ha llamado pero del que todavía no se ha retornado. Esta trama contiene las variables locales del procedimiento y la dirección de retorno a utilizar cuando termine la llamada al procedimiento.



**Figura 2.11. Cada thread tiene su propia pila.**  
**Fuente: [Tanenbaum,2001]**

Cuando estamos en un sistema multihilo, los procesos normalmente comienzan teniendo un único thread. Este thread tiene la capacidad de crear nuevos threads llamando a un procedimiento de biblioteca, por ejemplo *thread\_create*. A veces los threads mantienen una relación jerárquica de tipo padre-hijo, pero a menudo no existe tal relación, siendo todos los threads iguales. Con o sin relación jerárquica, siempre se devuelve al thread que hace la llamada de creación un identificador de thread que sirve de nombre para el nuevo thread.

Cuando un thread concluye su trabajo puede dar por terminada su vida llamando a un procedimiento de biblioteca, por ejemplo, *thread\_exit*. En ese momento el thread se desvanece dejando ya de ser planificable. En algunos sistemas de threads, un thread puede esperar a que termine otro thread (específico), a través de una llamada al sistema, como por ejemplo, *thread\_wait*. Este procedimiento bloquea al proceso que lo invoca hasta que un thread (específico) termina. Desde este punto de vista la creación y terminación de los threads se parece mucho a la creación y terminación de los procesos, teniendo

aproximadamente las mismas opciones. Otra llamada común relacionada con los threads es *thread\_yield*, que permite que un thread abandone voluntariamente la CPU para permitir que se ejecute algún otro thread. Tal llamada es importante ya que no existe ninguna interrupción de reloj que dé soporte al tiempo compartido como en el caso de los procesos. Por tanto para los threads es importante ser corteses y ceder voluntariamente la CPU de cuando en cuando para dar la oportunidad de que se ejecuten otros threads. Hay otras llamadas que permiten a un thread esperar a que otro thread termine algún trabajo, o a que un thread anuncie que ha terminado algún trabajo, etc.

### 2.7.2.2. Utilización de los Threads

La razón principal para tener threads es que son numerosas las aplicaciones en las que hay varias actividades que están en marcha simultáneamente. De vez en cuando, alguna de estas actividades puede bloquearse. En esa situación el modelo de programación resulta más sencillo si descomponemos tal aplicación en varios threads secuenciales que se ejecutan en paralelo.

Primero con los threads introducimos un nuevo elemento: la capacidad de las entidades paralelas para compartir entre ellas un espacio de direcciones y todos sus datos. Esta capacidad es esencial para ciertas aplicaciones, y es por lo que el tener simplemente múltiples procesos (con sus espacios de direcciones separados) no puede funcionar en estos casos.

Un segundo argumento para tener threads es que ya que no tienen ningún recurso ligado a ellos, son más fáciles de crear y destruir que los procesos.

Una tercera razón para tener threads es también un argumento sobre el rendimiento. Los threads no proporcionan ninguna ganancia en el rendimiento cuando todos ellos utilizan intensamente la CPU, sino cuando hay una necesidad substancial tanto de cálculo en la CPU como de E/S, de manera que teniendo threads se puede conseguir que esas dos actividades se solapen, acelerando la ejecución de la aplicación.

Finalmente, los threads son útiles sobre sistemas con varias CPUs, donde es posible un paralelismo auténtico.

### 2.7.3. Implementación de los Threads en el Espacio del Usuario

Existen dos formas fundamentales de implementar un paquete de threads: en el espacio del usuario y en el núcleo (*kernel*).

El primer método consiste en poner el paquete de threads enteramente en el espacio de usuario. En consecuencia el núcleo del sistema no sabe nada de su existencia. En lo que concierne al núcleo, sólo se gestionan procesos ordinarios con un único thread. La primera, y más obvia, ventaja es que el paquete de threads a nivel de usuario puede implementarse sobre un sistema operativo que no soporte threads.

Todas estas implementaciones tienen la misma estructura general, que se ilustra en la Figura 2.12(a). Los threads se ejecutan en lo alto de un **sistema en tiempo de ejecución** (*runtime system*), que es una colección de procedimientos que gestiona los threads.

Cuando los threads se gestionan en el espacio del usuario, cada proceso necesita su propia **tabla de threads** privada para llevar el control de sus threads. Esta tabla es análoga a la tabla de procesos del núcleo, salvo que sólo controla las propiedades propias de los threads tales como el contador de programa del thread, su puntero de pila, sus registros, su estado, etc. La tabla de threads está gestionada por el sistema en tiempo de ejecución. Cuando un thread pasa al estado preparado o bloqueado, la información necesaria para proseguir posteriormente con su ejecución se guarda en la tabla de threads, exactamente de la misma forma que el núcleo almacena la información sobre los procesos en la tabla de procesos.

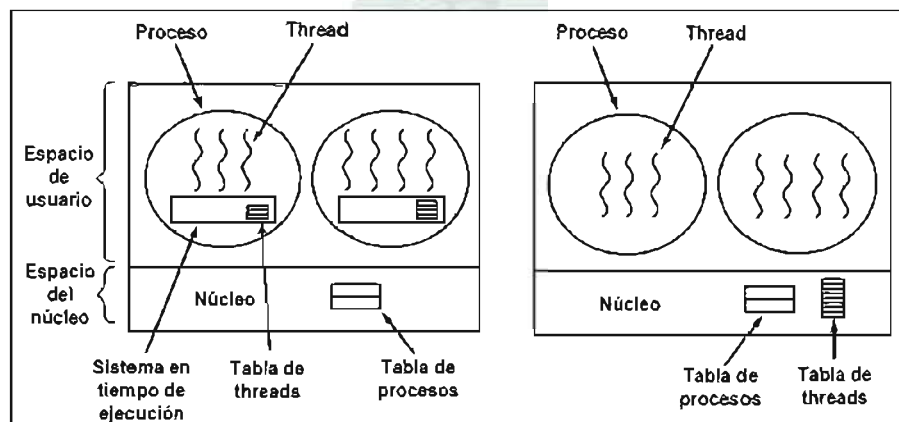


Figura 2.12. (a) Un paquete de threads a nivel de usuario. (b) Un paquete de threads gestionado por el núcleo. Fuente: [Tanenbaum,2001]



Cuando un thread hace algo que puede provocar que se bloquee localmente, como por ejemplo, esperar a que otro thread en su mismo proceso complete algún trabajo, entonces llama a un procedimiento del sistema en tiempo de ejecución. Este procedimiento comprueba si el thread debe pasar al estado bloqueado. Si es así, el procedimiento guarda los registros del thread en la tabla de threads, busca en la tabla un thread preparado para ejecutarse, y recarga los registros de la máquina con los valores que se tienen guardados correspondientes al nuevo thread. Tan pronto como se conmutan el puntero de pila y el contador de programa, el nuevo thread vuelve automáticamente otra vez a la vida. Este tipo de conmutación de threads es al menos un orden de magnitud más rápido que hacer un trap al núcleo, lo que representa un potente argumento a favor de los paquetes de threads a nivel de usuario.

Los threads a nivel de usuario tienen también otras ventajas. Para empezar, permiten que cada proceso tenga su propio algoritmo de planificación ajustado a sus necesidades. También se dimensionan mejor, ya que los threads a nivel del núcleo requieren invariablemente en el núcleo algún espacio para la tabla de threads y algún espacio de pila, lo que puede resultar un problema cuando el número de threads es muy grande.

A pesar de su mayor eficiencia, los paquetes de threads a nivel de usuario tienen algunos serios problemas. El primero de ellos es el problema de cómo se implementan las llamadas al sistema bloqueantes sin que el bloqueo de un thread afectase a los demás. El segundo es el problema de las faltas de página. Otro problema con los paquetes de threads a nivel de usuario es que si un thread comienza a ejecutarse, ningún otro thread en ese proceso podrá volver a ejecutarse mientras que el primer thread no ceda voluntariamente la CPU.

#### **2.7.4 Implementación de los Threads en el Núcleo**

Ahora vamos a considerar el caso en el que el núcleo conoce y gestiona los threads. En ese caso, como se muestra en la Figura 2.12 (b), no es necesario ningún sistema en tiempo de ejecución dentro de cada proceso. Igualmente no existe ninguna tabla de threads en cada proceso. En vez de eso, el núcleo mantiene una tabla de threads que sigue la pista de todos los threads en el sistema. Cuando un thread desea crear un nuevo thread o destruir uno que ya existe, hace una llamada al núcleo, que es el que se encarga efectivamente de su creación o destrucción actualizando la tabla de threads del sistema.

Todas las llamadas que puedan bloquear a un thread se implementan como llamadas al sistema, a un coste considerablemente más alto que una llamada a un procedimiento del sistema en tiempo de ejecución. Cuando se bloquea un thread, el núcleo tiene la opción de decidir si pasa a ejecutar otro thread del mismo proceso (si es que hay alguno preparado), o pasa a ejecutar un thread de un proceso diferente. Con threads a nivel de usuario el sistema en tiempo de ejecución tiene que seguir ejecutando threads de su propio proceso hasta que el núcleo le arrebatase la CPU (o hasta que no le queden threads preparados para ejecutarse).

Los threads a nivel del núcleo no requieren ninguna llamada al sistema nueva de tipo no bloqueante. Además, si un thread de un proceso provoca una falta de página, el núcleo puede comprobar fácilmente si el proceso tiene todavía threads ejecutables, y en ese caso pasar a ejecutar uno de ellos mientras espera a que la página que provocó la falta se cargue desde el disco. Su principal desventaja es que el coste de una llamada al sistema es considerable, de manera que si las operaciones con threads (creación, terminación, etc.) son frecuentes, puede incurrirse en una elevada sobrecarga para el sistema.

### 2.7.7 Threads Emergentes

Frecuentemente los threads son muy útiles en sistemas distribuidos. Un ejemplo importante es cómo se tratan los mensajes de entrada, por ejemplo peticiones de servicio. El enfoque tradicional es tener un proceso o thread que está bloqueado en una llamada al sistema *receive* esperando a que llegue un mensaje de entrada. Cuando llega un mensaje, ese proceso o thread acepta el mensaje y lo procesa.

Sin embargo, es posible un enfoque completamente diferente, en el cual la llegada de un mensaje provoca que el sistema cree un nuevo thread para tratar el mensaje. Un tal thread se denomina un **thread emergente** (*pop-up thread*) y se ilustra en la Figura 2.13. Una ventaja clave de los threads emergentes es que ya que son trigu nuevo, no tienen ninguna historia – registros, pila, etc. que deba restaurarse. Cada uno comienza limpio y cada uno es idéntico a todos los demás. Esto hace posible crear muy rápidamente tales threads. Al nuevo thread se le da el mensaje de entrada a procesar. El resultado de utilizar los threads emergentes es que la latencia entre la llegada del mensaje y el comienzo del procesamiento puede hacerse muy pequeña.

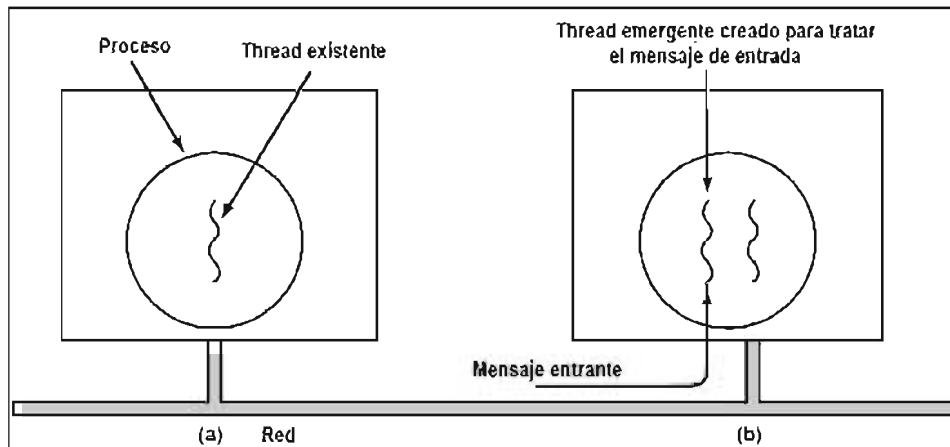


Figura 2.13. Creación de un nuevo thread cuando llega un mensaje. (a) Antes de que llegue el mensaje. (b) Después de que llegue el mensaje. Fuente: [Tanenbaum,2001]

## 2.8. PLANIFICACIÓN

En un ordenador multiprogramado es frecuente que en un momento dado haya múltiples procesos compitiendo por el uso de la CPU al mismo tiempo. Esta situación se da siempre que dos o más procesos están simultáneamente en el estado preparado. Si sólo hay una CPU disponible, es necesario hacer una elección para determinar cual de esos procesos será el siguiente que se ejecute. La parte del sistema operativo que realiza esa elección se denomina el **planificador** (*scheduler*), y el algoritmo que se utiliza para esa elección se denomina el **algoritmo de planificación**.

El planificador, además de tener que escoger el proceso correcto para ejecutar, tiene que preocuparse también de realizar un uso eficiente de la CPU, debido a que la conmutación de procesos es muy costosa. Para empezar, debe producirse el paso de modo usuario a modo supervisor. Luego debe salvarse el estado del proceso actual, incluyendo el almacenamiento de los registros en la tabla de procesos de forma que puedan recargarse posteriormente. En muchos sistemas, también debe salvarse el mapa de memoria (por ejemplo, los bits de referencia a memoria en la tabla de páginas). A continuación debe seleccionarse un nuevo proceso mediante la ejecución del algoritmo de planificación. Después de eso, la MMU (Unidad de Gestión de Memoria) debe volverse a cargar con el mapa de memoria del nuevo proceso. Finalmente, debe arrancarse el nuevo proceso. Adicionalmente a todo eso, la conmutación del proceso usualmente inválida toda la memoria caché, forzando que sea recargada dinámicamente desde la memoria principal dos veces (tras entrar en el núcleo y tras abandonarlo). En resumen, la realización de demasiadas conmutaciones de procesos

por segundo puede consumir una cantidad significativa de tiempo de CPU, por lo que debemos ser prudentes.

### 2.8.1 Planificación en Sistemas de Tiempo Real

El comportamiento en tiempo real se logra dividiendo el programa en varios procesos cuyo comportamiento es predecible y conocido por adelantado. Generalmente, tales procesos son cortos y pueden terminar su trabajo en mucho menos de un segundo. Cuando se detecta un suceso externo, el planificador debe planificar los procesos de tal modo que se cumplan todos los plazos. Los sucesos a los que un sistema de tiempo real debe tener que responder pueden clasificarse como **periódicos** (que se presentan a intervalos regulares) o **aperiódicos** (cuya ocurrencia es impredecible). Un sistema puede tener que responder a múltiples flujos de sucesos periódicos. Dependiendo de cuanto tiempo se requiere para procesar cada suceso, podría no ser siquiera posible atenderlos a todos. Por ejemplo, si hay  $m$  sucesos periódicos y el suceso  $i$  tiene lugar con un periodo  $P_i$  y su tratamiento requiere  $C_i$  segundos de tiempo de CPU, entonces el sistema sólo podrá soportar esa carga si :

$$\sum_{i=1}^m \frac{C_i}{P_i} \leq 1$$

Un sistema en tiempo real que cumpla este criterio se dice que es **planificable**.

Por ejemplo, consideremos un sistema de tiempo real moderado con tres sucesos periódicos, con periodos de 100, 200 y 500 milisegundos, respectivamente. Si el tratamiento de estos sucesos requiere 50, 30 y 100 milisegundos de tiempo de CPU respectivamente, el sistema es planificable porque  $0,5 + 0,15 + 0,2 < 1$ . Si se añade un cuarto suceso con un periodo de 1 segundo, el sistema seguirá siendo planificable, en tanto que ese proceso no necesite más de 150 milisegundos de tiempo de CPU para su tratamiento. Los algoritmos de planificación para tiempo real pueden ser estáticos o dinámicos. Los primeros toman sus decisiones de planificación antes de que el sistema comience a ejecutarse. Los segundos toman las decisiones en tiempo de ejecución. La planificación estática sólo funciona si se está perfectamente informado por anticipado sobre el trabajo que debe realizarse y los plazos que deben cumplirse. Los algoritmos de planificación dinámica no tienen estas restricciones.

## 2.9. PROCESO UNIFICADO RUP

El proceso unificado es un proceso de desarrollo de software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. Sin embargo, el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños reprojectos [RUP,2005].

Rational Unified Process (RUP) es un proceso de Ingeniería de Software planteado por Kruchten (1996) cuyo objetivo es producir software de alta calidad, es decir:

- a) Cumple con los requerimientos de los usuarios dentro de una planificación y presupuestos establecidos.
- b) Proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo de software.
- c) Utiliza el Unified Modeling Language (UML), para especificar, visualizar, construir y documentar componentes de sistemas de software, así también permite especificar modelos de negocios y otros sistemas.
- d) Es un proceso controlado de iteraciones con fuerte focalización hacia la arquitectura de sistemas [Contreras-Laciar,2005].

No obstante, los verdaderos aspectos definidos del Proceso Unificado se resumen en tres facetas clave: dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental [RUP,2005].

### 2.9.1. Proceso Unificado Dirigido Por Casos De Uso

Uno de los principales aspectos que definen el Proceso Unificado es que está dirigido por casos de uso que nos ayudan a realizar la captura de requisitos.

La captura de requisitos tiene dos objetivos: encontrar los verdaderos requisitos y representarlos de modo adecuado para los usuarios, clientes y desarrolladores. Entendemos por “verdaderos requisitos” aquellos que cuando se implementen, añadan el valor esperado



para los usuarios. Con “representarlos de un modo adecuado para los usuarios, clientes y desarrolladores”, se entiende, que la descripción obtenida de los requisitos debe ser comprensible por los usuarios y clientes.

Un caso de uso, es un fragmento de confidencialidad del sistema que proporciona al usuario un resultado importante. Los casos de uso representan los requisitos funcionales, todos los casos de uso juntos contribuyen al modelado de casos de uso, el cual describe la funcionalidad total del sistema [RUP,2005].

### **2.9.2. Proceso Unificado Centrado En La Arquitectura**

El proceso de arquitectura de software incluye los aspectos estáticos y dinámicos más significativos del sistema. La arquitectura surge de las necesidades de la empresa como perciben usuarios e inversores, y se reflejan en los casos de uso. Sin embargo se ve influida por muchos otros factores, como la plataforma en que tiene que funcionar el software (arquitectura hardware, sistema operativo, sistema de gestión de bases de datos, protocolos para comunicaciones de red), los bloques de construcción reutilizables que se dispone (Por ejemplo un marco de trabajo para interfases gráficas de usuario), consideraciones de implantación, sistemas heredados, y requisitos no funcionales (por ejemplo, rendimiento, fiabilidad) [RUP,2005].

La arquitectura se desarrolla mediante iteraciones, principalmente durante la fase de elaboración. Cada iteración se desarrolla como esbozo, comenzando con los requisitos y siguiendo con el análisis, diseño, implementación y pruebas, pero centrándonos en los casos de uso relevantes desde el punto de vista de la arquitectura y en otros requisitos [RUP,2005].

### **2.9.3. Proceso Unificado Iterativo e Incremental**

Es práctico dividir el trabajo en partes más pequeñas o mini proyectos. Cada mini proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos, al crecimiento del producto (Fig. 2.14). Para una efectividad máxima, las iteraciones deben estar controladas; esto es que deben seleccionarse y ejecutarse de una forma planificada [RUP,2005].





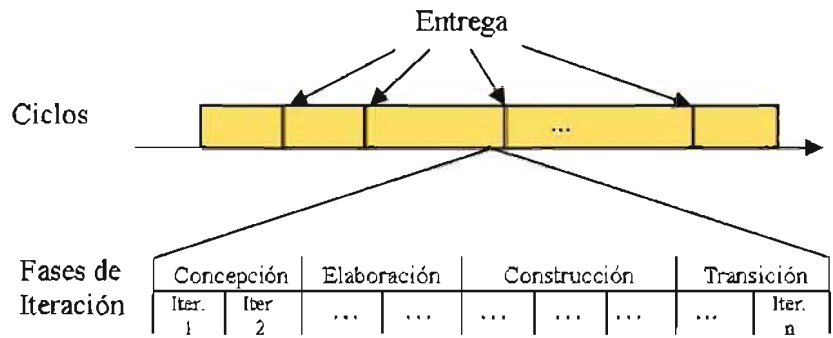
**Fig. 2.14. Proceso Iterativo e Incremental**  
**Fuente: [RUP,2005].**

Son muchos los beneficios de un proceso iterativo controlado:

- La iteración controlada reduce el coste de riesgo a los costes de un solo incremento. Si los desarrolladores tienen que repetir la iteración, la organización solo pierde el esfuerzo mal empleado de la iteración, no el valor del producto entero.
- La iteración controlada reduce el riesgo de no sacar al mercado el producto en el calendario previsto.
- La iteración controlada acelera el ritmo del esfuerzo de desarrollo en su totalidad debido a que los desarrolladores trabajan de manera más eficiente para obtener resultados claros a corto plazo, en lugar de tener un calendario largo, que se prolonga eternamente [RUP,2005].

#### 2.9.4. Ciclo De Vida De R.U.P.

El proceso Unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema [RUP,2005], como se muestra en la figura 2.15 cada ciclo concluye con una versión del producto para los clientes. Así mismo cada ciclo consta de cuatro fases: inicio, elaboración, construcción y transición. Cada fase se subdivide en iteraciones .



**Fig. 2.15. Un ciclo con sus fases e iteraciones**  
**Fuente: [RUP,2005].**

### 2.9.5. Fases R.U.P.

Por otro lado cada ciclo se desarrolla a lo largo del tiempo. Este tiempo, a su vez, se divide en cuatro fases, como se muestra en la figura 2.16. A través de una secuencia de modelos, implicados muestra lo que esta sucediendo en estas fases.

#### 2.9.5.1. Fase inicio: establecer viabilidad

En esta fase, principalmente se dirige al entendimiento de los requerimientos de desarrollo. Se define la idea, la visión y el alcance del proyecto. Su principal objetivo es el análisis del negocio: casos de uso fundamentales para el negocio.

Las actividades que deben realizarse en esta fase:

1. Delimitar el ámbito (interfases con los sistemas).
2. Proponer una arquitectura especialmente en lo nuevo, arriesgado o difícil (expresada en función de algunos modelos).
3. Identificar riesgos críticos (los que afecten a la viabilidad).
4. Demostrar a usuarios y clientes un prototipo [RUP,2000].

En esta fase se pretende establecer los objetivos del ciclo de vida para el proyecto.

#### 2.9.5.2. Fase de elaboración: factibilidad

El objetivo principal de la fase de elaboración es formular la línea base de la arquitectura. Esto implica desarrollar alrededor de 80 por ciento de los casos de uso y abordar los riesgos

que interfieren en la consecución de este objetivo. En esta fase, se acrecienta el entorno de desarrollo, no sólo para llevar a cabo las actividades de esta fase, sino para estar preparados para la fase de construcción. Hacia el final de esta fase, habremos acumulado la información necesaria para planificar la fase de construcción.

Entre las actividades que se desarrollan en esta fase se tienen:

1. Línea base de la arquitectura. Consiste en: modelos, descripción de la arquitectura e implementación ejecutable de la arquitectura.
2. Identificación de riesgos que puedan perturbar los planes y costes posteriores.
3. Especificar niveles para los atributos de calidad, fiabilidad y tiempo de respuesta.
4. Recopilar casos de uso para el 80% de los requisitos funcionales para planificar la fase de construcción.
5. Planificación: personal, coste [RUP,2000].

### **2.9.5.3. Fase de construcción**

El análisis de esta fase se traslada de la acumulación del conocimiento básico necesario para construir el proyecto a la construcción propiamente dicha de un sistema o producto dentro de unos parámetros de coste, esfuerzo y agenda.

Durante la fase de construcción, el jefe del proyecto, el arquitecto y los desarrolladores se aseguran de establecer la prioridad de los casos de uso, que son agrupados en construcciones e iteraciones y desarrollados según un orden que evita las vueltas atrás. Sus principales actividades son:

1. Terminar la identificación, descripción y realización de todos los casos de uso.
2. Finalizar el análisis, diseño, implementación y pruebas.
3. Mantener la integridad de la arquitectura.
4. Monitorizar riesgos críticos [RUP,2000].

#### **2.9.5.4. Fase de transición: en el entorno de usuario**

Esta fase se centra en implantar el producto en su entorno de operación. La forma en que el proyecto lleva acabo este objetivo varía con la naturaleza de la relación del producto con su mercado.

En esta fase no se busca reformular el producto. Por el contrario, el equipo busca pequeñas diferencias que pasaron desapercibidas durante la fase de construcción y que pueden ser corregidas en el marco de la línea base de la arquitectura existente.

Las actividades principales en esta fase son:

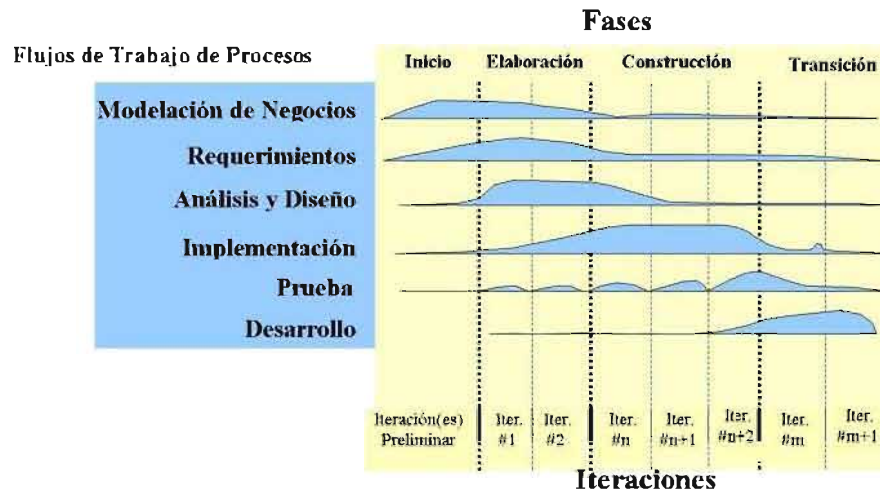
1. Corregir defectos detectados en la versión beta.
  - Lecciones aprendidas
  - Asuntos útiles para la versión siguiente [RUP,2000].
2. Preparar actividades, por ejemplo, el lugar de ejecución del sistema.
3. Aconsejar sobre el entorno de funcionamiento.
4. Manuales y documentos para la entrega del sistema.
5. Ajustar el software al entorno del usuario.

Al concluir esta fase podemos garantizar que tenemos un producto preparado para la entrega a la comunidad de usuarios.

#### **2.9.6. Modelos y flujos de trabajo**

El modo en el que describimos un proceso es en términos de flujos de trabajo (ver figura 2.16.), entendiéndose como un conjunto de actividades.

Se identifica primero los distintos tipos de trabajadores que participan en el proceso. Después identificamos los artefactos que necesitamos crear durante el proceso para cada tipo de trabajador. Una vez que los hemos identificado, podemos describir cómo fluye el proceso a través de los diferentes trabajadores, y como ellos crean, producen y utilizan los artefactos [RUP,2000].



**Fig. 2.16. Flujos de Trabajo**  
Fuente: [RUP,2005].

### 2.9.6.1. Modelado Del Negocio

El primer paso del modelado del negocio consiste en capturar los procesos de negocio en la organización bajo estudio. La definición del conjunto de procesos del negocio es una tarea crucial, ya que define los límites del proceso de modelado posterior.

Capturamos los procesos del negocio a partir de los objetivos principales de la empresa. En primer lugar, consideramos los objetivos estratégicos de la organización. Teniendo en cuenta que estos objetivos van a ser muy complejos y de un nivel de abstracción muy alto, serán descompuestos en un conjunto de subobjetivos más concretos. Para cada uno de estos subobjetivos de segundo nivel definimos un proceso de negocio que deberá dar soporte a dicho subobjetivo. Representa cada proceso del negocio como un caso de uso del negocio, que inicialmente será descrito en forma textual.

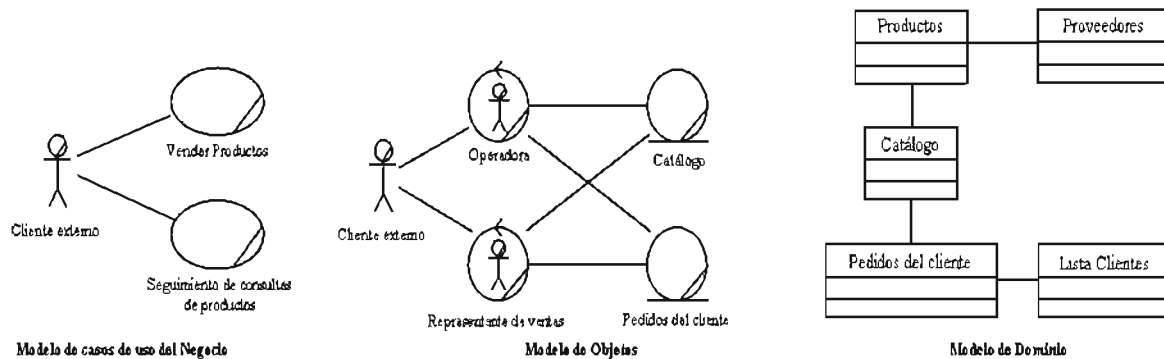
En el flujo de modelado de trabajo de **modelado del negocio** [LOPEZ,2008] se definen los siguientes modelos: modelo del negocio, modelos de objetos del negocio y modelo del dominio (ver figura 2.17.).

Para tener una visión general de los diferentes procesos de negocio de la organización, puede construirse un diagrama de casos de uso del negocio, en el cual aparece cada proceso del negocio como un caso de uso. Este diagrama permite mostrar los límites y el

entorno de la organización bajo estudio. Sólo se mostrarán en este diagrama los actores del negocio correspondientes a los miembros externos al sistema, de forma que los procesos de negocio en los que sólo tomen parte roles internos de la organización no están conectados a ningún actor del negocio.

El modelo de objetos es un modelo interno al negocio. Describe cómo cada caso de uso de negocio, es llevado a cabo por parte de un conjunto de trabajadores que utilizan un conjunto de entidades del negocio y de unidades de trabajo.

El objetivo del modelado del dominio es comprender y describir las clases más importantes dentro del contexto del negocio.



**Fig. 2.17. Modelos resultantes del flujo de trabajo modelado del negocio**  
Fuente: [RUP,2005].

### 2.9.6.2. Requisitos

El esfuerzo principal en la fase de requisitos es desarrollar un modelo de sistema que se va a construir, la utilización de casos de uso es una forma adecuada de crear ese modelo. Mediante la utilización de los casos de uso, los analistas se ven obligados a pensar en términos de quienes son los usuarios y que necesidades u objetivos de la empresa pueden cumplir [RUP,2000].

Es importante subrayar que el flujo de trabajo de Requisitos es el más importante dentro de la fase de elaboración del RUP.



Para la captura de requisitos es necesario utilizar los siguientes artefactos (ver figura 2.18.):

**Actor:** Un actor es un conjunto coherente de roles que desempeñan los usuarios de los casos de uso cuando interactúan con estos. Por ejemplo usuarios, sistemas externos que interactúan con el sistema, dispositivos externos, etc.

**Caso de uso:** Un caso de uso especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores.

**Modelo de casos de uso:** Es un modelo de sistema que resume los casos de uso, sus actores y sus relaciones. Este modelo sirve como acuerdo entre clientes y desarrolladores, y proporciona la entrada para los flujos de análisis, diseño y pruebas.



**Fig. 2.18. Artefactos del grupo de trabajo Requisitos**  
Fuente: [RUP,2005].

**Especificación de casos de uso:** El flujo de sucesos para cada caso de uso puede plasmarse como una descripción textual de la secuencia de acciones del caso de uso. Por tanto el flujo de sucesos especifica lo que el sistema hace y como interactúa con los actores cuando se lleva a cabo los casos de uso. El objetivo principal de detallar cada caso de uso es describir su flujo de procesos en detalle incluyendo cómo comienza, termina e interactúa con los actores. Es aconsejable ordenar los datos en el siguiente formato para hacer una especificación de casos de uso completa y correcta (ver tabla 2.2).

<b>Nombre.</b>		
<b>Actores.</b>		
<b>Propósito.</b>		
<b>Descripción.</b>		
	<b>Eventos Actor</b>	<b>Eventos Sistema</b>
<b>Flujo Principal.</b>		
<b>Flujo Alternativo.</b>		
<b>Precondición</b>		
<b>Poscondición</b>		

**Tabla 2.2 Formato de especificación de casos de uso**  
Fuente: [RUP,2005].

### 2.9.6.3. Análisis

Durante el análisis se estudian los requisitos que se describieron en la captura de requisitos, refinándolos y estructurándolos. El objetivo es conseguir una comprensión precisa de los requisitos, para plasmarlos en casos de uso, mismos que serán representados desde la perspectiva de los desarrolladores, ésta vista estará centrada en el sistema, y no en el problema.

Un modelo de análisis ofrece una especificación precisa de los requisitos que tenemos como resultado de la captura de requisitos, incluyendo el modelo de casos de uso. Un modelo de análisis que describe utilizando el lenguaje de los desarrolladores, por tanto, si se introduce un mayor formalismo puede ser utilizado para razonar sobre los funcionamientos internos del sistema. Además el modelo de análisis estructura los requisitos de modo que facilita su comprensión, su preparación, su modificación, y en general, su mantenimiento.

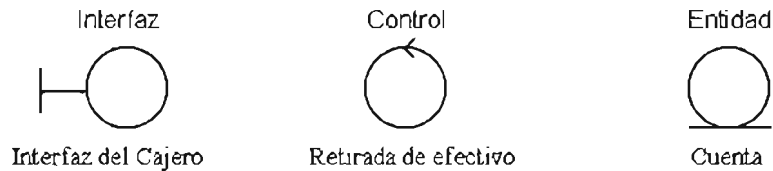
A continuación se detallan los artefactos que se utilizan para desarrollar el flujo de trabajo de análisis.

**Modelo de análisis** se presenta mediante un sistema de análisis. Dentro del modelo de análisis, los casos de uso se describen mediante clases de análisis y sus objetivos. Esto se presenta dentro del modelo de análisis que llamamos *realizaciones de casos de uso-análisis*. Los artefactos del modelo de análisis se describen en detalle a continuación.

Las clases de análisis se clasifican en uno de tres estereotipos básicos: de interfaz, de control o de entidad (ver figura 2.19). Cada estereotipo indica una semántica específica, lo cual constituye un método potente de identificación y descripción de las clases de análisis.

**La Clase de interfaz** se utiliza para modelar la interfaz entre el sistema y sus actores. Representan abstracciones de ventanas, formularios, paneles, interfaces de comunicación, interfaces de impresoras, sensores, terminales y API.

Cada clase de interfaz debería asociarse con al menos un actor y viceversa.



**Fig. 2.19. Estereotipos Básicos**  
Fuente: [RUP,2005].

La **Clase de entidad** se utiliza para modelar información que posee una larga vida y que a menudo es persistente. Modela información y el comportamiento asociado a algún fenómeno o concepto.

La **Clase control** representa coordinación, secuencia, transacciones y control de otros objetos y se usan con frecuencia para encapsular el control de un caso de uso en concreto. Sirven para modelar los aspectos dinámicos del sistema.

La **realización de caso de uso-análisis** es una colaboración dentro del modelo de análisis que describe como se lleva a cabo y se ejecuta un caso de uso determinado en términos de las clases del análisis y de sus objetos del análisis en interacción.

Se utiliza el **diagrama de colaboración**, ya que lo que se busca es identificar requisitos y responsabilidades sobre los objetos y no secuencia de interacción detallada, como en el diagrama de secuencia.

#### 2.9.6.4. Diseño

En el diseño modelamos el sistema y encontramos su forma para que soporte todos los requisitos. Una entrada fundamental del diseño es el modelo de análisis, que proporciona una comprensión detallada de los requisitos e impone una estructura al sistema.

Los objetivos del diseño son:

- Crear una entrada apropiada y un punto de partida para actividades de implementación subsiguientes.

- Ser capaces de descomponer los trabajos de implementación en partes manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo, teniendo en cuenta la posible concurrencia [RUP,2000].

Durante el desarrollo del diseño se utilizan los siguientes artefactos:

La **Realización de Casos de Uso-Diseño** describe cómo un caso de uso se lleva a cabo en términos de clases de diseño y sus objetivos. Hay una correspondencia directa entre la realización de un caso de uso de diseño y la realización de un caso de uso de análisis.

- **Diagrama de Clases** que contiene las clases que participan en el caso de uso, aunque algunas de ellas puedan participar en varios.
- **Diagrama de Secuencia** que muestra una secuencia detallada de interacción entre objetos de diseño. En algunos casos es posible incluir subsistemas dentro de los diagramas. Los diagramas de secuencia visualizan el intercambio de mensajes entre objetos.

El **Modelo de Diseño** es un conjunto formado por las Clases de Diseño y la Realización de los Casos de Uso.

**Modelo de Despliegue** que describe la distribución física del sistema en términos de cómo las funcionalidades se distribuyen entre los nodos de computación sobre los cuales se va a instalar el sistema.

- Cada nodo representa un recurso de computación.
- Los nodos tienen relaciones entre ellos que representan los medios de comunicación que hay entre ellos como una Intranet o Internet.
- El modelo de despliegue representa un manejo claro entre la arquitectura software y hardware.

### **Identificación de subsistemas y sus interfaces**

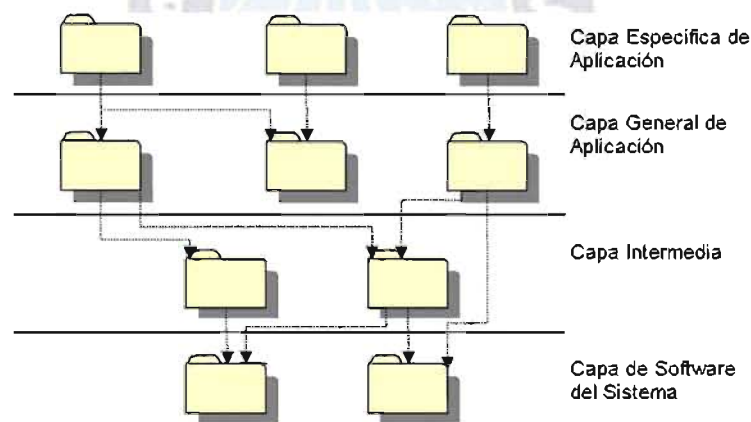
Los subsistemas constituyen un medio para organizar el modelo de diseño en piezas manejables. Pueden bien identificarse inicialmente como forma de dividir el trabajo de

diseño, o bien puede irse encontrando a medida que el modelo de diseño evoluciona y va “creciendo” hasta convertirse en una gran estructura que debe ser descompuesta.

Se deben identificar los *subsistemas de aplicación*, donde resalta los subsistemas de las capas específicas y en general de aplicación.

La identificación de *subsistemas intermedios* y de *software del sistema* constituyen los cimientos de un sistema, ya que toda funcionalidad destaca sobre el software, sistemas operativos, sistemas de gestión de bases de datos, software de comunicaciones, tecnologías de distribución de objetos, kits de diseño de IGU, y tecnologías de gestión de transacciones.

En la Fig. 2.20 se muestra los subsistemas de capas de aplicación, intermedia y de software del sistema.



**Fig. 2.20. Subsistemas en capas de aplicación, intermedia y software del sistema**  
Fuente: [RUP,2005].

### 2.9.6.5. Implementación

En la implementación empezamos con el resultado de la fase de diseño e implementamos el sistema en términos de *componentes* tales como ficheros frente, ejecutables, scripts, etc. Los objetivos de la implementación son:

- Implementar las clases encontradas durante el diseño. En concreto, se implementan dentro de componentes (ficheros) que contienen código fuente.
- Asignar los componentes ejecutables a los nudos del diagrama de despliegue.
- Probar los componentes individualmente e integrarlos en uno o más ejecutables.
- Integrar los componentes en el sistema siguiendo un enfoque incremental.

Los artefactos que se usarán durante el flujo de trabajo de implementación son:

Los **Componentes** son empaquetados físicos de elegidos de un modelo. Algunos ejemplos de componentes pueden ser ficheros con código fuente de una o varias clases, ejecutables, librerías, tablas de bases de datos, documentos, entre otros.

El **Modelo de Implementación** describe cómo los elementos del diseño, se implementan en términos de componentes de acuerdo a las estructuras y los mecanismos de modularización que se dispongan el entorno y lenguaje de programación elegido, y cómo dependen los componentes los unos de los otros.

Los propósitos de la implementación son:

- Planificar las integraciones de sistema necesarias en cada iteración.
- Distribuir el sistema asignando componentes ejecutables a nodos en el diagrama de despliegue.
- Implementar las clases y subsistemas encontrados durante el diseño.
- Probar los componentes individualmente e integrarlos.

La implementación es el centro durante las iteraciones de construcción.

También se lleva a cabo el trabajo de implementación durante la fase de elaboración, para crear la línea base ejecutable de la arquitectura. Durante la fase de transición puede tratar efectos tardíos ya que el modelo de implementación denota la implementación actual del sistema en términos de componentes y subsistemas de implementación, es importante mantener el modelo de implementación a lo largo de todo el ciclo de vida del software [RUP,2000].

## 2.10. UML MAST METAMODEL

UML Mast es una metodología y un conjunto de herramientas gráficas desarrolladas para modelar y analizar sistemas de tiempo real que están siendo desarrollados utilizando métodos orientados a objetos sobre herramientas CASE basadas en la notación UML. Los



componentes de modelado y las herramientas de análisis proceden del entorno MAST (Modeling and Analysis Suit for Real Time Applications).

El modelo de tiempo real del sistema que se desarrolla se formula mediante la "Mast RT View". Esta es una vista complementaria de la descripción UML del sistema que modela la capacidad de la plataforma que se utiliza, las características de temporización de los componentes lógicos de su software y las transacciones que pueden ocurrir con los requerimientos temporales que en ellas se establecen. Con la Mast RT View el diseñador puede construir gradualmente el modelo de tiempo real del sistema que diseña, de forma paralela al desarrollo de su modelo lógico.

El metamodelo constituye la descripción básica y formal de la Mast RT View. En este documento se describe con notación UML este metamodelo y con él se definen:

1. Se define la estructura general del modelo, estableciendo sus secciones, los aspectos del sistema que describe y los componentes con que se construyen.
2. Se describen los componentes que se utilizan para construir el modelo de tiempo real, definiendo su naturaleza así como los atributos que describen su comportamiento cuantitativo.
3. Se describen las relaciones que se pueden establecer entre los componentes dentro del modelo.
4. En el modelo, a través de estereotipos se indica el tipo de componentes UML (paquete, clase, atributo, actividad, acción, evento, etc.) con el que se instancia cada componente del metamodelo en el modelo concreto de un sistema.

La descripción del metamodelo se realiza a través de diagramas de clases que describen gráficamente su estructura, y de párrafos de texto que describen conceptualmente cada componente así como sus atributos. A fin de minimizar dependencias cruzadas en la descripción de cada diagrama, se incluye junto a cada uno la mayoría de las descripciones de sus clases, aunque ello implique la repetición de clases a lo largo del documento.

**Process View:** Es una de las vista del "paradigma 4+1" que se usa para describir la performance, escalabilidad, y throughput de un sistema bajo desarrollo en una herramienta CASE UML.

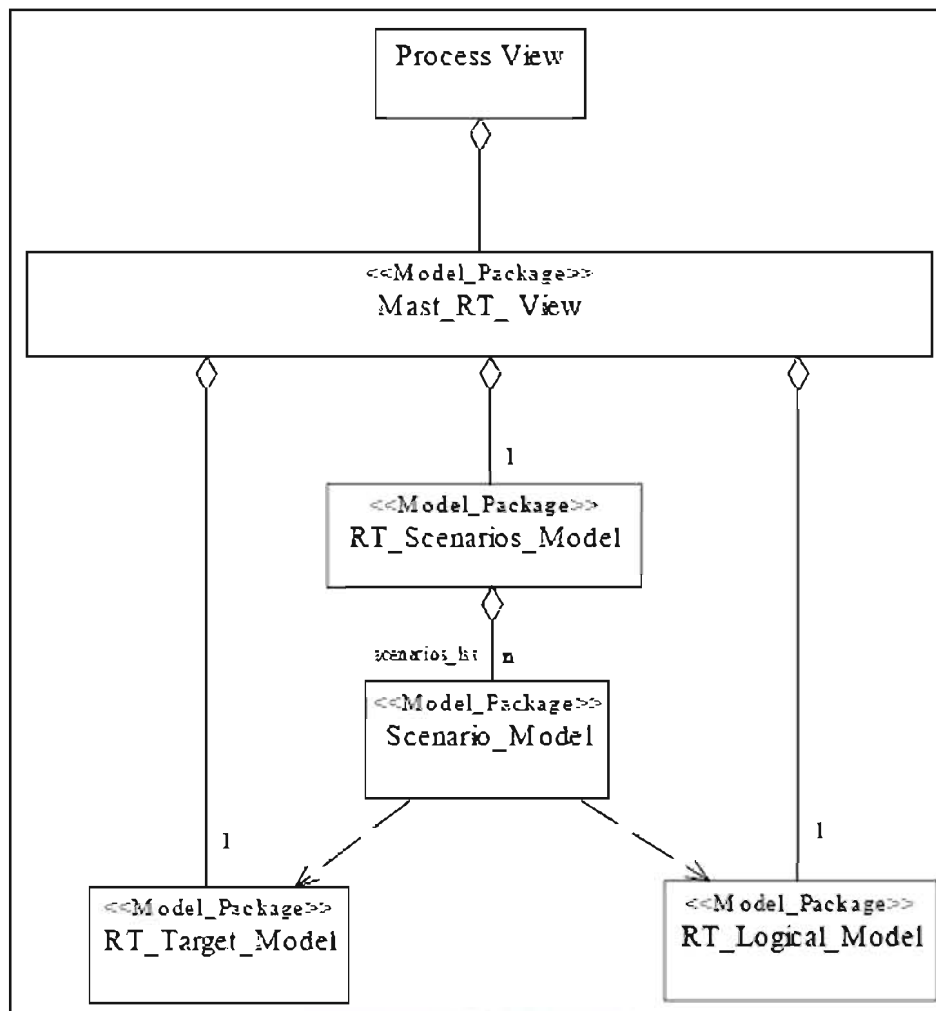


Fig. 2.21 Diagrama Mast RT View.  
Fuente: [Drake,2001]

**Mast\_RT\_View:** Es una vista adicional que complementa la descripción estándar UML de un sistema durante su fase de desarrollo. Su finalidad es definir un modelo de tiempo real que describa la temporización y los recursos que se asignan a las actividades del sistema y sirve de base para analizar su planificabilidad mediante herramientas automáticas.

Se implementa como el paquete raíz dentro del que se definen todos los componentes del modelo de tiempo real. Este paquete debe estar definido directamente sobre la Logical View a efecto de que pueda ser localizada por la herramienta automática.

**RT\_Target\_Model:** Modela la capacidad de procesamiento y las restricciones operativas de los recursos de procesamiento hardware y software que constituyen la plataforma sobre la que se ejecuta el sistema.

**RT\_Logical\_Model:** Modela los requerimientos de procesado que requiere la ejecución de las operaciones funcionales definidas en los componentes lógicos que se utilizan en el diseño, tales como métodos, procedimientos y funciones definidos en las clases, primitivas de sincronización entre threads, sesiones de comunicación, etc.

Describe el comportamiento de tiempo real de los componentes funcionales (clases, métodos, procedimientos, operaciones, etc.) que están definidos en el sistema y cuyos tiempos de ejecución condicionan el cumplimiento de sus especificaciones de tiempo real.

El RT\_Logical\_Model modela dos aspectos de un componente:

- La complejidad de los algoritmos con que se implementa.
- Las posibilidades de bloqueo que pueden retrasar su ejecución y que son consecuencia de tener que hacer uso exclusivo de recursos compartidos . (Shared\_Resource).

**RT\_Scenarios\_Model:** Conjunto de configuraciones o de modos de operación del sistema en los que hay definidas especificaciones de tiempo real. Se implementa como un paquete que debe estar definido directamente dentro del paquete Mast\_RT\_View.

### **Scenario\_Model**

Modela cada configuración o modo de operación que puede ser alcanzado por el sistema en el que estén establecidos requerimientos de tiempo real.

Representa una carga concreta de trabajo (workload) para la que deben satisfacerse un conjunto de requerimientos de tiempo real. Está constituido por el conjunto de los modelos de las transacciones que concurren en él. En un modelo cada Scenario\_Model se representa mediante un package con el nombre del escenario y se define en el package RT\_Scenarios\_Model.

## 2.11. HERRAMIENTAS DE PRUEBAS

En una de las últimas fases del ciclo de vida antes de entregar un programa para su exploración, es la fase de pruebas.

Cuando se considera que un módulo está terminado se realizan las pruebas sistemáticas, el objeto de estas es buscar fallos a través de un criterio específico, estos criterios se denominan “pruebas de caja negra y de caja blanca” [D’onofrio,2007].

### 2.11.1. Pruebas de caja Blanca

A este tipo de técnicas se conoce también como Técnicas de Caja Transparente o de Cristal. Este método se centra en cómo diseñar los casos de prueba atendiendo al comportamiento interno y la estructura del programa. Se examina así la lógica interna del programa sin considerar los aspectos de rendimiento.

El objetivo de la técnica es diseñar casos de prueba para que se ejecuten, al menos una vez, todas las sentencias del programa, y todas sus condiciones tanto en su vertiente verdadera como falsa. Por ello, se han definido distintos criterios de cobertura lógica, que permiten decidir que sentencias o caminos se deben examinar con los casos de prueba. Estos criterios son:

- **Cobertura de Sentencias:** Se describen casos de prueba suficientes para que cada sentencia en el programa se ejecute, al menos una vez.
- **Cobertura de Decisión:** Se describen casos de prueba suficientes para que cada decisión en el programa se ejecute una vez con resultado verdadero y otra con el falso.
- **Cobertura de Condiciones:** Se describen casos de prueba suficientes para que cada condición en una decisión tenga una vez resultado verdadero y otra falso.
- **Cobertura Decisión/Condición:** Se describen casos de prueba suficientes para que cada condición en una decisión tome todas las posibles salidas, al menos una vez, y cada decisión tome todas las posibles salidas, al menos una vez.
- **Cobertura de Condición Múltiple:** Se describen casos de prueba suficientes para que todas las combinaciones posibles de resultados de cada condición se invoquen al menos una vez.

- **Cobertura de Caminos:** Se describen casos de prueba suficientes para que se ejecuten todos los caminos de un programa. Entendiendo camino como una secuencia de sentencias encadenadas desde la entrada del programa hasta su salida.

La aplicación del criterio **Cobertura de Caminos** asegura que los casos de prueba diseñados permitan que todas las sentencias del programa sean ejecutadas al menos una vez y que las condiciones sean probadas tanto para su valor verdadero como falso.

Una de las técnicas empleadas para aplicar este criterio de cobertura es la **Prueba del Camino Básico**. Esta técnica se basa en obtener una medida de la complejidad del diseño procedimental de un programa (o de la lógica del programa) [Sevilla,2005].



## CAPITULO III MODELO TEORICO

### 3.1. REQUERIMIENTOS DEL SISTEMA

Partiendo de la identificación de problemas se pudo establecer los siguientes requerimientos :

- Percibir y almacenar los eventos que envía el Receptor D6600
- Realizar el control de horarios.
- Generar alertas sonoras para cada tipo de evento .
- Desplegar la información de una cuenta seleccionada juntamente con los últimos # reportes de la cuenta.

A partir de los requerimientos establecidos dividiremos el sistema en los siguientes módulos:

- Percibir y almacenar los eventos recepcionados
- Control de horarios
- Generar alertas sonoras
- Elaborar reportes
- Despliegue de información

### 3.2. IDENTIFICACION DE LOS CASOS DE USO

En el monitoreo de sistemas de seguridad electrónicas se pudieron identificar los siguientes actores y casos de uso, ver Fig. 3.1:

#### 3.2.1. Descripción de Actores

##### a) Receptor D6600

Es el encargado de la comunicación vía línea telefónica con el panel de alarma, Percibir los eventos y filtrarlos.



- Percibir los eventos enviados por el Panel de alarma.
- Enviar al Sistema los eventos filtrados.

#### **b) Usuario del Panel de Alarma**

Es el usuario que manipula el Panel de Alarma, (esta dotado de una clave numérica y una palabra clave), puede ser un usuario de cuenta tipo 1 o tipo 2.

- Interactuar con el Panel de Alarma.
- Amar/Desarmar el Panel de Alarma.
- Aclarar de forma narrativa los sucesos acontecidos en el inmueble que generaron una señal de alarma.

#### **c) Operador de Central**

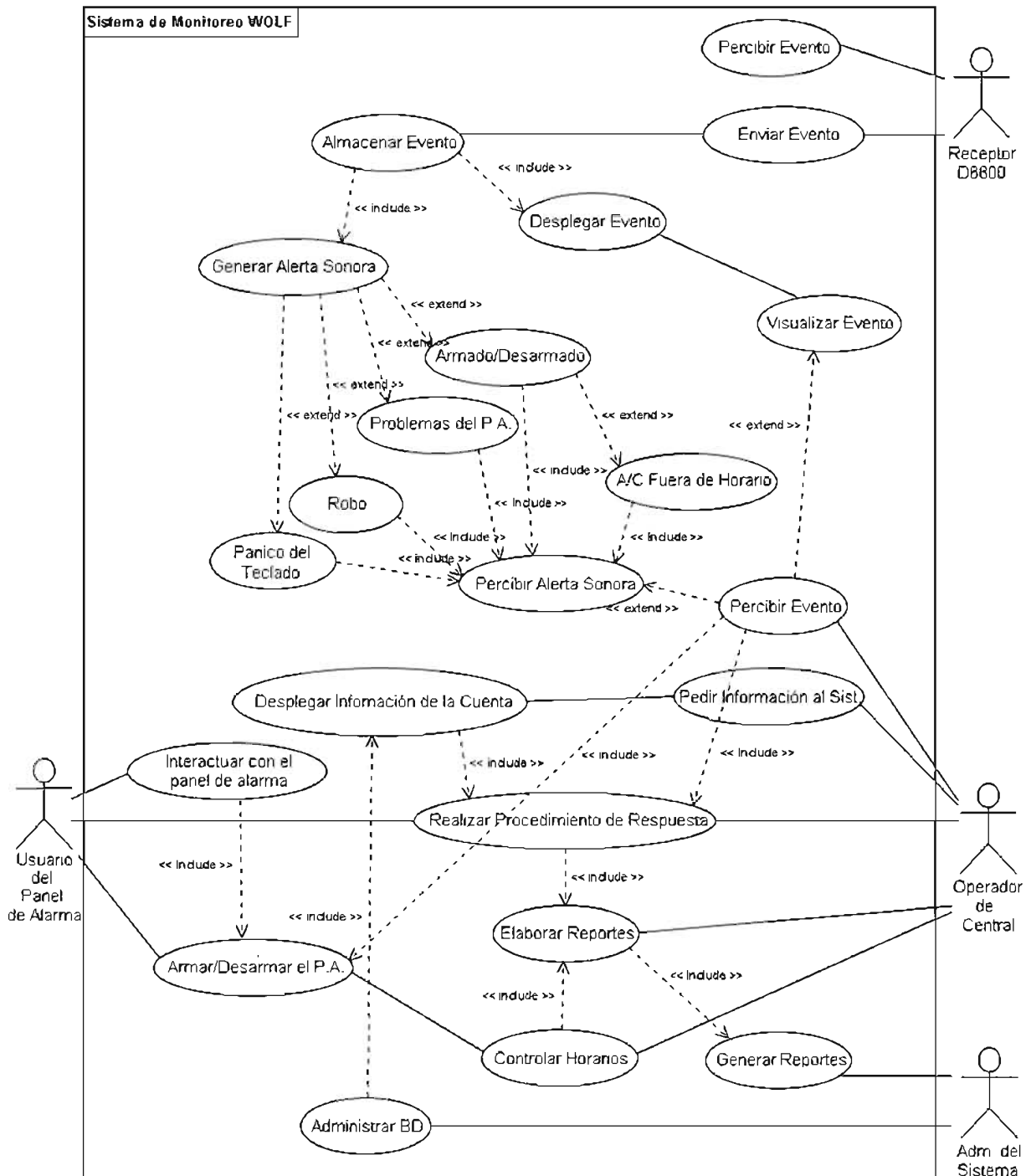
Representa a la persona que esta encargada de la central de monitoreo, y realizar el procedimiento de respuesta ante un evento como ser: Comunicarse con el cliente, Patrullero, etc.

- Prestar atención a los eventos entrantes y a los sonidos emitidos por el sistema .
- Controlar aperturas y cierres.
- Realizar el procedimiento establecido dentro de la empresa para cada tipo de evento .
- Elaborar los reportes de acuerdo a los relatos recabados del cliente, usuario del panel de alarma, patrullero, policía, etc.

#### **d) Administrador del Sistema**

El Administrador del Sistema representa al apersona responsable de administrar el sistema

- Adicionar nuevas cuentas.
- Modificas los datos de la Base de Datos
- Eliminar las cuentas que decidan dejar de recibir el servicio de monitoreo.
- Dar de Alta/Baja a un Operador de Central



### 3.2.2. Descripción De Casos De Uso

#### 3.2.2.1. Percibir y Almacenar Eventos (ver Fig. 3.2.)

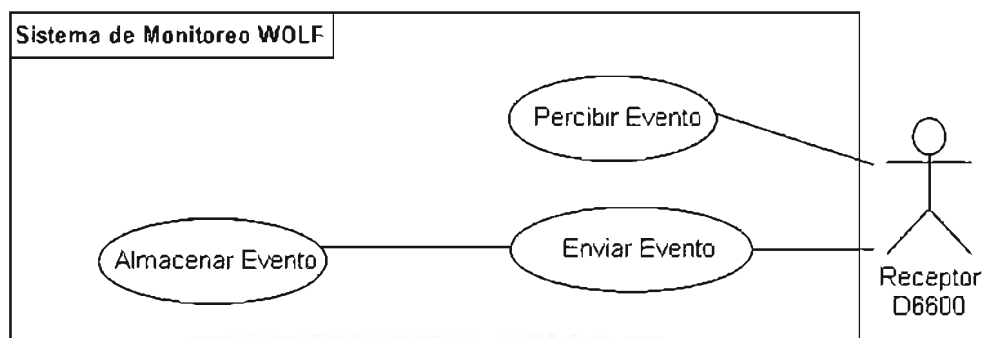
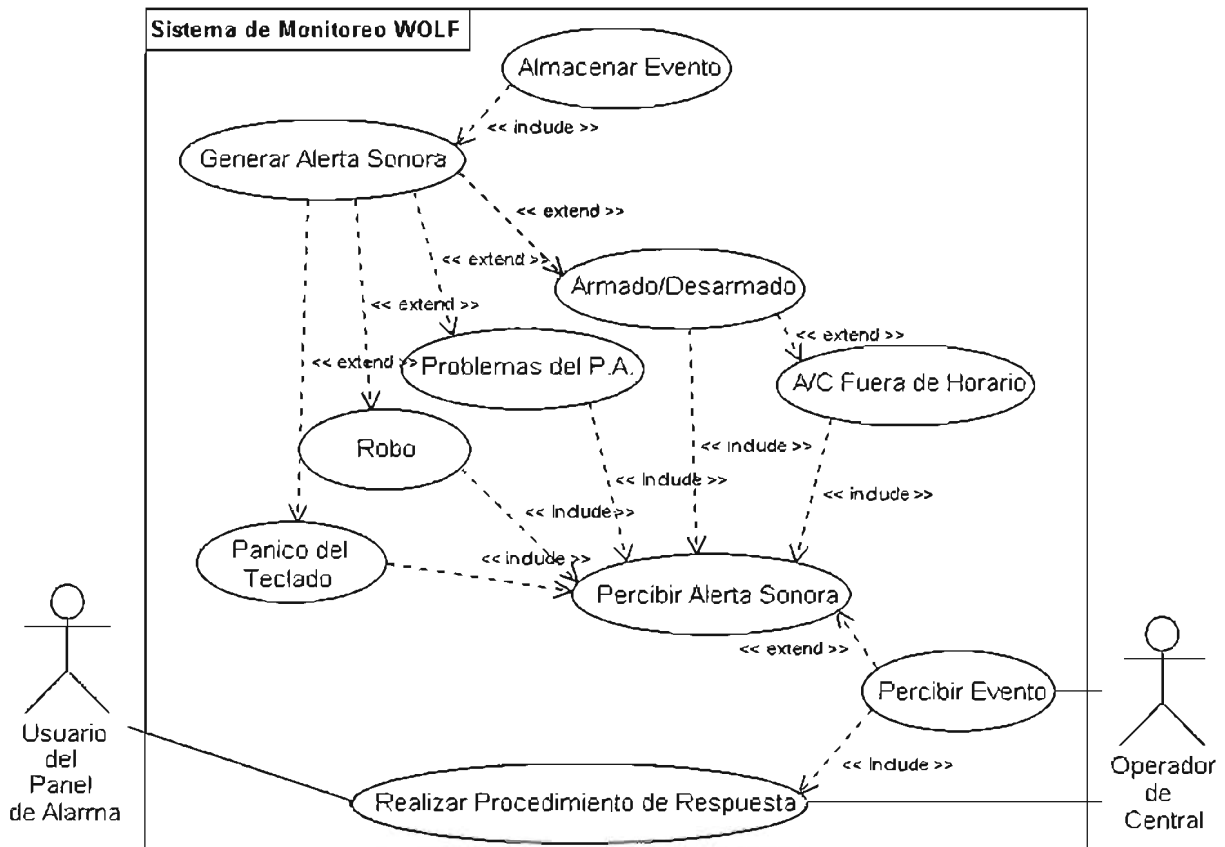


Fig. 3.2. Modelo de Casos de Uso: Percibir y almacenar eventos  
Fuente: Elaboración propia.

<b>Nombre.</b>	Percibir y almacenar eventos	
<b>Actores.</b>	Receptor D6600 (R. D6600)	
<b>Propósito.</b>	Percibir y almacenar un nuevo evento enviado por un panel de alarma.	
<b>Descripción.</b>	El R. D6600 recibe y filtra el evento, lo envía al sistema y se almacena en la base de datos del sistema.	
<b>Flujo Principal.</b>	<b>Eventos Actor</b>	<b>Eventos Sistema</b>
	1. El R. D6600 percibe el evento. 2. El R. D6600 envía un evento al sistema.	3. El sistema percibe el evento 4. Valida el evento. 5. Almacena el evento en la Base de Datos
<b>Flujo Alternativo.</b>	Paso 4. Si el evento no es valido, termina el caso de uso.	
<b>Precondición</b>	Un panel de alarma envía un evento al Receptor D6600.	
<b>Poscondición</b>	El Sistema deberá guardar correctamente el nuevo evento en el archivo de texto correspondiente a la fecha actual.	

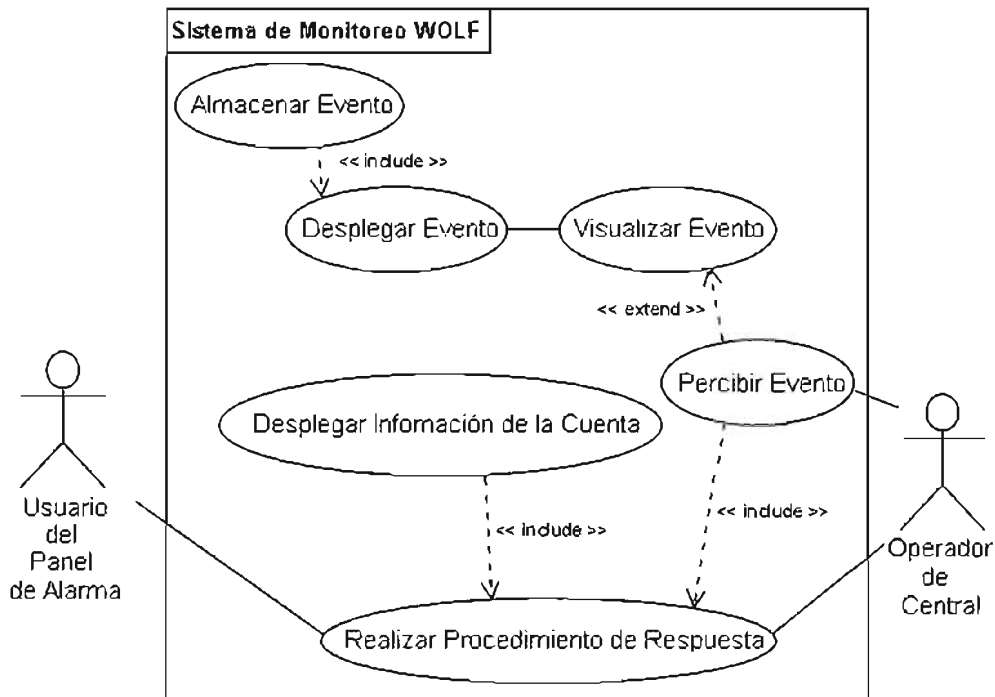
### 3.2.2.2. Percepción de la Alerta Sonora por el Operador de Central (Ver Fig. 3.3.)



**Fig. 3.3. Modelo de Casos de Uso: Percepción de la alerta sonora por el Operador de Central**  
**Fuente: Elaboración propia.**

<b>Nombre.</b>	Percepción de la alerta sonora por el Operador de Central	
<b>Actores.</b>	Usuario del Panel de Alarma (UPA), Operador de Central (OC)	
<b>Propósito.</b>	Alertar al OC de la llegada de cada evento de forma audible.	
<b>Descripción.</b>	Luego que el sistema almacena un evento, genera una alerta sonora que es percibida por el OC, luego este realiza su procedimiento de respuesta.	
<b>Flujo Principal.</b>	<b>Eventos Actor</b>	<b>Eventos Sistema</b>
	3. El OC percibe la alerta sonora. 4. Consulta información acerca de la cuenta 5. El OC realiza su procedimiento de respuesta con el UPA.	1. Almacena el evento. 2. Genera una alerta sonora diferente para cada tipo de evento.
<b>Flujo Alternativo.</b>	Paso 3. Si el sonido pertenece a un evento que no necesite procedimiento de respuesta, el OC no realizara los Pasos 4 y 5.	
<b>Precondición</b>	El OC debe estar en el ambiente de la Central de Monitoreo.	
<b>Poscondición</b>	El operador debe Elaborar un reporte del evento .	

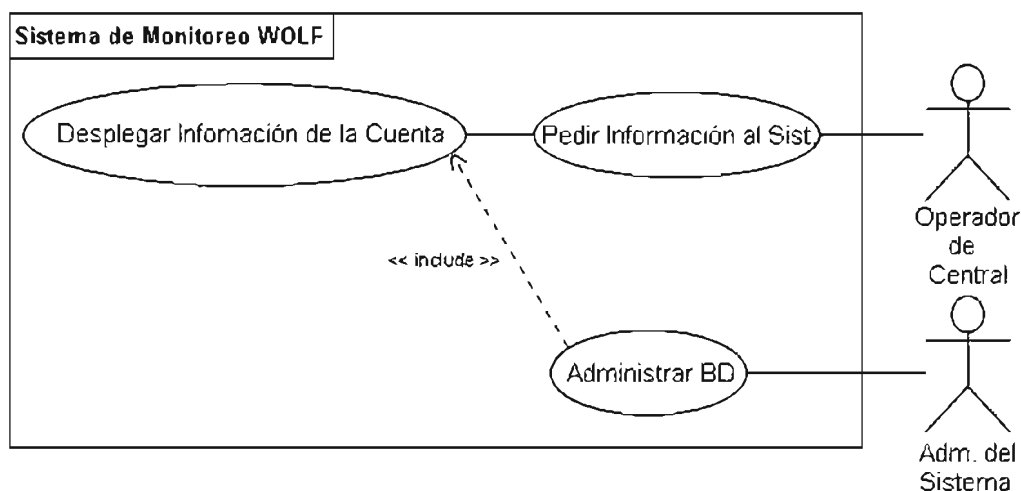
### 3.2.2.3. Percepción Visual del Evento por el Operador de Central (Ver Fig. 3.4.)



**Fig. 3.4. Modelo de Casos de Uso: Percepción visual.**  
Fuente: Elaboración propia.

<b>Nombre.</b>	Percepción visual del evento por el Operador de Central	
<b>Actores.</b>	Usuario del Panel de Alarma (UPA), Operador de Central (OC)	
<b>Propósito.</b>	Mostrar al OC de la llegada de cada evento de forma que pueda reconocer el evento de forma visual.	
<b>Descripción.</b>	Luego que el sistema almacena un evento, despliega el evento en pantalla, que es percibido por el OC y, a continuación realiza su procedimiento de respuesta.	
<b>Flujo Principal.</b>	<b>Eventos Actor</b>	<b>Eventos Sistema</b>
	4. El OC visualiza el evento en pantalla. 5. Consulta información de la cuenta. 6. El OC realiza su procedimiento de respuesta.	1. Almacena el evento 2. Despliega el evento en pantalla. 3. Genera un objeto temporal en la pantalla.
<b>Flujo Alternativo.</b>	Paso 3. Si el evento desplegado por el Sistema pertenece a un evento que no necesite realizar ningún procedimiento, el caso de uso termina.	
<b>Precondición</b>	El operador debe estar atento al monitor de la PC.	
<b>Poscondición</b>	El operador debe escribir un reporte del evento.	

### 3.2.2.4. Despliegue de la Información de la Cuenta (Ver Fig. 3.5.)

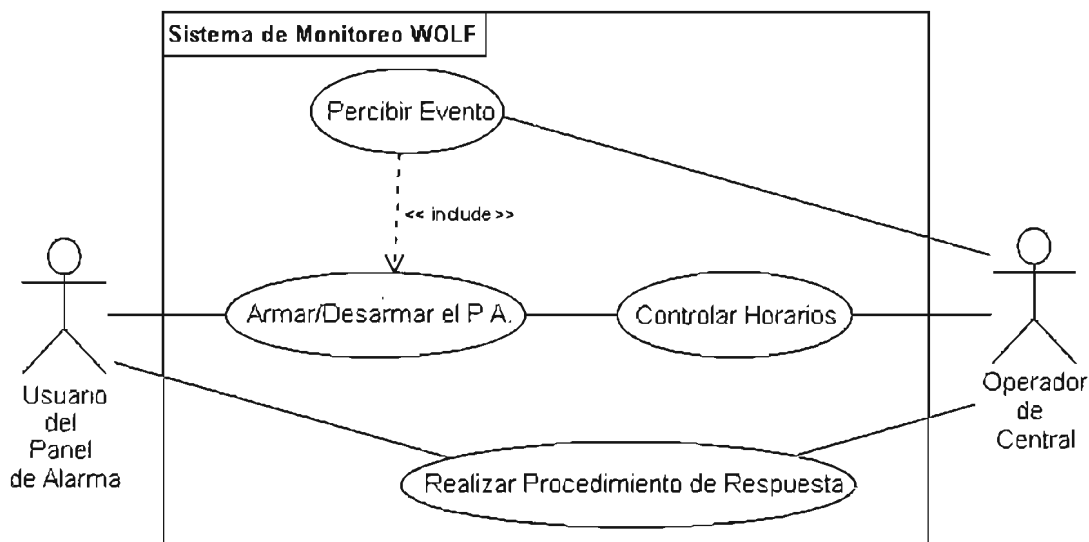


**Fig. 3.5. Modelo de Casos de Uso: Despliegue de la información de la cuenta**  
**Fuente: Elaboración propia.**

<b>Nombre.</b>	Despliegue de la información de la cuenta	
<b>Actores.</b>	Operador de Central (OC), Administrador del Sistema (Adm.)	
<b>Propósito.</b>	Desplegar la información necesaria para que el OC pueda realizar sus procedimientos de respuesta.	
<b>Descripción.</b>	El OC consultará información de la Base de Datos del Sistema que previamente fue Adicionada o modificada por el Adm. del Sistema.	
<b>Flujo Principal.</b>	<b>Eventos Actor</b>	<b>Eventos Sistema</b>
	1. El OC realiza un clic en el enlace de la cuenta, de la cual necesita la información. 3. El OC visualiza la información desplegada en pantalla.	2. El sistema despliega la información de la cuenta de la siguiente forma: * Datos de filiación. * Datos de los Usuarios. * Últimos eventos registrados. * Últimos reportes realizados. * Un TextBox para adicionar un nuevo reporte.
<b>Precondición</b>	El Adm. debe contar con la información necesaria de la cuenta para poder adicionar una nueva cuenta o modificar sus datos. El OC necesita realizar un procedimiento de respuesta o un reporte.	
<b>Poscondición</b>	El OC realiza su procedimiento de respuesta con los datos desplegados.	



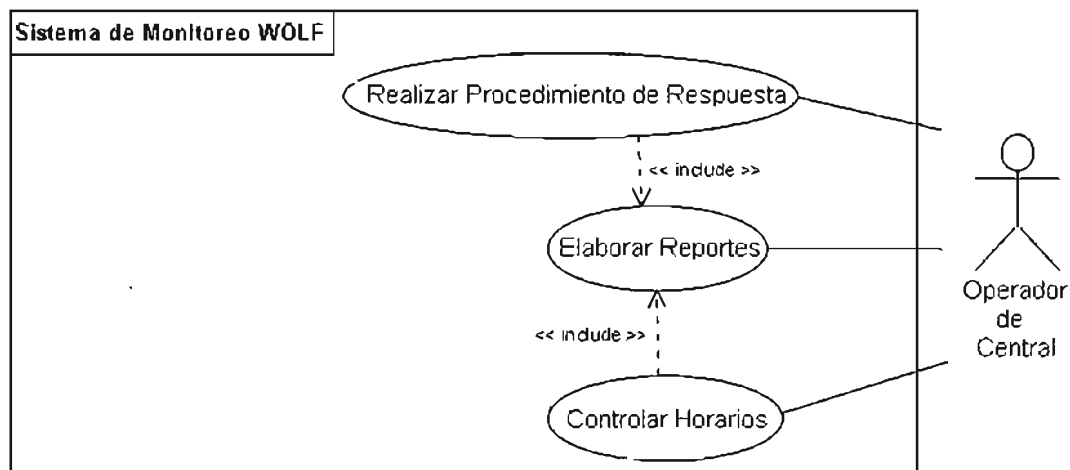
### 3.2.2.5. Controlar Horarios de Apertura y Cierre (Ver Fig. 3.6.)



**Fig. 3.6. Modelo de Casos de Uso: Controlar horarios de apertura y cierre**  
Fuente: Elaboración propia.

<b>Nombre.</b>	Controlar horarios de apertura y cierre	
<b>Actores.</b>	Usuario del Panel de Alarma (UPA), Operador de Central (OC).	
<b>Propósito.</b>	Realizar el control automático de los horarios de apertura y cierre de las cuentas que cuenten con este servicio.	
<b>Descripción.</b>	El UPA realiza un armado/desarmado de su panel de alarma, luego el OC percibe el evento y se asegura de que este evento este en el rango establecido para cada cuenta.	
<b>Flujo Principal.</b>	<b>Eventos Actor</b>	<b>Eventos Sistema</b>
	1. El UPA realiza un armado/desarmado de su panel de alarma. 3. El OC percibe el evento y realiza su procedimiento de respuesta.	2. El sistema emite una alerta sonora y despliega el evento en pantalla
<b>Flujo Alternativo.</b>	Paso 2. Si el evento de apertura/cierre no esta fuera del rango establecido, el sistema evite no emite señal de alerta.	
<b>Precondición</b>	El Adm. debe registrar los horarios de aperturas/cierres de la cuenta. El UPA debe estar dentro del inmueble o alrededor (usando un control remoto), y realizar un armado o desarmado del panel de alarma.	
<b>Poscondición</b>	El OC debe elaborar un reporte si es necesario.	

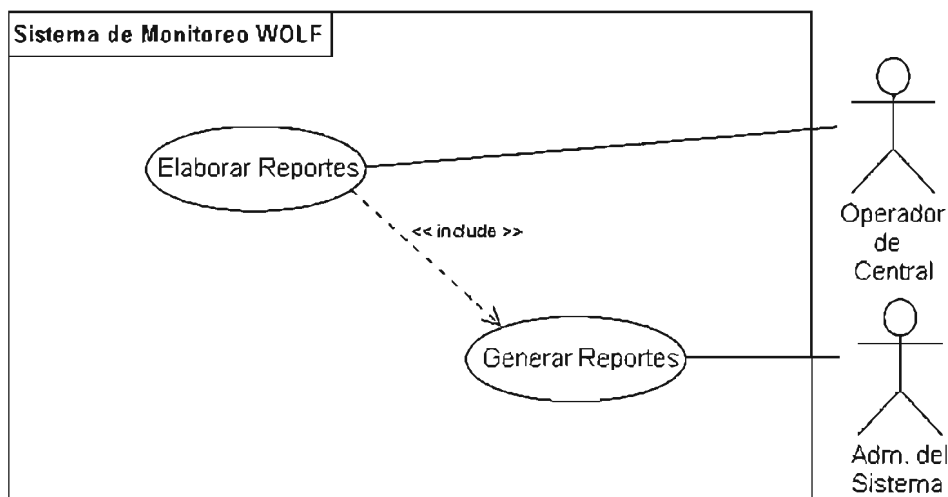
### 3.2.2.6. Elaboración de Reportes (Ver Fig. 3.7.)



**Fig. 3.7. Modelo de Casos de Uso: Elaboración de reportes.**  
**Fuente: Elaboración propia.**

<b>Nombre.</b>	Elaboración de reportes	
<b>Actores.</b>	Operador de Central (OC)	
<b>Propósito.</b>	Elaboración de reporte por el OC (de cada evento que requirió de un procedimiento de respuesta) directamente en el Sistema para luego ser almacenados en la Base de Datos	
<b>Descripción.</b>	Un operador debe elaborar un reporte de los eventos que necesitaron de un procedimiento de respuesta.	
<b>Flujo Principal.</b>	<b>Eventos Actor</b>	<b>Eventos Sistema</b>
	1. El OC debe realizar un procedimiento de respuesta. 2. El OC realiza un clic sobre el enlace de la cuenta a la que añadirá un reporte. 4. El OC escribe el reporte en el TextBox. 5. El OC Elige "Guardar reporte".	3. El sistema despliega la información de la cuenta. Juntamente con un TextBox para escribir el reporte. 6. El Sistema guarda el reporte en la base de datos
<b>Flujo Alternativo.</b>	Paso 1. Si se da el caso que el cliente o UPA notificara con anticipación su apertura el Paso uno de debe obviar y continuar con el Paso 2.	
<b>Precondición</b>	El OC debe contar con al menos un relato de los siguientes: Usuario del Panel de Alarma, Cliente, Patrullero, Policía, Bomberos, etc.	
<b>Poscondición</b>	El Sistema deberá guardar correctamente el nuevo reporte en la Tabla "Reporte".	

### 3.2.2.7. Generar Reportes (Ver Fig. 3.8.)



**Fig. 3.8. Modelo de Casos de Uso : Generar Reportes.**  
**Fuente: Elaboración propia.**

<b>Nombre.</b>	Generar Reportes	
<b>Actores.</b>	Administrador del Sistema (Adm.), Operador de Central (OC)	
<b>Propósito.</b>	Generar reportes de los eventos que requirieron un procedimiento de respuesta	
<b>Descripción.</b>	El Adm. Ingresa al módulo de administración del sistema generar los reportes	
<b>Flujo Principal.</b>	<b>Eventos Actor</b>	<b>Eventos Sistema</b>
	1. El Adm. ingresa su contraseña. 4. El Adm. selecciona la opción generar reportes. 6. El Adm. Verifica los reportes y los envía al cliente.	2. El Sistema Valida su contraseña. 3. El Sistema despliega una ventana con las opciones para elaborar reportes 5. El Sistema genera reportes por cada cuenta.
<b>Flujo Alternativo.</b>	Paso 5. si no existe ningún reporte en la Tabla reportes el sistema no muestra ningún reporte y el caso de uso termina	
<b>Precondición</b>	El OC debe elaborar al menos un reporte perteneciente a una cuenta.	
<b>Poscondición</b>	El Adm. Debe obtener el reporte para ser enviado al o los clientes o encargados de cada cuenta.	

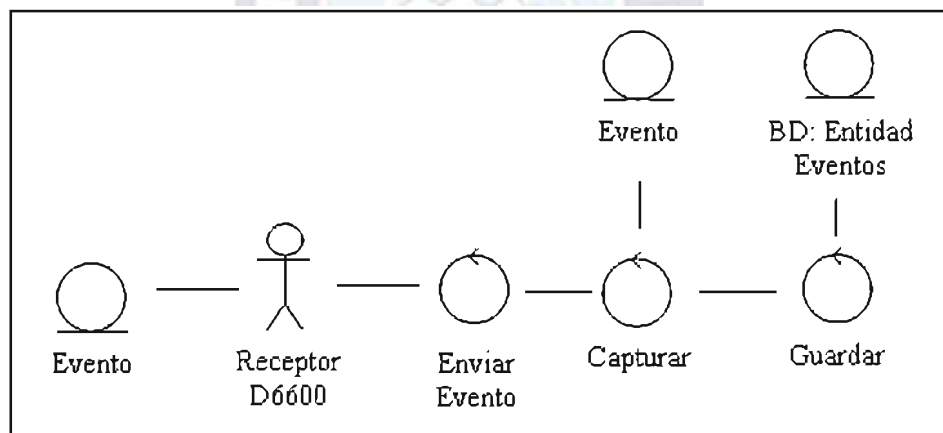
### 3.3. MODELO DE ANALISIS

En esta sección refinaremos los requisitos y se indaga en aspectos internos del sistema. Para tal efecto se utiliza los diagramas casos de uso – análisis y los diagramas de colaboración

#### 3.3.1. Casos de Uso Análisis

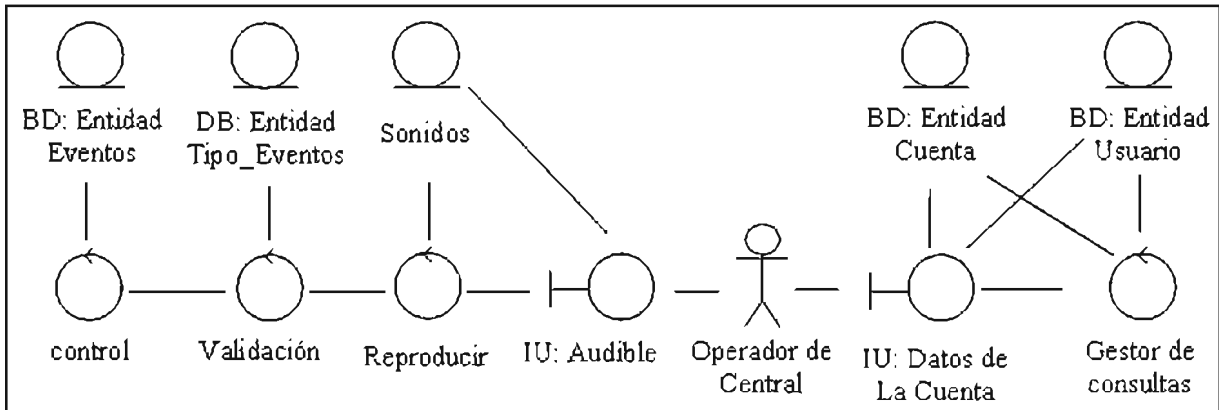
A continuación, se refleja la abstracción del sistema por medio de los casos de uso -análisis, donde se ve una coordinación de los requisitos. Para ello, se muestran los actores participantes y las relaciones del modelo de casos de uso de la sección 3.2.

En la Fig. 3.9 mostramos como el R. D6600 recibe, lo envía al Sistema de Monitoreo que se encarga de capturar y almacenar el evento en la base de datos del sistema.



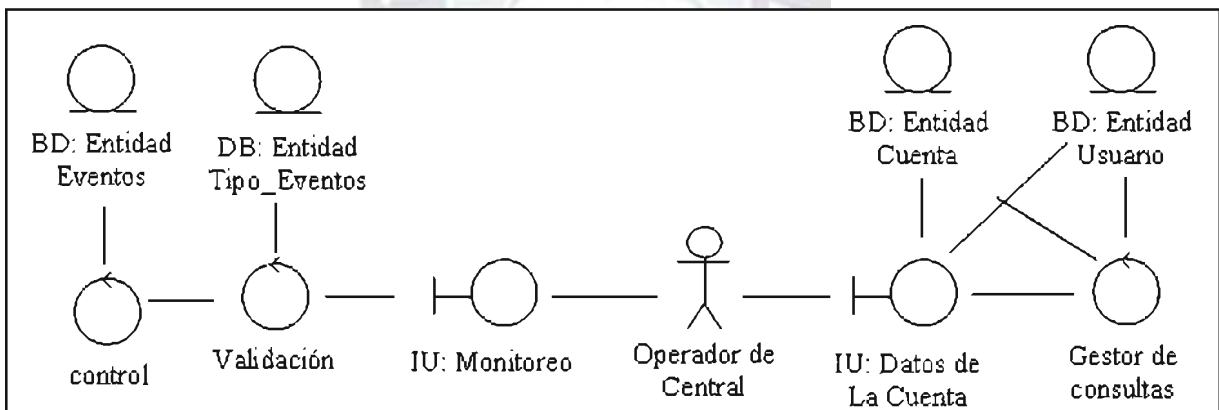
**Fig. 3.9. Diagrama de casos de uso -análisis: Percibir y almacenar eventos**  
**Fuente: Elaboración propia**

En la Fig. 3.10 se describe como el Sistema de Monitoreo genera una alerta sonora que es percibida por el Operador de Central y, a continuación el Operador por medio de la interfaz de usuario datos de la cuenta, realiza una consulta en la base de datos para realizar su procedimiento de respuesta.



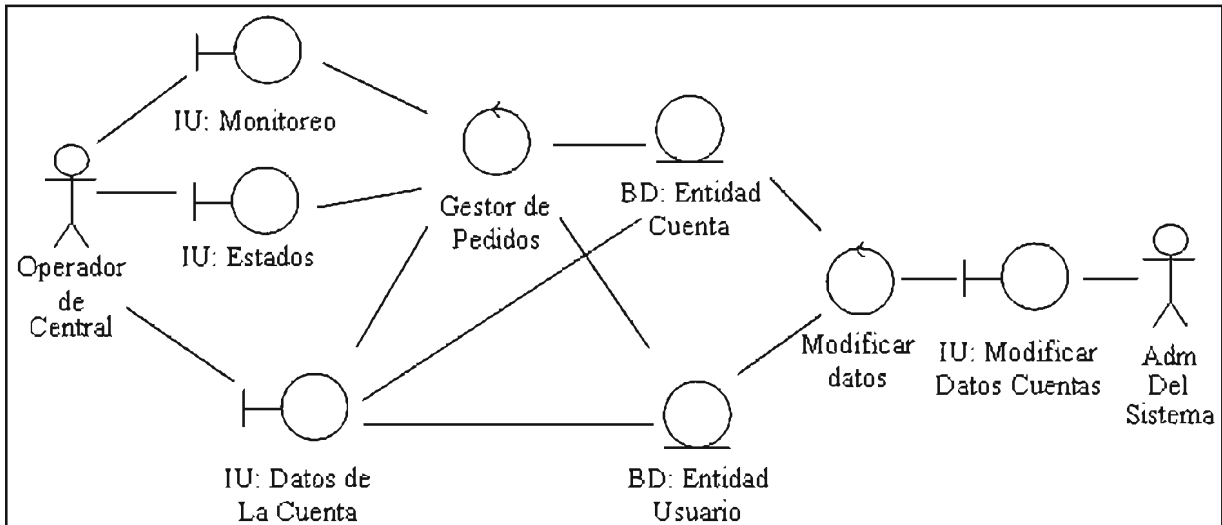
**Fig. 3.10. Diagrama de casos de uso-análisis: Percepción de la alerta sonora por el Operador de Central .**  
**Fuente: Elaboración propia**

En la Fig. 3.11 Se describe como el Sistema de Monitoreo despliega el evento en pantalla, que es percibido por el OC y, a continuación como en el caso anterior, Operador por medio de la interfaz de usuario datos de la cuenta, realiza una consulta en la base de datos para realizar su procedimiento de respuesta.



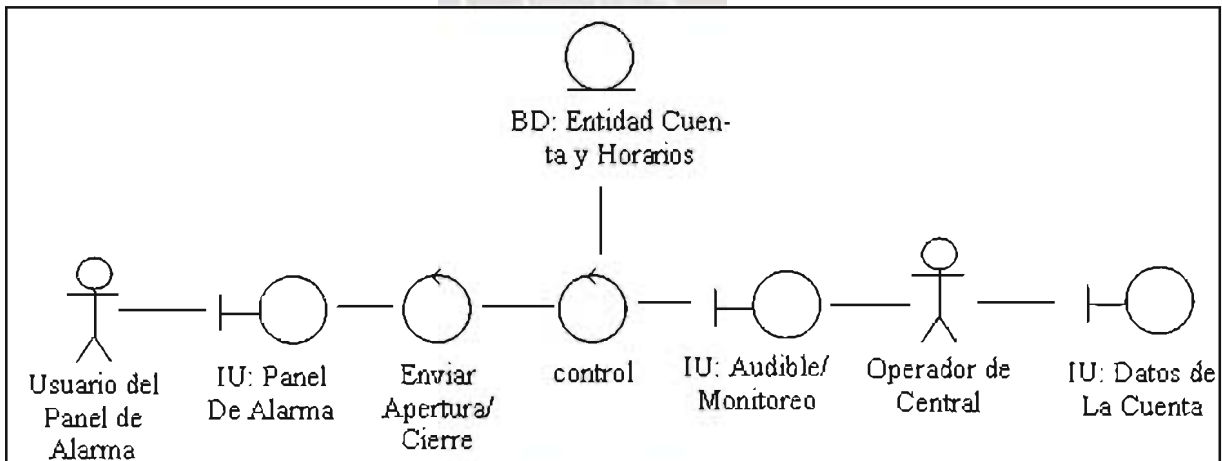
**Fig. 3.11. Diagrama de casos de uso-análisis: Percepción visual del evento por el Operador de Central**  
**Fuente: Elaboración propia**

En la Fig. 3.12 Se describe como el Operador de Central consultará información de la Base de Datos del Sistema que previamente fue Adicionada o modificada por el Administrador del Sistema.



**Fig. 3.12. Diagrama de casos de uso-análisis: Despliegue de la información de la cuenta**  
**Fuente: Elaboración propia**

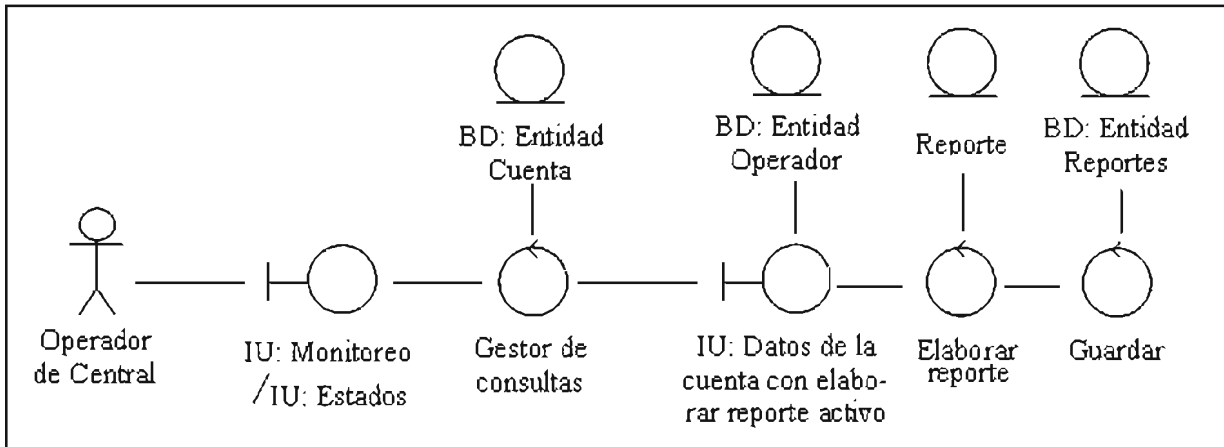
En la Fig. 3.13 Se describe el control de Aperturas/Cierres de cada cuenta que tiene este servicio. El Usuario del Panel de Alarma realiza un armado/desarmado de su panel de alarma. El Sistema de Monitoreo realiza el control de horarios para verificar que este evento este en el rango establecido para cada cuenta. Luego el Operador de Central percibe el evento de forma visual o audible.



**Fig. 3.13. Diagrama de casos de uso-análisis: Controlar horarios de apertura y cierre**  
**Fuente: Elaboración propia**

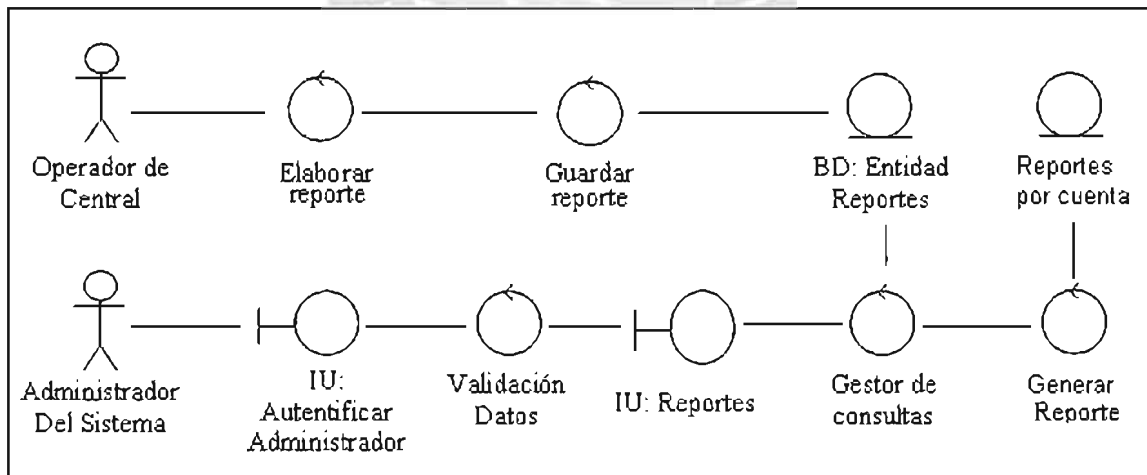
En la Fig. 3.14 Se describe la elaboración de un reporte por el Operador de Central quien luego de realizar un procedimiento de respuesta debe elaborar un reporte y luego el Sistema de Monitoreo almacena el reporte.





**Fig. 3.14. Diagrama de casos de uso -análisis: Elaboración de reportes**  
Fuente: Elaboración propia

En la Fig. 3.15 Se refleja el proceso elaboración de reportes. El Operador de Central como lo vimos en la Fig. 3.14 es el encargado que elabora los reportes, sin embargo no basta con elaborar los reportes también debemos generar un reporte de los reportes que se elaboraron por cada cuenta existente en la base de datos del sistema, y esta labor la realiza el administrador del sistema para luego poder enviar dicho reporte a cada cliente.

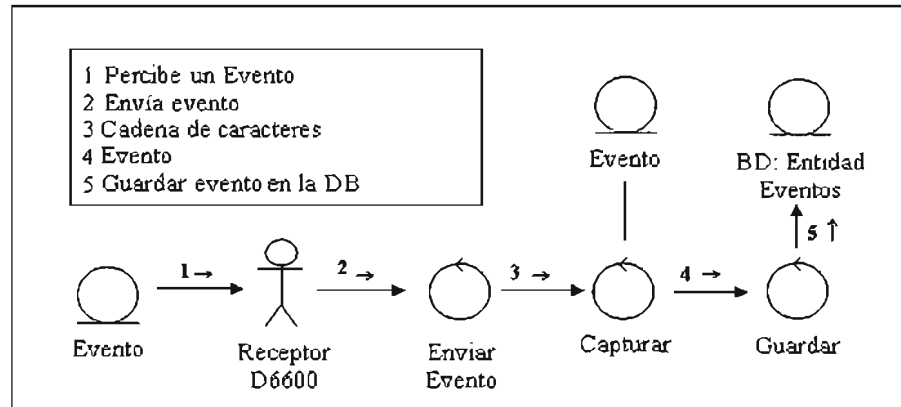


**Fig. 3.15. Diagrama de casos de uso -análisis: Generar Reportes**  
Fuente: Elaboración propia

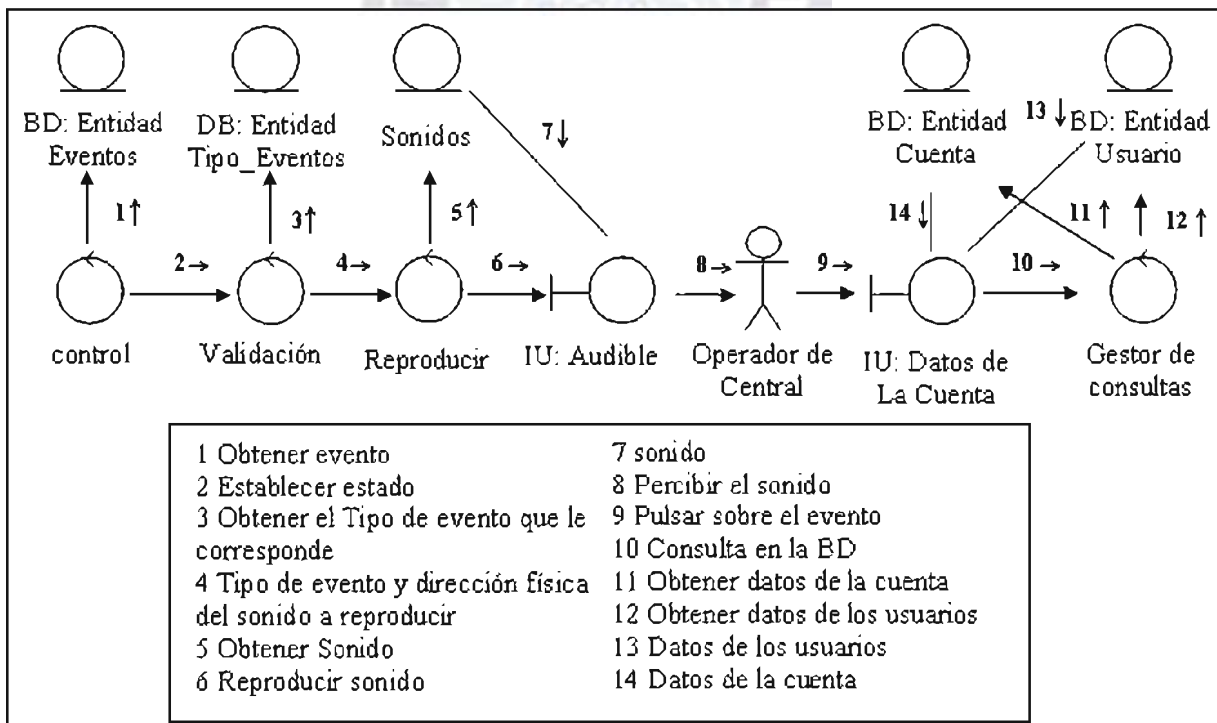
### 3.3.2. Diagramas de Colaboración

En esta sección se presentan diagramas identificando los requisitos y responsabilidades sobre los objetos, creando enlaces entre ellos y añadiendo mensajes a los enlaces, donde, el nombre de un mensaje denota el propósito del objeto invocante en la interacción con el objeto invocado.

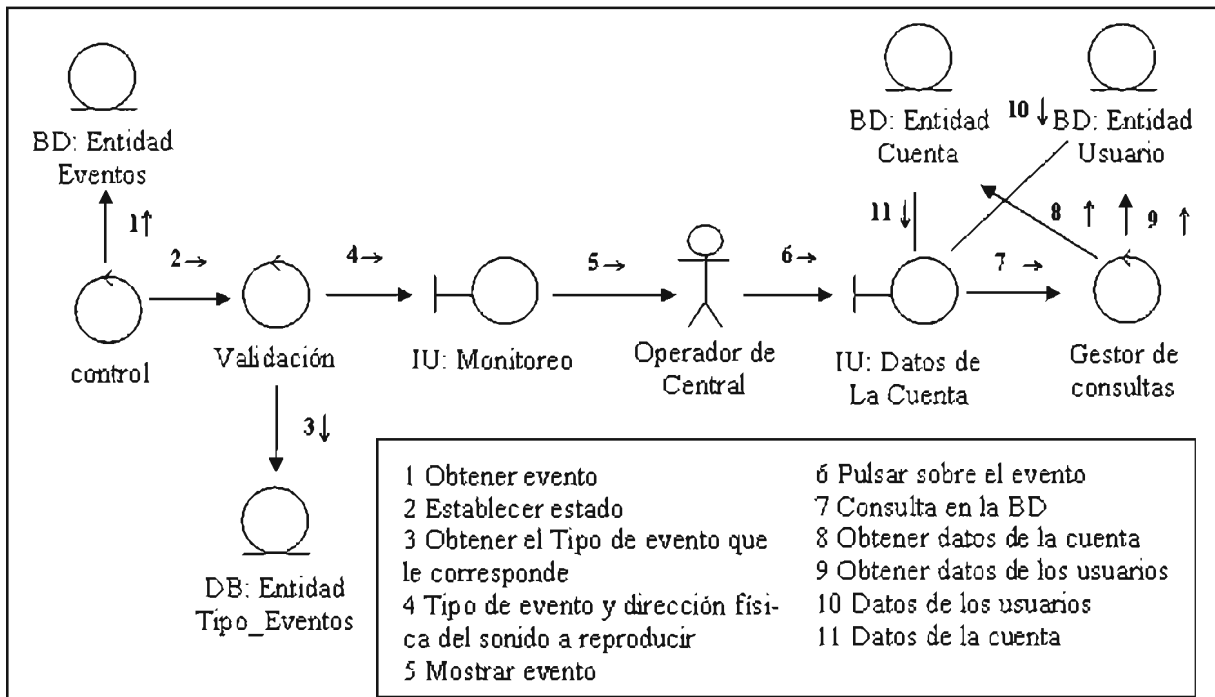
A continuación se muestran las figuras: Fig. 3.16, Fig. 3.17, Fig. 3.18, Fig. 3.19, Fig. 3.20, Fig. 3.21 y Fig. 3.22, que representan los diagramas de colaboración elaborados en función a los diagramas de casos de uso-análisis descritos en la sección 3.3.1.



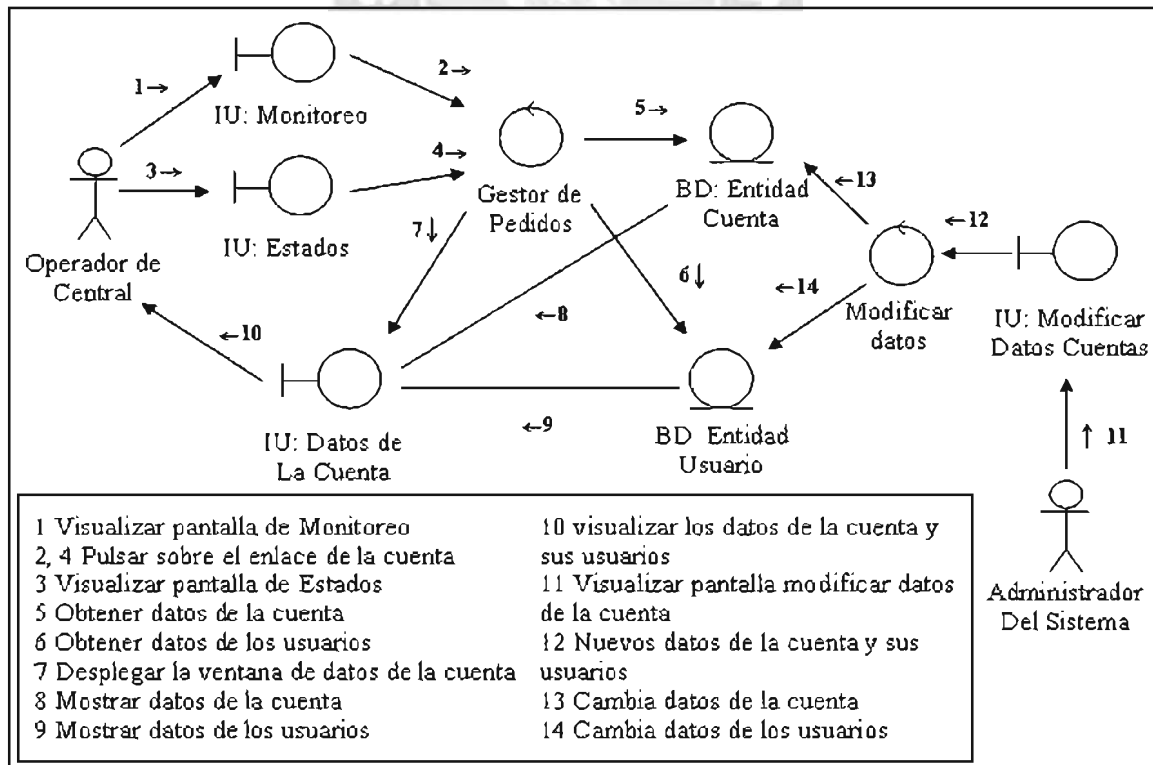
**Fig. 3.16. Diagrama Colaboración: Percibir y almacenar eventos**  
Fuente: Elaboración propia



**Fig. 3.17. Diagrama Colaboración: Percepción de la alerta sonora por el Operador de Central .**  
Fuente: Elaboración propia



**Fig. 3.18. Diagrama Colaboración: Percepción visual del evento por el Operador de Central**  
Fuente: Elaboración propia



**Fig. 3.19. Diagrama Colaboración: Despliegue de la información de la cuenta**  
Fuente: Elaboración propia

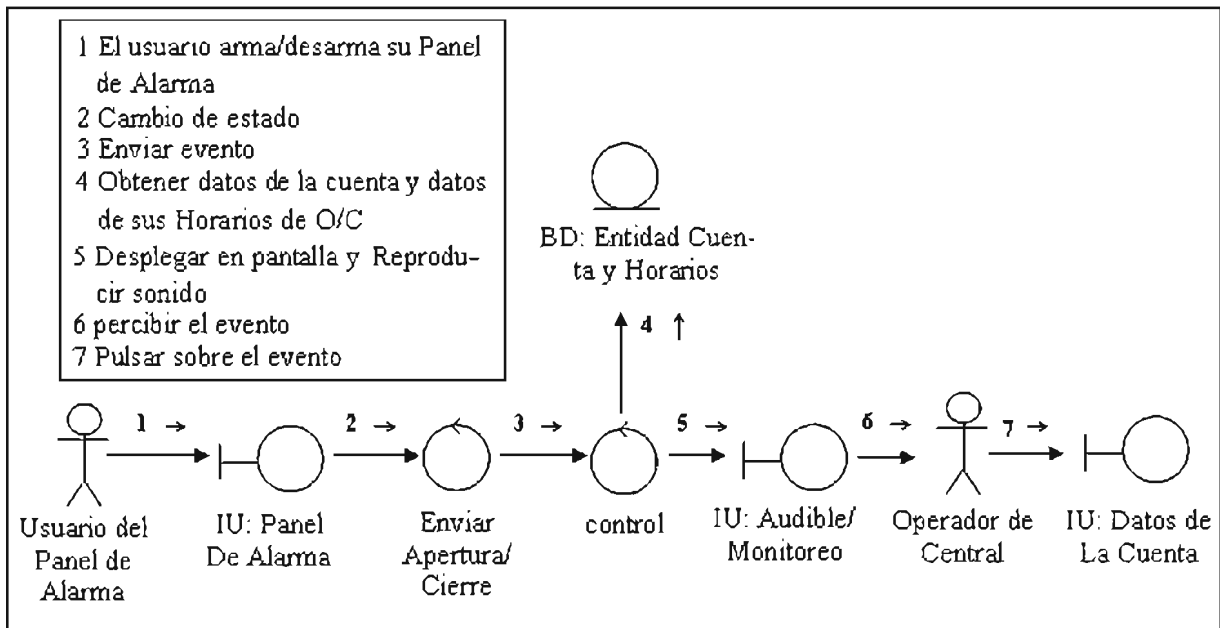


Fig. 3.20. Diagrama Colaboración: Controlar horarios de apertura y cierre  
 Fuente: Elaboración propia

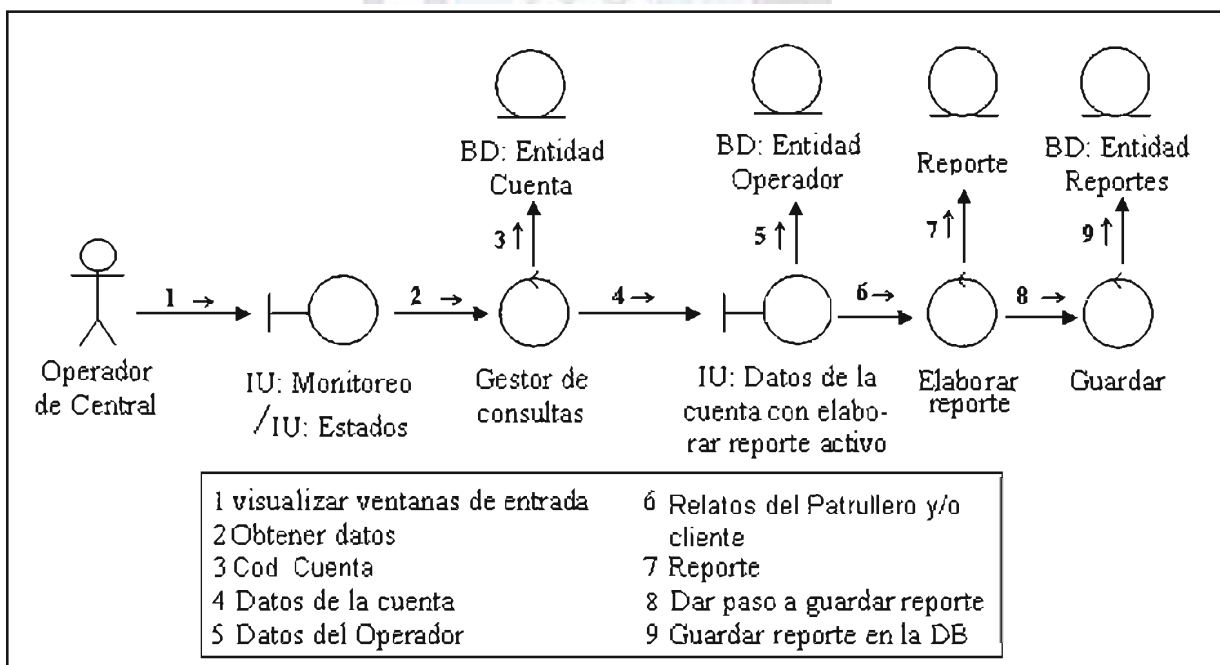
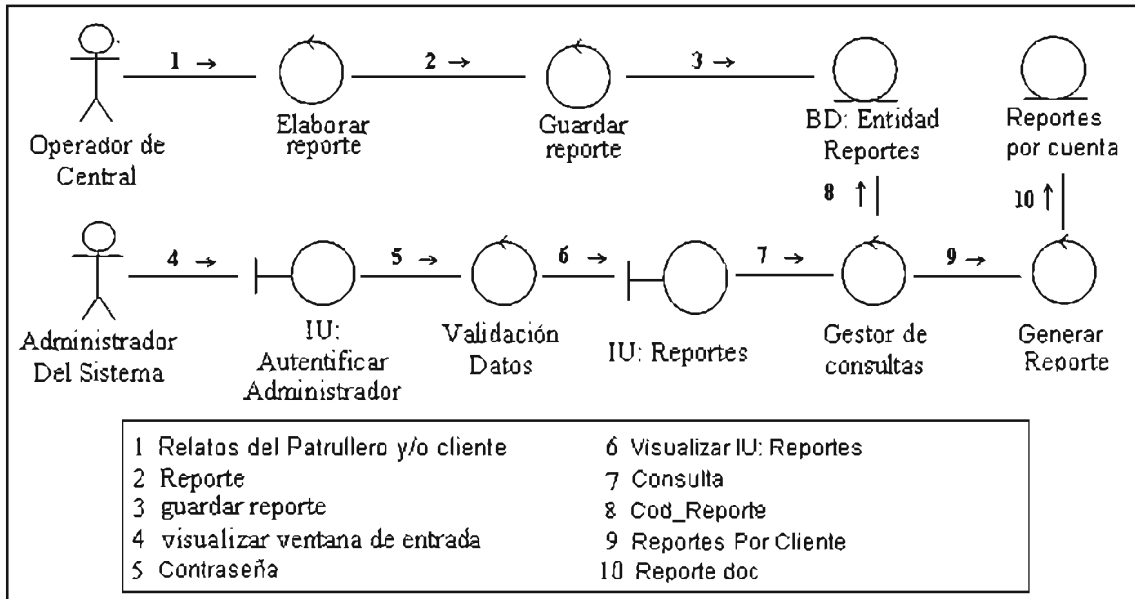


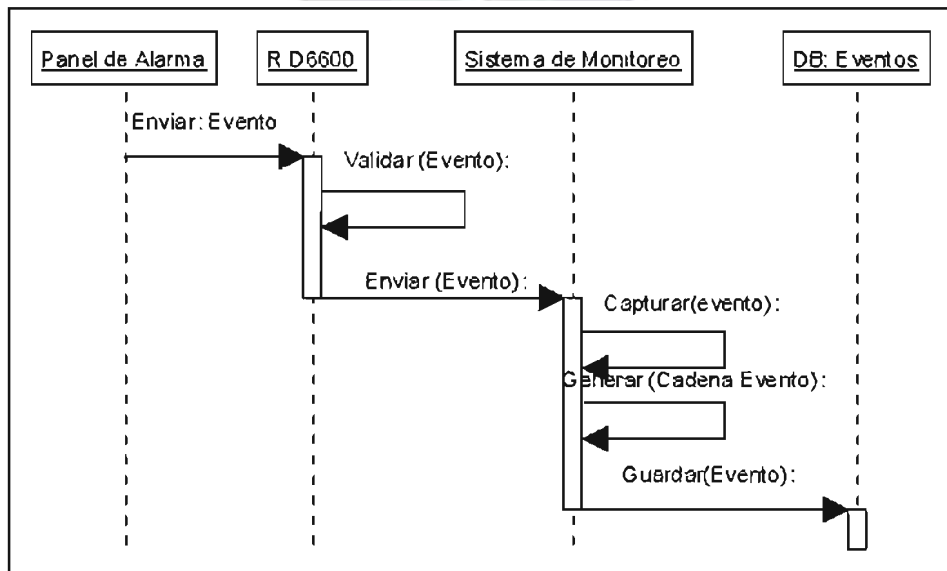
Fig. 3.21. Diagrama Colaboración: Elaboración de reportes  
 Fuente: Elaboración propia



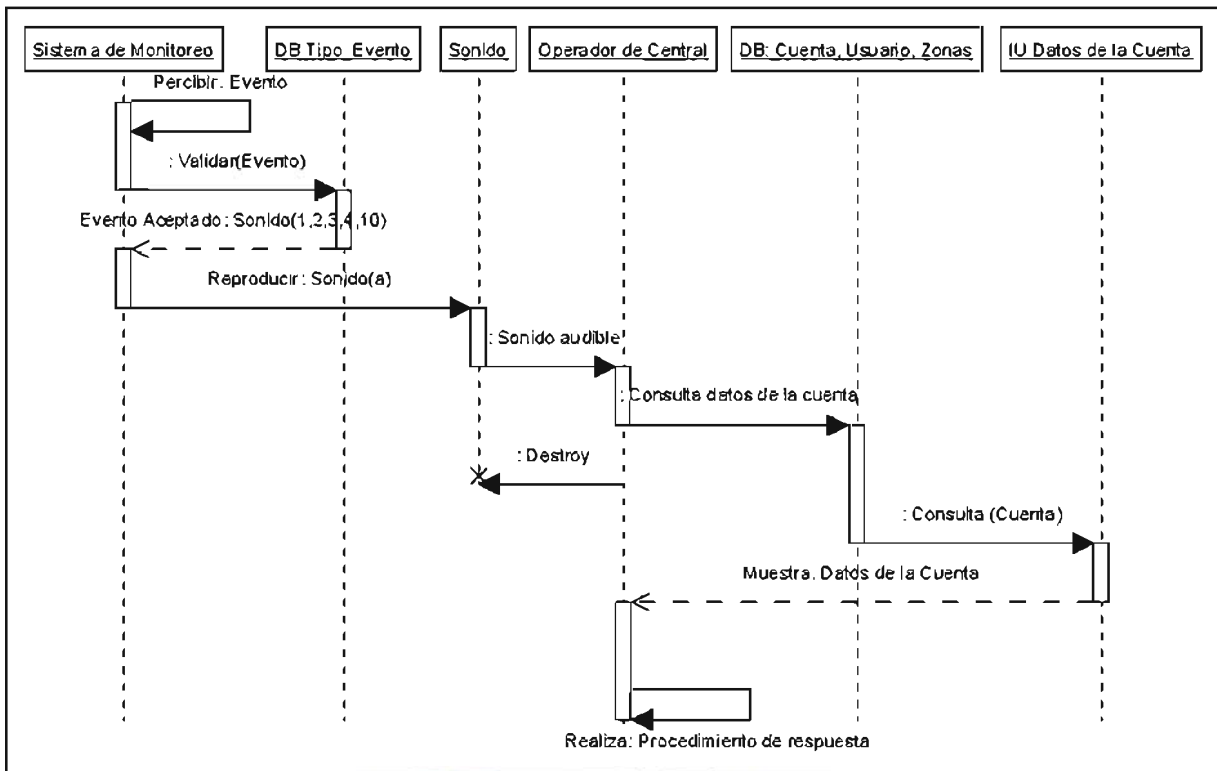
**Fig. 3.22. Diagrama Colaboración: Generar reportes**  
Fuente: Elaboración propia

### 3.3.3. Diagrama de Secuencias

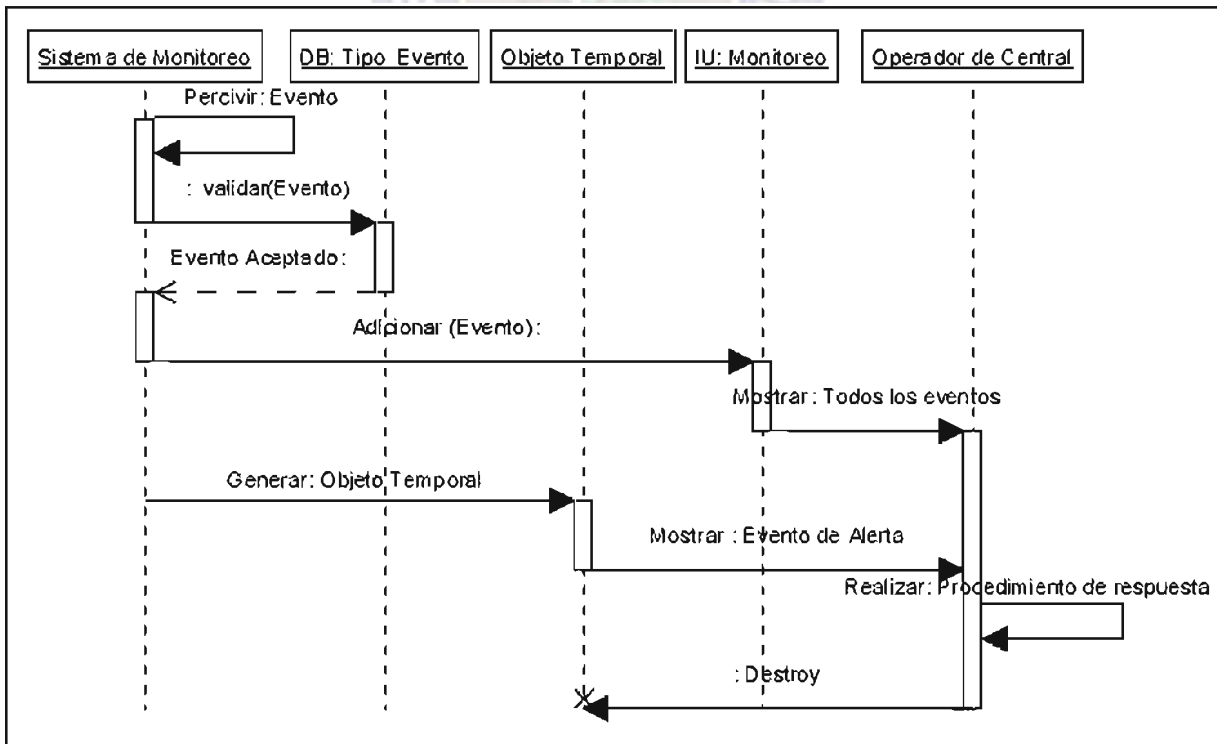
Los diagramas de secuencia que se muestran a continuación son para modelar los aspectos dinámicos del sistema. Un diagrama de secuencia muestra una interacción, que consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos, además se destaca la ordenación temporal de los mensajes.



**Fig. 3.23. Diagrama Secuencia: Percibir y almacenar eventos**  
Fuente: Elaboración propia



**Fig. 3.24. Diagrama Secuencia: Percepción de la alerta sonora por el Operador de Central.**  
**Fuente: Elaboración propia**



**Fig. 3.25. Diagrama Secuencia: Percepción visual del evento por el Operador de Central**  
**Fuente: Elaboración propia**



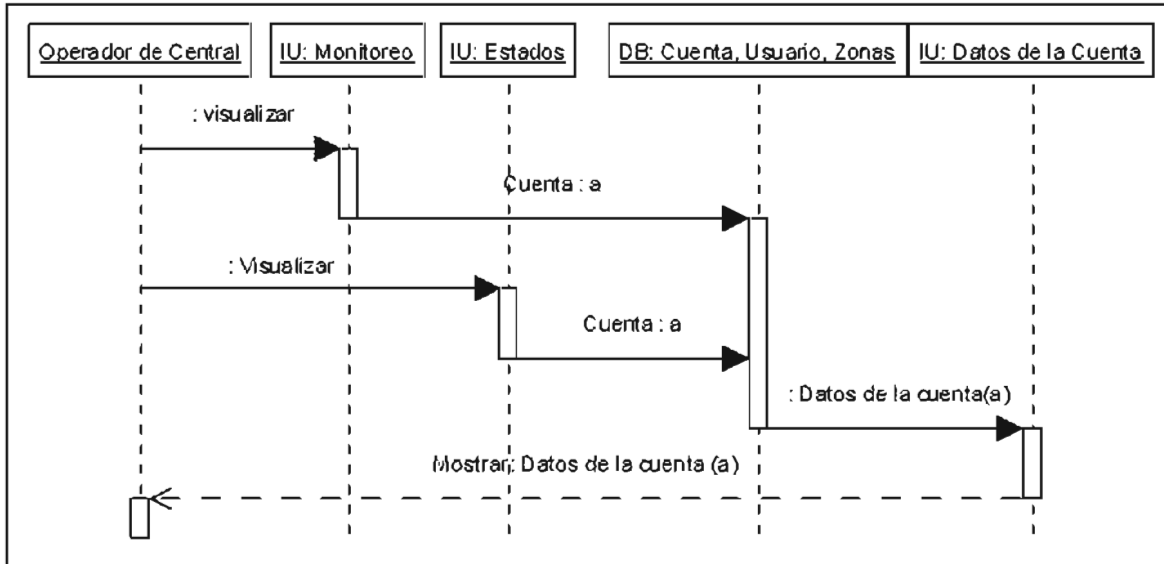


Fig. 3.26. Diagrama Secuencia: Despliegue de la información de la cuenta  
Fuente: Elaboración propia

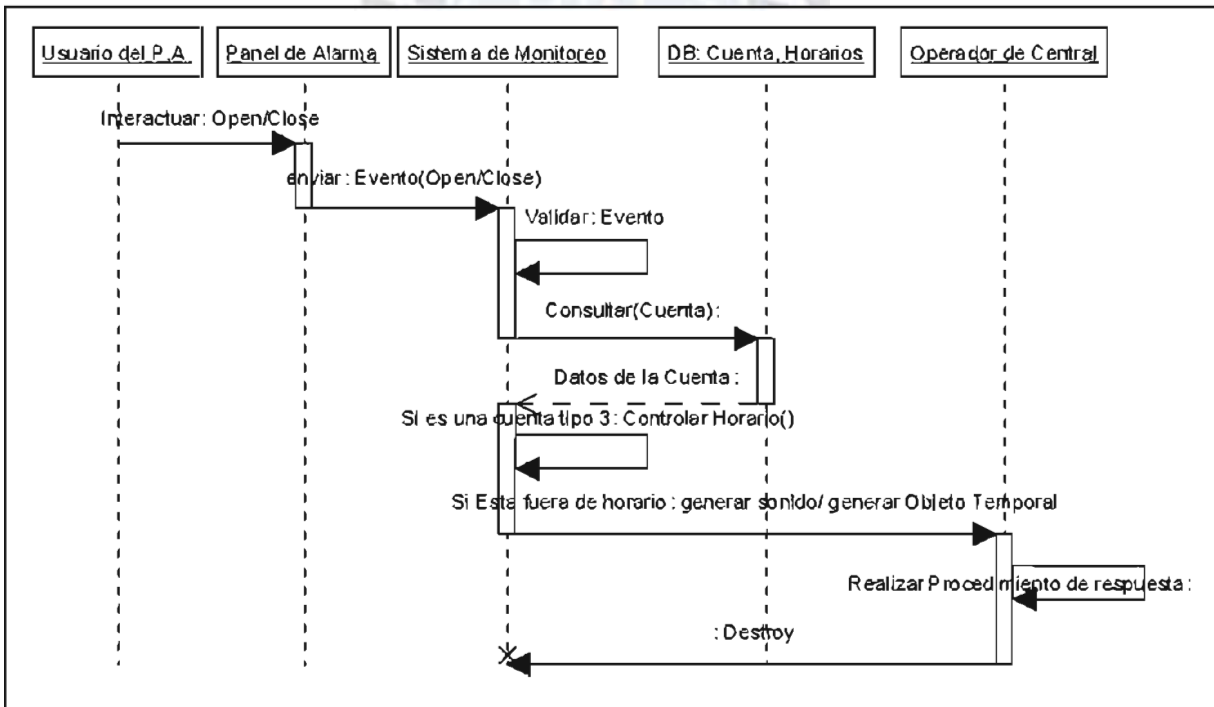


Fig. 3.27. Diagrama Secuencia: Controlar horarios de apertura y cierre  
Fuente: Elaboración propia

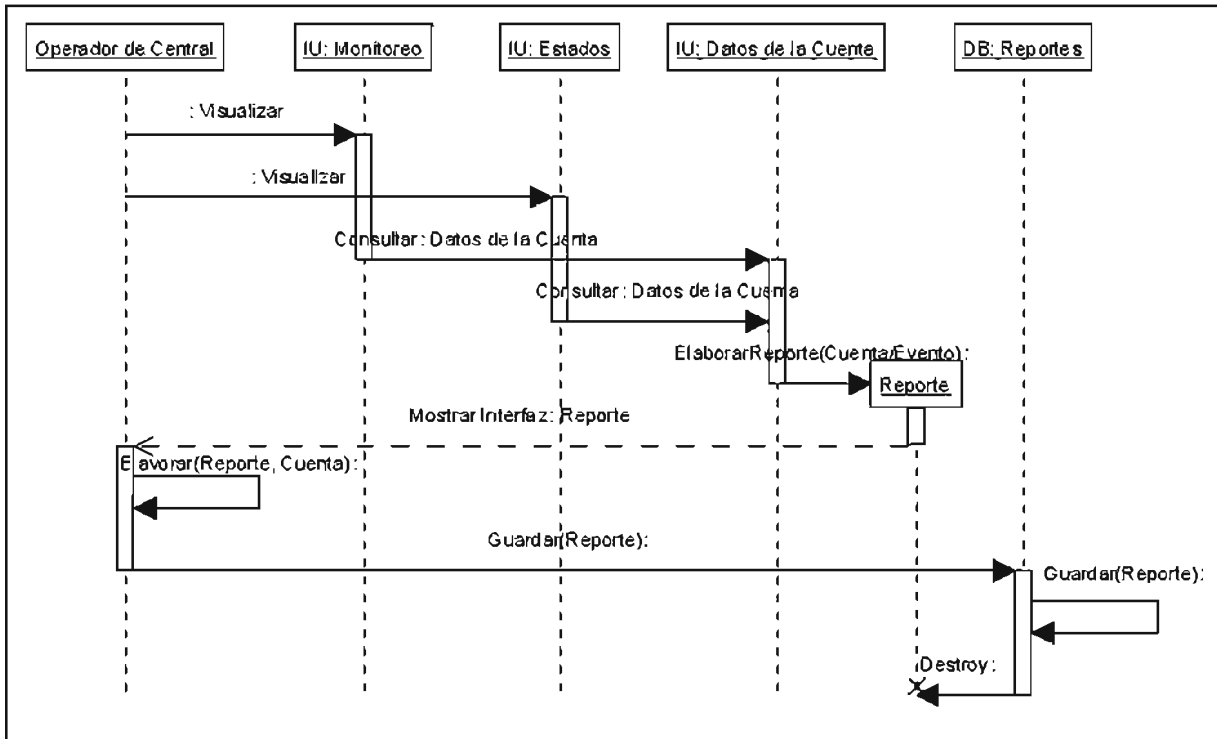


Fig. 3.28. Diagrama Secuencia: Elaboración de reportes  
Fuente: Elaboración propia

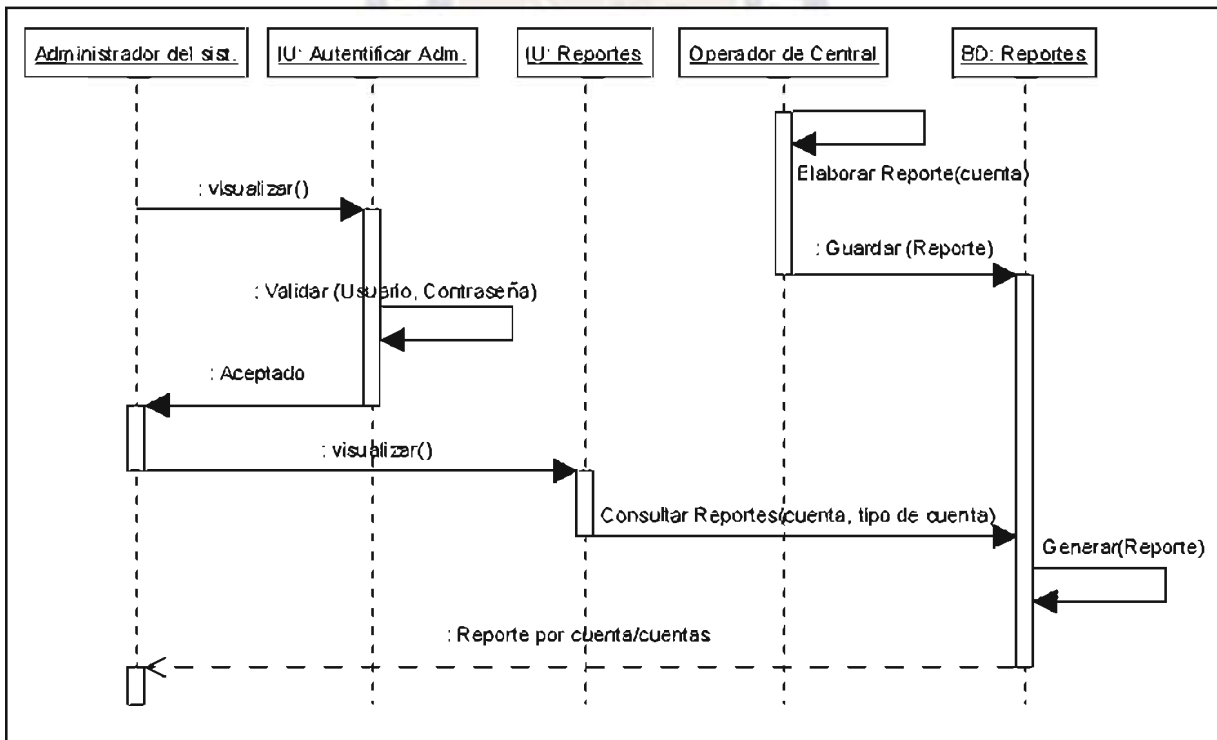


Fig. 3.29. Diagrama Secuencia: Generar Reportes  
Fuente: Elaboración propia

### 3.3.4. Diseño del sistema en Tiempo Real

Primeramente identificaremos el tipo de sistema de tiempo real al que pertenece el Sistema de Monitoreo WOLF de acuerdo con las especificaciones en el Capítulo II sección 2.6.5 es un sistema acrítico (soft real-time system) por que tiene un plazo de respuesta flexible, comportamiento temporal determinado por el computador, cuenta con recuperación de fallos y maneja un gran volumen de datos.

En cuanto al tiempo de respuesta, es un Best-effort System por que su comportamiento temporal es del tipo “lo mejor que se pueda” y además no hace una caracterización específica de carga. También podemos indicar que es un sistema dirigido por sucesos (Event-Triggered System) ya que el módulo de recepción de eventos se acciona cuando se produce un suceso de cambio de estado en el puerto serial al recibir una trama procedente del Receptor D6600. El módulo de control de horarios trabaja bajo un mecanismo de reloj, por tanto el sistema también es dirigido por tiempo (time-triggered system). También bebemos indicar que el sistema se instalará sobre un computador convencional y el control será centralizado.

A continuación identificaremos las restricciones de tiempo que deben ser satisfechas por el sistema, para ello debemos hallar el tiempo máximo en que el Receptor D6600 envía una trama. Para dicho cálculo tomaremos las siguientes tramas de ejemplo:

- a) [10]B4491Ca52290101[#0105|18340100001][13]      [10]EE840B[9]52290101[[13]
- b) [10]6FA410[9]51710101[NYB12][13]      [10]E1D90B[9]51710101[[13]
- c) [10]A1D80B[9]51800100[[13]      [10]A1D80B[9]51800100[[13]

Debemos recalcar que en los primeros dos incisos las tramas llegan en periodos irregulares de modo que su ocurrencia es impredecible. La última trama sin embargo se presenta en periodos regulares de 30 segundos.

Las dos tramas del inciso a) representan a una trama valida de un evento enviado por un panel de alarma. De la primera trama se extraen los datos que el sistema necesita para poder realizar su trabajo de monitoreo. La segunda es una trama de verificación, ambas tramas llegan siempre juntas, y en ese orden (Anexo A3).

Las dos tramas del inciso b) representan un evento propio del Receptor D6600 y las tramas del inciso c) son tramas de control entre el Software D6202 y el Receptor D6600, las cuales no son controladas por el sistema de monitoreo WOLF. Su tiempo de proceso es despreciable debido a que una vez percibidas solo se verifica el tipo de trama, y al no ser del tipo "a" la trama es destruida.

Ahora pasaremos a identificar las restricciones de tiempo del sistema. De acuerdo con los datos del Receptor D6600 mostrados en el software de administración D6202 tenemos que:

Baud rate speed = 38400 baudios

Data bits select = 8 bits

Stop bits select = 1

Luego con estos datos podemos calcular el tiempo que requiere cada bit para ser recibido por el sistema WOLF.

$$1 / 38400 \text{ baudios} = 2.6401e-5 \text{ bits/segundo}$$

Y tomando en cuenta las tramas del inciso a), b) y c) tenemos el número de bits por trama:

a1) 36\*8 bits

a2) 19\*8 bits

b1) 24\*8 bits

b2) 19\*8 bits

c1) 19\*8 bits

c2) 19\*8 bits

Finalmente multiplicando en tiempo que requiere un bit para ser recibido por el número de bits de cada tipo de trama, tenemos que el tiempo requerido por cada trama es:

$$a1) ( 36*8 ) \text{ bits} * 2.6401e-5 \text{ bits/segundo} = 0.0075 \text{ segundos} = 7.5000 \text{ milisegundos}$$

$$a2) ( 19*8 ) \text{ bits} * 2.6401e-5 \text{ bits/segundo} = 0.0039 \text{ segundos} = 3.9000 \text{ milisegundos}$$

$$b1) ( 24*8 ) \text{ bits} * 2.6401e-5 \text{ bits/segundo} = 0.0050 \text{ segundos} = 5.0000 \text{ milisegundos}$$

$$b2) ( 19*8 ) \text{ bits} * 2.6401e-5 \text{ bits/segundo} = 0.0039 \text{ segundos} = 3.9000 \text{ milisegundos}$$

$$c1) ( 19*8 ) \text{ bits} * 2.6401e-5 \text{ bits/segundo} = 0.0039 \text{ segundos} = 3.9000 \text{ milisegundos}$$

$$c2) ( 19*8 ) \text{ bits} * 2.6401e-5 \text{ bits/segundo} = 0.0039 \text{ segundos} = 3.9000 \text{ milisegundos}$$

De acuerdo a los resultados podemos ver que las tramas de los incisos a2), b1), b2), c1) y c2) requieren de un tiempo de recepción menor que la trama del inciso a1) y como estas tramas requieren un tiempo de proceso despreciable en el sistema no serán tomadas en cuenta más adelante.

El tiempo obtenido para el inciso a1) es de 7.5 milisegundos y es precisamente este el tiempo máximo que llamaremos **T<sub>m</sub>** que el sistema tiene como restricción, de tal manera que:

$$T_m \geq T_p$$

Donde **T<sub>p</sub>** es igual al tiempo de procesamiento del sistema para una trama del tipo “a”.

La arquitectura software del sistema se muestra con el diagrama de clases de la Fig. 3.30. El software del Sistema de Monitoreo WOLF está constituido por tres clases activas y una clase pasiva a través de la que se comunican. La clase `Serial_Port_Refresh` se activa a la llegada de un bit por el puerto serial. Su función es esperar la llegada completa de una trama del tipo “a” (Anexo A3) perteneciente a la Receptora D6600 y enviar las tramas válidas a la IU (Interfaz de Usuario) de Monitoreo para que esta se actualice; está una tarea que se presenta en intervalos irregulares que dependen del envío de eventos por parte de un Panel de Alarma. El procedimiento `Actualizar_IU_Monitoreo(String Trama)` se encarga de procesar la trama, extrae los datos especificados en el anexo A3, los almacena en un el `Archivo_Fecha_Actual.txt`, realiza algunas consultas en la Base de Datos WOLF y luego actualiza el `dataGridView` de la IU de Monitoreo y de Estados. Cada vez que una trama es recibida se genera un thread emergente.

La clase `TimerControl` es una tarea que se activa por un timer que normalmente tiene periodos regulares de 900000 ms (15 minutos). Al momento de inicializar el sistema, el periodo inicial del timer es igual a 950 ms y continúa con este periodo hasta que los minutos del reloj del sistema operativo marquen uno de los siguientes valores: 00, 15, 30,45; es en ese momento cuando el timer inicia con periodo de 900000 ms. La función del `TimerControl` cada vez que termina un periodo es monitorizar el cierre (armado) de los paneles de alarma de las cuentas tipo2 y 3, verificando su estado en la IU de Estados, Así como adicionar una nueva fila en el `dataGridViewDemoras` de la IU de Estados si se diera el caso de una demora en el armado. Para todo esto el `TimerControl` hace uso de la Base de Datos WOLF.

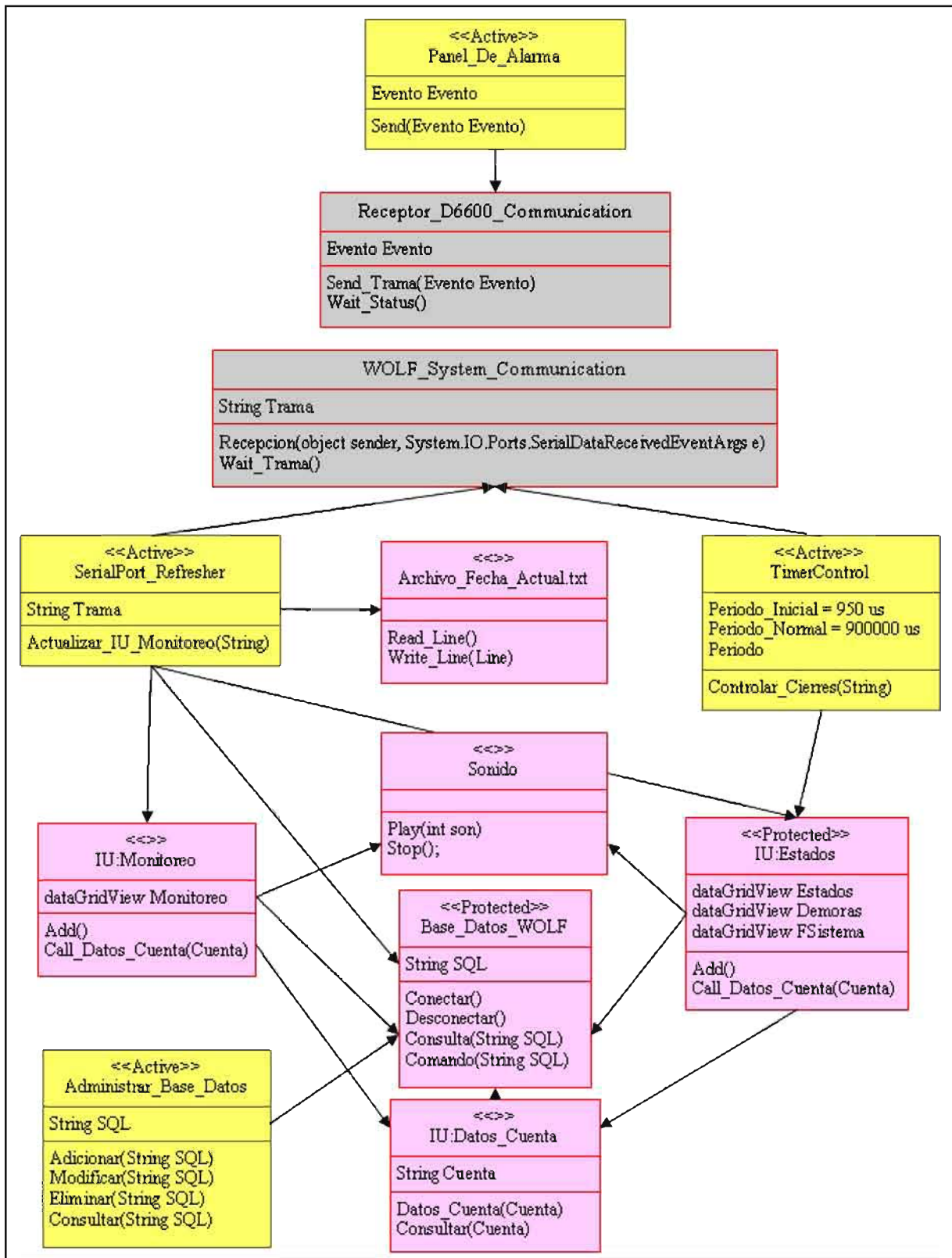


Fig. 3.30. Diagrama de clases y arquitectura del software en tiempo real

Fuente: Elaboración propia



La IU Datos\_Cuenta es la interfaz que muestra al operador de la central la información de la cuenta, para ello esta IU también hace uso de la Base de Datos. Lo que convierte a la Base de datos WOLF en un objeto protegido que proporciona acceso seguro a los datos de las cuentas a los dos procesos que acceden a ella.

La clase Administrar\_Base\_Datos es un proceso separado diferente al del Sistema de Monitoreo WOLF. Es una tarea que se presenta en intervalos irregulares que dependen del trabajo que realice el administrador del sistema (adicionar, modificar (actualizar ), eliminar y/o consultar la Base de Datos WOLF).

### **3.3.5. Diagrama de Clases**

Los diagramas de clases se utilizan para modelar la vista de diseño estática de un sistema. Esta vista soporta principalmente los requisitos funcionales de un sistema, servicios que el sistema debe proporcionar a sus usuarios finales [RUP,2000]. Además el diagrama de clases es la base para modelar el esquema lógico de la base de datos. En la Fig. 3.31 se muestra el diagrama de clases del sistema.

### **3.3.6. Modelado de Base de Datos**

En la Fig. 3.32 se detalla el modelado de la Base de Datos del Sistema De Control De Aperturas-Cierres Y Efectos Sonoros Para El Monitoreo De Alarmas Electrónicas , este diagrama muestra todas las tablas con sus respectivos atributos y relaciones.

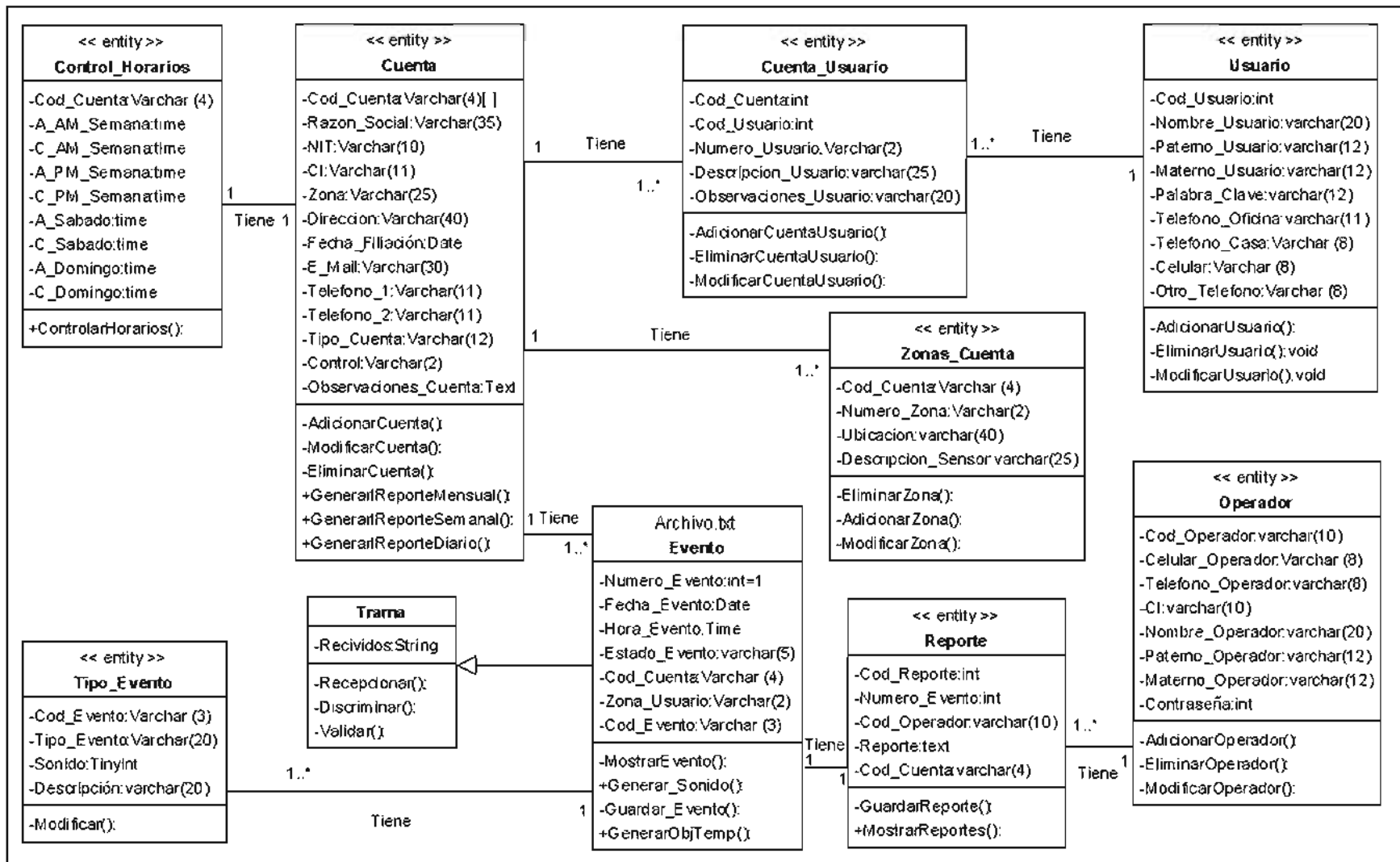


Fig. 3.31 Diagrama de Clases  
Fuente: Elaboración propia

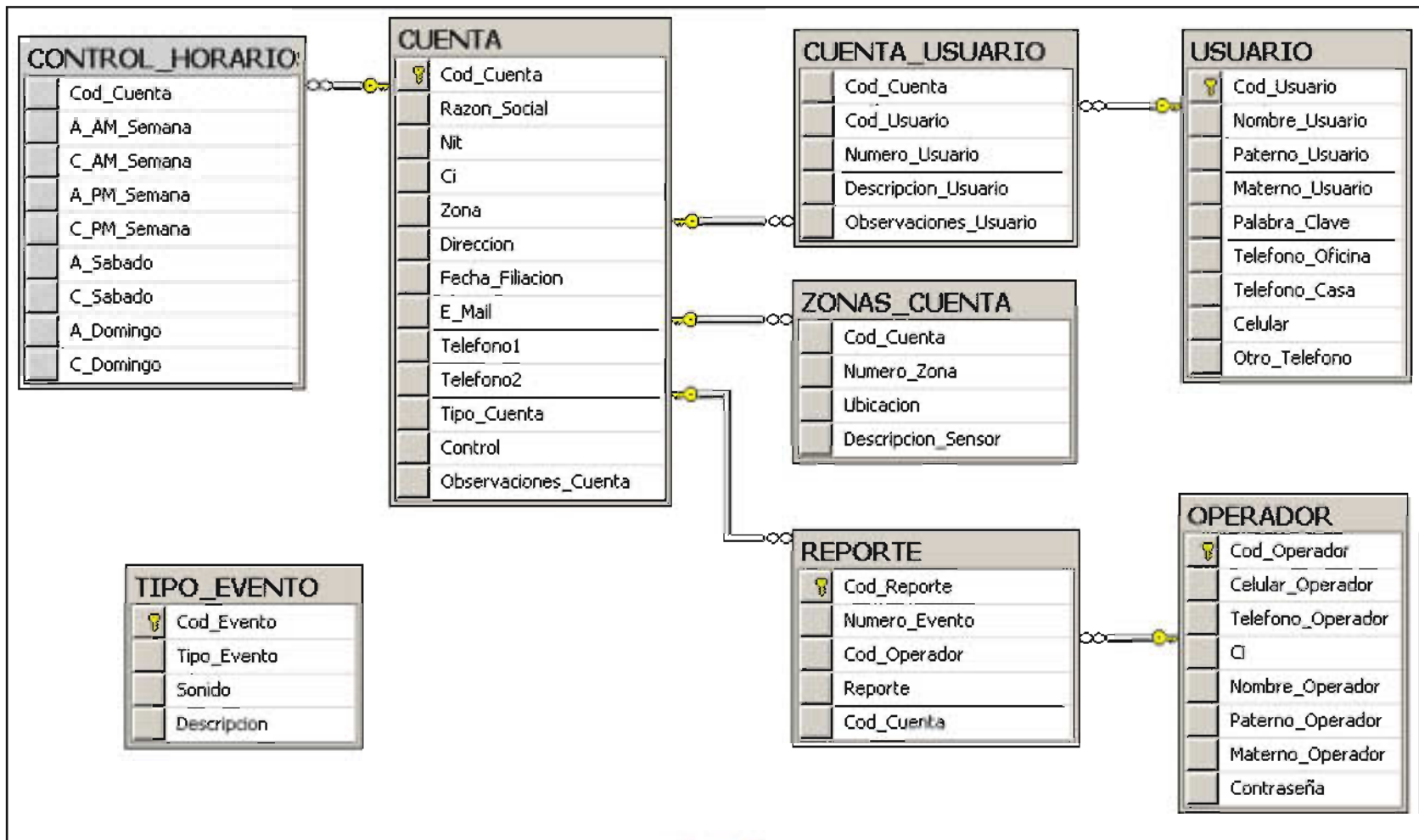


Fig. 3.32 Modelo de Bases de Datos  
Fuente: Elaboración propia

<b>ENTIDAD: cuenta</b>			
<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Comentarios</b>
Cod_Cuenta	Varchar(4)	No	Número de cuenta único e inequívoco
Razon_Social	Varchar(35)	Sí	Razón Social con la que la empresa fue inscrita como sociedad
NIT	Varchar(10)	Sí	Número de registro único NIT
CI	Varchar(11)	Sí	Número de cedula de identidad del usuario o cliente representante del inmueble
Zona	Varchar(25)	No	Zona donde esta ubicado el inmueble
Direccion	Varchar(40)	No	Avenida o calle sobre la cual esta ubicado el inmueble
Fecha_Filiación	Date	No	Fecha de filiación
E_Mail	varchar(30)	Sí	E-mail de la empresa o el cliente
Telefono_1	Varchar (11)	No	Número telefónico principal de la empresa o domicilio
Telefono_2	Varchar (11)	Sí	Número telefónico secundario de la empresa o domicilio
Tipo_Cuenta	varchar(12)	No	Tipo de cuenta detallado en el Anexo A2
Control	Varchar(2)	No	Activa la opción de control de horarios en la cuenta
Observaciones_Cuenta	Text	Sí	Espacio reservado para comentarios u observaciones referentes a la cuenta

<b>ENTIDAD: cuenta_usuario</b>			
<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Comentarios</b>
Cod_Cuenta	Varchar (4)	No	Número de cuenta único e inequívoco
Cod Usuario	Int	No	Código asignado a cada usuario
Numero Usuario	Varchar(2)	Sí	Número de usuario en la cuenta
Descripcion Usuario	varchar(25)	No	Descripción del usuario referente al cargo que ocupa en una cuenta tipo 2 y 3 o parentesco familiar en las cuentas tipo 1 (Anexo 2)
Observaciones Usuario	varchar(20)	Sí	Espacio reservado para comentarios u observaciones referentes a la cuenta

<b>ENTIDAD: usuario</b>			
<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Comentarios</b>
Cod_Usuario	Varchar (4)	No	Código asignado a cada usuario
Nombre_Usuario	varchar(20)	No	Nombre(s) del Usuario de la cuenta
Paterno_Usuario	varchar(12)	Sí	Apellido paterno del Usuario de la cuenta
Materno_Usuario	varchar(12)	Sí	Apellido materno del Usuario de la cuenta
Palabra_Clave	varchar(12)	Sí	Palabra clave asignado al usuario para ser usada como santo y seña
Telefono_Oficina	varchar(11)	Sí	Número de teléfono de la oficina del Usuario de la cuenta
Telefono_Casa	Varchar (8)	Sí	Número de teléfono de la casa del Usuario de la cuenta
Celular	Varchar (8)	Sí	Número de teléfono celular de la casa del Usuario de la cuenta
Otro_Telefono	Varchar (8)	Sí	Número de teléfono adicional del Usuario de la cuenta

<b>ENTIDAD: operador</b>			
<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Comentarios</b>
Cod_Operador	varchar(10)	No	Nombre clave asignado al Operador dentro de la empresa
Celular_Operador	Varchar (8)	No	Número de teléfono celular del Operador
Telefono_Operador	varchar(8)	Sí	Número de teléfono del domicilio del Operador
CI	varchar(10)	No	Número de carnet de identidad del Operador
Nombre_Operador	varchar(20)	No	Nombre(s) del Operador
Paterno_Operador	varchar(12)	Sí	Apellido paterno del Operador
Materno_Operador	varchar(12)	Sí	Apellido materno del Operador
Contraseña	varchar(4)	No	Contraseña del operador para registrar sus actividades en el sistema

<b>ENTIDAD: zonas cuenta</b>			
<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Comentarios</b>
Cod_Cuenta	Varchar (4)	No	Número de cuenta único e inequívoco
Numero_Zona	Varchar(2)	No	Número de Zona asignado en el Panel de Alarma
Ubicacion	varchar(40)	No	Ubicación del o los sensores pertenecientes a la zona
Descripcion_Sensor	varchar(25)	No	Descripción de los sensores instalados en la zona

<b>ENTIDAD: reporte</b>			
<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Comentarios</b>
<u>Cod_Reporte</u>	int	No	Código asignado a cada reporte
Numero_Evento	int	Sí	Número de llegada asignado al evento
Cod_Operador	varchar(10)	No	Nombre clave asignado al Operador dentro de la empresa
Reporte	text	No	Descripción del procedimiento realizado de acuerdo al evento acontecido
Cod_Cuenta	varchar(4)	No	Número de cuenta único e inequívoco

<b>ENTIDAD: tipo_evento</b>			
<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Comentarios</b>
Cod_Evento	Varchar (3)	No	Código del evento descrito en el Anexo A1
Tipo_Evento	Varchar(20)	No	Representa el tipo de evento al que pertenece
Sonido	Tinyint	No	Numero de identificación del sonido seleccionado para el evento
Descripción	varchar(20)	No	Descripción del evento (Anexo A1)

<b>ENTIDAD: control_horarios</b>			
<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>	<b>Comentarios</b>
<u>Cod_Cuenta</u>	Varchar (4)	No	Número de cuenta único e inequívoco
A_AM_Semana	time	No	Horario de apertura por la mañana (Lunes-Viernes)
C_AM_Semana	time	Sí	Horario de cierre medio día (Lunes-Viernes)
A_PM_Semana	time	Sí	Horario de apertura por la tarde (Lunes-Viernes)
C_PM_Semana	time	No	Horario de cierre por la tarde (Lunes-Viernes)
A_Sabado	time	Sí	Horario de apertura por la tarde (Sábado)
C_Sabado	time	Sí	Horario de cierre por la tarde (Sábado)
A_Domingo	time	Sí	Horario de apertura por la tarde (Domingo)
C_Domingo	time	Sí	Horario de cierre por la tarde (Domingo)



### 3.3.7. ARQUITECTURA DEL SISTEMA

#### 3.3.7.1. Entorno del Sistema

El entorno de un sistema de tiempo real como se muestra en la Fig. 2.3 del apartado 2.6.1 nos muestra tres actores principales (Operador, Sistema de tiempo real y Sistema controlado) y dos interfaces (Interfaz de operador e Interfaz de proceso), es de acuerdo a estas especificaciones que se muestra el Entorno del Sistema de monitoreo WOLF en la Fig. 3.33.

La Interfaz entre el Receptor D6600 se muestra en la Fig. 3.34 donde se detalla una comunicación de puerto serial de acuerdo a las especificaciones del cable Null Modem descrito en el capítulo 2 apartado 2.2.2.1.

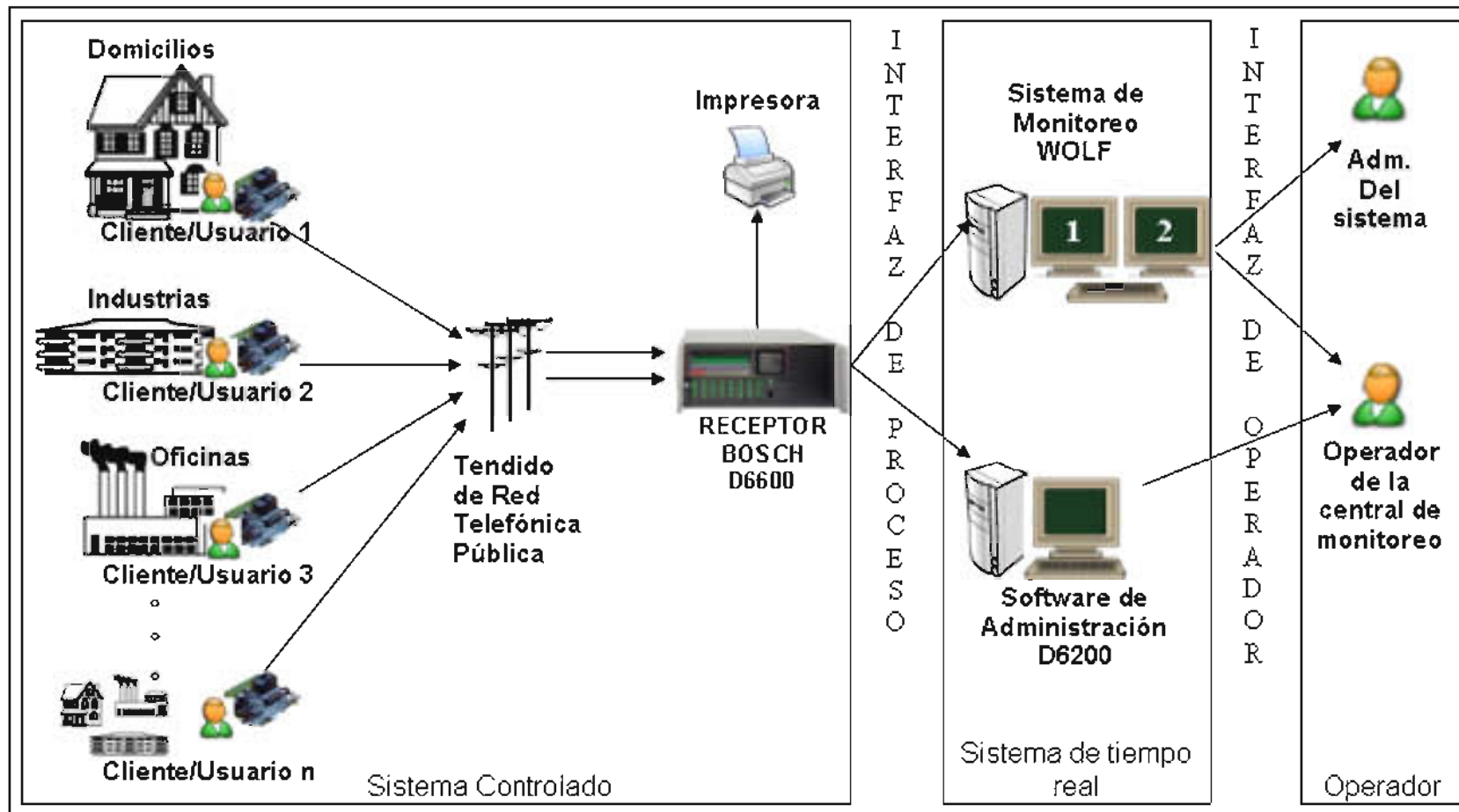
#### 3.3.7.2. Requerimientos de Hardware

Para la implementación del Sistema de Monitoreo WOLF se requiere como mínimo los siguientes componentes de hardware para su respectiva instalación, ver Tabla 3.1.

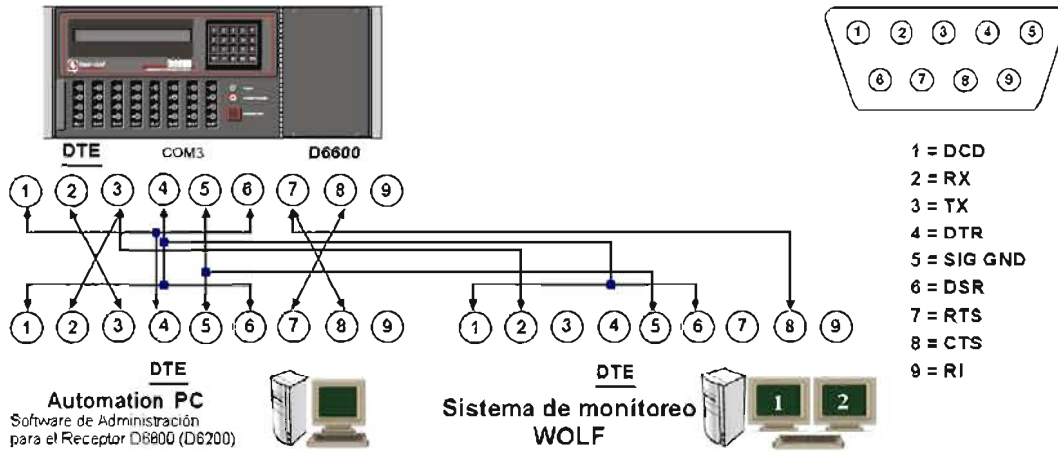
Requerimiento de Hardware	
Procesador	Mínimo 60 MHz, recomendado 1 GHz
Memoria	Mínimo 192 Mb, recomendado 256 Mb
Espacio disponible en disco duro	5.3 Gb
Unidad de disco	Mínimo de 4 Gb o superior
Tarjeta de video	1024 x 768 High Color — 16-bit
Monitores	2 Monitores de 17" Wide Scream
Puerto serial	Un puerto serial DB9
Cables de conexión	Null MODEM Cable female-2female Fig. 3.34.

Tabla 3.1. Requerimiento de Hardware

Fuente: Elaboración propia



**Fig. 3.33. Entorno del Sistema de Monitoreo WOLF**  
 Fuente: Elaboración propia



**Fig. 3.34. Descripción física de la comunicación por el puerto serial**  
**Fuente: Elaboración propia**

### 3.3.7.3. Requerimientos de Software

El sistema requiere como mínimo los siguientes componentes de software para su respectiva instalación, ver Tabla 3.2.

Requerimiento de Software	
Sistema operativo	Microsoft® Windows® XP Professional SP2 o Microsoft® Windows® XP Home Edition SP2
Lenguaje de programación	C Sharp Express Edition (C#)
Gestor de Base de Datos	SQL Server Express Edition

**Tabla 3.2. Requerimiento de Software**  
**Fuente: Elaboración propia**

## CAPITULO IV

### CONCRECION DEL MODELO

#### 4.1. IMPLEMENTACION

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 [RUP,2000]. El propósito principal de este capítulo es desarrollar la arquitectura y el sistema como un todo. Para lo cual debemos planificar las integraciones necesarias para cada iteración, distribuir el sistema asignando componentes ejecutables y nodos en el diagrama de despliegue, implementar las clases y subsistemas encontrados durante el diseño, probar los componentes individualmente, y a continuación integrarlos compilándolos y enlazándolos en uno o más ejecutables individualmente, y finalmente, integrarlos y llevar a cabo las comprobaciones del sistema.

#### 4.2. INTERFACES DE USUARIO

De acuerdo con lo mencionado en el apartado 2.6.2.3. la interfaz de un sistema de tiempo real con el operador y el administrador del sistema debe contar con ciertos aspectos típicos:

- Presentación de datos.
- Presentación de alarmas.
- Presentación de tendencias.
- Registro de datos.
- Generación de informes [Carretero ,2009].

Por lo cual dividiremos la presentación de las interfaces de usuario tomando en cuenta esas 5 características existentes. A continuación se muestran las interfaces gráficas principales de la aplicación final del sistema de monitoreo de alarmas WOL F. Es importante mencionar que por razones de seguridad que los datos que se presentan son ficticios.

## 1. Presentación de datos, presentación de alarmas y presentación de tendencias

En la presentación de datos tenemos como ventanas principales a la ventana de Monitoreo, Estados y Datos de la cuenta. Adicionalmente ventana de Monitoreo es la encargada de la presentación de alarmas que consiste en la reproducción de sonidos de alarma y el despliegue visual del evento y la presentación de tendencias.

### a) Ventana de autenticación del Operador de la Central de Monitoreo:

En la Fig. 4.1 se muestra la ventana de autenticación, donde el usuario ingresa su clave y contraseña. La clave y contraseña es asignada a cada operador de la central de monitoreo por el administrador del sistema. Si el operador de la central de monitoreo ingresa los datos de forma errónea, el sistema WOLF emite un mensaje de error.



**Fig. 4.1. Ventana de autenticación del Operador de la Central de Monitoreo**  
**Fuente: Sistema de Monitoreo WOLF**

### b) Ventana de Monitoreo:

La Fig. 4.2 muestra la ventana de monitoreo que es el centro del sistema de monitoreo, los campos mostrados dentro de la ventana de monitoreo se describen ha continuación:

- Fecha y Hora.- Representa la fecha y hora en la que el evento fue recepcionado.
- Nombre o Razón Social.- Representa el nombre o razón social del cliente.



- #Cuenta.- Representa el número de cuenta único e inequívoco.
- #Z/U.- Representa la zona en la que se produjo el evento (Robo, Bypass, etc.), en el caso de que el evento sea de interacción entre el usuario y el panel de alarma representa el número de usuario que realizó la interacción (armado, desarmado, etc.)
- Descripción Zona/Usuario.- Descripción textual del anterior campo donde podemos observar la ubicación física de los sensores o el nombre y apellido del usuario.
- Descripción.- Este campo muestra el tipo de evento (ANEXO A1).
- Evento.- Representa un adjetivo calificativo del evento (Open, Close o Previous).

En la parte baja de la ventana tenemos los eventos de alarma (Robo, Pánico del teclado, Intrusión y otros) y los eventos de sistema (Bypass, Systemperimetral, y otros) (Anexo A1).

Fecha / Hora	Tipo	Nombre o Razon Social	Cuenta	z/u	Descripción Zona / Usuario	Descripción	Evento
16/06/2009 14:33:45	DM	Fla. Arzo Mendola	9007	000	NO EXISTE	Falla de AC	CLOSE
16/06/2009 14:33:45	DM	Gubenez Sarmiento	9006	000	NO EXISTE	Falla de AC	CLOSE
16/06/2009 14:33:44	DM	Fla. Diaz Cabina	9005	000	NO EXISTE	Falla de AC	CLOSE
16/06/2009 14:33:44	DM	Familia Kankula Flores	9004	000	NO EXISTE	Armado Interno	CLOSE
16/06/2009 14:33:43	MHM	NO ESTA REGISTRADO	9015	000	NO EXISTE	Armado Interno	CLOSE
16/06/2009 14:33:43	DMT	Galeria Clavel	9018	001	PETER CHAVARRIA	Usuario	CLOSE
16/06/2009 14:33:42	DMT	Tree Eco! Dreams "BEDS"	9017	001	PETER CHAVARRIA	Usuario	CLOSE
16/06/2009 14:33:42	DMT	Tela Matiel La Paz	9016	000	NO EXISTE	TEST PERIODICO	OPEN
16/06/2009 14:33:41	DMT	Libreria Generaciones	9015	002	PETER CHAVARRIA	Usuario	CLOSE
16/06/2009 14:33:41	MHM	NO ESTA REGISTRADO	9014	000	NO EXISTE	Armado Interno	CLOSE
16/06/2009 14:33:40	DMD	Cooperativa San Martin	9013	000	NO EXISTE	Armado Interno	CLOSE
16/06/2009 14:33:40	DMD	Consultora Bolivia SRL	9012	002	Sala de Espera	ROBO	OPEN
16/06/2009 14:33:39	DMD	Interne Merian	9011	000	NO EXISTE	Armado Interno	CLOSE
16/06/2009 14:33:38	DMD	Cooperativa San Martin	9013	000	NO EXISTE	Falla de AC	CLOSE
16/06/2009 14:33:37	DM	Familia Kankula Flores	9004	000	NO EXISTE	BYPASS	OPEN
16/06/2009 14:33:36	DM	Gubenez Sarmiento	9006	000	NO EXISTE	Falla Linea Telf. 1 del Usu.	OPEN
16/06/2009 14:33:35	DM	Fla. Arzo Mendola	9007	001	Puerta Cocina	System Perimetral	OPEN
16/06/2009 14:33:34	DM	Martin Rivera	9008	000	NO EXISTE	Falla de comunicaci3n	OPEN
16/06/2009 14:33:32	DM	Martin Rivera	9008	000	NO EXISTE	TEST PERIODICO	OPEN
16/06/2009 14:33:30	DM	Martin Rivera	9008	000	NO EXISTE	TAMPER	CLOSE
16/06/2009 14:33:29	DM	Gomez de la Fuente	9003	000	NO EXISTE	TAMPER	OPEN

Fecha / Hora	Ti	Nombre o	Cui	z/u	Descripc	Descripci3n
16/06/2009 14:33:37	DM	Familia Kankula Flores	9004	000	NO EXISTE	BYPASS
16/06/2009 14:33:35	DM	Fla. Arzo Mendola	9007	001	Puerta Cocina	System Perimetral

Fecha / Hora	Tip	Nombre o	Cui	z/u	Descripci3n	Descripci3n
16/06/2009 14:33:40	DMD	Consultora Bolivia SRL	9012	002	Sala de Espera	ROBO
16/06/2009 14:33:29	DM	Gomez de la Fuente	9003	000	NO EXISTE	TAMPER
16/06/2009 14:33:28	DM	Rafel Eduardo Villamoré	9002	000	NO EXISTE	Paraz - Duet
16/06/2009 14:33:26	DM	Fla Arzo Mendola	9007	002	Cosina	ROBO

Fig. 4.2. Ventana de Monitoreo  
Fuente: Sistema de Monitoreo WOLF



### c) Ventana de Estados:

La Fig. 4.3 muestra la ventana donde se puede ver el estado actual de las cuentas, Los eventos de alarma producto del control de horarios y los eventos de alarma a causa de eventos de problemas del sistema del Panel de Alarma , donde podemos encontrar: falla de batería y corte de energía eléctrica.

Se han dividido o clasificado las cuentas de acuerdo a con los tipos de cuentas existentes y se muestran en las siguientes tres tablas: En la primera tabla podemos observar las cuentas del tipo 1 y las del tipo 2 que no tiene el servicio de control de horarios, en la segunda tabla podemos encontrar las cuentas del tipo 2 que cuentan con el servicio de control de horarios, pero que a la vez no pertenecen a un subgrupo del tipo 2 que son las cuentas monitoreadas de la financiera XXX, y finalmente, tenemos la última tabla donde podemos encontrar a todas las cuentas del tipo 2 que pertenezcan al subgrupo XXX.

The screenshot shows a window titled 'Estados' with three main sections:

- Cuentas Sin Control de Horarios:** A table with 4 columns: Cod, Razon\_Social, Column1, Column2. It lists 8 accounts with various statuses like 'Sin novedad', 'Restaurado', and 'Usuario'.
- Cuentas Con Control de Horarios:** A table with 4 columns: Cod, Razon\_Social, Column1, Column2. It lists 8 accounts, including 'Armando Interno' and 'Usuario'.
- Financiera XXX:** An empty table area.

At the bottom, there are two alarm panels:

- ALARMAS DE SYSTEM:** An empty panel.
- ALARMAS DE CONTROL DE HORARIOS:** A panel with 3 columns: Column1, Column2, Column3. It shows two entries: 'Tele Market La Paz' (OPEN) and 'Consultora Bolivia SRL' (DPEN).

**Fig. 4.3. Ventana de Estados**  
**Fuente: Sistema de Monitoreo WOLF**

### d) Ventana Datos de la Cuenta:

La ventana mostrada en la Fig. 4.4 es la que despliega toda la información necesaria para poder comunicarse con el cliente y guiar a al patrullero para que este pueda

dirigirse al lugar del evento si fuera necesario. Esta ventana se despliega cuando el operador de la central de monitoreo hace un click sobre un evento en la ventana de Monitoreo o sobre los datos de una cuenta en la ventana de Estados o Monitoreo, descritas en los anteriores dos incisos. Además también proporciona la facilidad de elaborar reportes ofreciendo las plantillas de los reportes más comunes, una vez elaborado el reporte el operador de la central de monitoreo puede guardar el evento en la base de datos para ser consultado luego y elaborar un reporte final semanal o mensual de acuerdo con las exigencias del cliente

**Datos Cuenta**

**CUENTA 9016** DMT Tele Market La Paz

**Zona** Calacoto

**Dirección** Entre calle 5 y 6 #435

**Teléfonos** 2754849 2475775

**Observaciones** En caso de Botón de Pánico llamar a Omar Cerruto

Fecha de Filiación 10/06/2008  
Control de Horarios si

**Usuarios**

#Num	Column1	Palabra_Clave	Telefono_Oficina	Telefono_Casa	Celular	Otro_Telefono	Descripcion_Usuario	Observaciones_Usuario
001	Oscar Cerruto Villegas	Licor	34243044	3783074	79663195		Gerente	llamar por Botón de Pánico
002	Clementina Cachicaba	Polo	2281568		72548125	79663195	Cajera	

**Zonificación**

Núm. Ubicación	Descripcion_Sensor
001 Puerta General	1 Mag Industrial
002 Puerta principal y pasillo	1 Movimiento
003 Tienda lado izquierdo	2 Movimientos
004 Tienda Lado derecho	2 Movimientos
005 Caja	1 Botón de Pánico
006 Ventanas	2 S Rotasa de Vidrio

**Reporte** Desbloquear Guardar Reporte **Reportes Anteriores**

A. Hrs. 13:00 se llamo por demora, Clementina Cachicaba informo su cierre en seguida

A. Hrs. 13:00 se llamo por demora,\*informo su cierre en seguida

Cc	Nu	Cc	Reporte
17	1	03	A Hrs. 13:01 se llamo por demora, Clementina Cachicaba informo

**Fig. 4.4. Ventana Datos de la Cuenta**  
Fuente: Sistema de Monitoreo WOLF

## 2. Registro de datos

De aquí en adelante las ventanas mostradas son de la parte de administración del sistema de Monitoreo WOLF, en estas ventanas podemos registrar nuevas cuentas y/o modificar las ya existentes juntamente con sus horarios de apertura y cierre, usuarios, zonas y además encontramos también los datos de los operadores de la central de monitoreo.

### e) Ventana de autenticación del Administrador del Sistema:

La Fig. 4.5 muestra la ventana de autenticación, donde el administrador del sistema WOLF ingresa su clave y contraseña. Si el administrador del sistema ingresa los datos de forma errónea, el sistema WOLF emite un mensaje de error, pero si el error se repite tres veces, el sistema se cierra automáticamente.



**Fig. 4.5. Ventana de autenticación del Administrador del Sistema**  
Fuente: Sistema de Monitoreo WOLF

**f) Ventana Principal de Administración del Sistema:**

Una vez que el administrador del sistema WOLF ingresa su password correcto se muestra la ventana principal presentada en la Fig. 4.6. Esta ventana muestra el menú general para la administración de la base de datos.



**Fig. 4.6. Ventana Principal de Administración del Sistema**  
Fuente: Sistema de Monitoreo WOLF

g) Ventana Adicionar Nueva Cuenta :

En la ventana mostrada en la Fig. 4.7. se muestra el primer formulario para el proceso de adición de nueva cuenta a la base de datos del sistema WOLF. De acuerdo con las especificaciones de la base de datos del sistema el formulario exige el llenado completo de los cuadros de texto que son de color rojo antes de poder proceder a guardar los datos suministrados por el usuario del panel de alarma, los técnicos de instalación de alarmas y los datos como fecha de filiación y observaciones además de los horarios de inicio y fin de jornada en las cuentas de tipo 2 y 3.

**ADICIONAR NUEVA CUENTA**

**DATOS DE LA NUEVA CUENTA**

Cuenta:  Fecha de filiación:

Razon\_Social:  Telefono1:

NIT:  Telefono2:

CI:  Tipo de cuenta:

Zona:  E-mail:

Direccion:

Observaciones:

controlar horarios de Apertura-Cierre de la cuenta

**CONTROL DE HORARIOS** Control:

	MAÑANA		TARDE	
	Apertura	Cierre	Apertura	Cierre
Lunes-Viernes	<input type="text" value="08:00:00"/>	<input type="text" value="00:30:00"/>	<input type="text" value="14:00:00"/>	<input type="text" value="18:00:00"/>
Sabados	<input type="text" value="08:00:00"/>			<input type="text" value="18:00:00"/>
Domingos	<input type="text" value="08:00:00"/>			<input type="text" value="18:00:00"/>

Fig. 4.7. Ventana Adicionar Nueva Cuenta  
Fuente: Sistema de Monitoreo WOLF

h) Ventana Registrar Zonas:

La Fig. 4.8 muestra la ventana donde se registran todas las zonas que pertenecen a la cuenta adicionada en la ventana anterior. Este formulario exige que todos los campos en la tabla sean llenados antes de proceder a guardarlos en la base de datos.

Num. Zona	Ubicación de los sensores en el inmueble	Descripción de sensores
-----------	--	-------------------------

**Fig. 4.8. Ventana Registrar Zonas**  
**Fuente: Sistema de Monitoreo WOLF**

**i) Ventana Adicionar Usuario:**

La ventana mostrada en la Fig. 4.9 se muestra el formulario de adición de nuevos usuarios donde el administrador ingresa los datos del usuario, y en el caso que el usuario a registrarse en la cuenta ya este registrado como usuario de otra cuenta el administrador del sistema podrá buscarlo en la base de datos con uno de los siguientes datos: nombre, apellido o el numero de cuenta en la que fue registrado anteriormente. En este caso existen campos que modifican los datos en ambas cuentas los cuales están resaltados con color rojo y los datos que solo afectan a la cuenta seleccionada los cuales están resaltados con color verde.



**Adicionar\_Usuario V.2**

**DATOS PERSONALES DEL USUARIO 0001**

**Cuenta N#: 7000**

Número de Usuario En el Panel de Alarma

Nombre Del Usuario  Apellido Paterno  Apellido Materno

Palabra Clave  Descripción del Usuario  Observaciones

Teléfono Oficina  Teléfono Casa  Celular  Otro Teléfono

Adicionar y Continuar Guardar Cambios Eliminar

Adicionar y Finalizar Cancelar

Nota: Si el Usuario está presente en más de una cuenta, los datos de color Rojo producirán cambios en ambas cuentas mientras que los de color Verde solo afectarán a la cuenta especificada.

Seleccionar  Buscar Usuario

**Fig. 4.9 Ventana Adicionar Usuario**  
**Fuente: Sistema de Monitoreo WOLF**

**j) Ventana Registrar, Modificar y Eliminar Operadores de Central:**

A continuación en la Fig. 4.10 tenemos la ventana de registro, modificación y eliminación de operadores de la central de monitoreo a la cual podemos acceder directamente desde la ventana principal de la administración del sistema WOLF. En esta ventana el administrador llenará los datos personales suministrados por el operador y le asignará un password para poder ingresar al sistema de monitoreo WOLF.

**Operadores\_de\_Central**

**OPERADORES DE CENTRAL**

Nombre Clave  BUSCAR

Carnet de Identidad

Nombre(s)  Apellido Paterno  Apellido Materno

Teléfono Fijo  Teléfono Celular

Contraseña  Confirmar Contraseña

Adicionar Eliminar Guardar Cambios Cancelar

**Fig. 4.10. Ventana Registrar, Modificar y Eliminar Operadores de Central**  
**Fuente: Sistema de Monitoreo WOLF**



### 3. Generación de informes

Para esta parte contamos con una sola ventana la ventana de Reportes descrita a continuación:

#### k) Ventana Generación de reportes:

Finalmente en la Fig. 4.11 tenemos la ventana de generación de reportes, donde el administrador del sistema puede consultar los reportes elaborados por los operadores de central por cada cuenta o tipo de cuentas existente en la base de datos.



**Fig. 4.11. Ventana Generar Reportes**  
**Fuente: Sistema de Monitoreo WOLF**

### 4.3. PRUEBAS

Dado que las pruebas son el proceso de ejercitar el software con la intención de encontrar errores; se representa dichas pruebas de software por la actividad en la cual un sistema o uno de sus componentes se ejecutan en circunstancias previamente especificadas, cuyos resultados son observados y registrados.

#### 4.3.1. Prueba De Caja Blanca

Para las pruebas de caja blanca se eligió el criterio de cobertura de caminos con la aplicación de la técnica Prueba del Camino Básico basada en el cálculo de la complejidad descrita en el apartado 2.8.

Los pasos a realizar para aplicar esta técnica son:

**a) Representar el programa en un grafo de flujo**

El grafo de flujo se utiliza para representar el flujo de control lógico de un programa. En la Fig. 4.12 se muestra el grafo de flujo del módulo de Recepción de Eventos, y a continuación se detallan sus nodos:

1. Recepción de trama valida con inicio y fin.
2. Verificar tipo de trama.
3. Si la trama es de Testeo del D6600 termina el proceso.
4. Si es una trama no identificada se emite una señal de alarma y se muestra la trama.
5. Si la trama es de tipo "a" se considera valida y se extraen los datos.
6. Se busca el carácter separador " #" en la trama.
7. Si no existe se envía a 4.
8. Se extrae en la variable cuenta 4 caracteres según índices prefijados.
9. Si existe se guarda el valor en la variable cuenta.
10. Se busca el carácter separador " | " en la trama.
11. Si no existe se envía a 4 y luego a 8.
12. Si existe se guarda el valor en la variable ocp (Open/Close/Previous).
13. Se extrae el valor de la variable codigoevento según índices prefijados.
14. Se extrae el valor de la variable zonausuario según índices prefijados.
15. Se crea una cadena con todos los datos extraídos de la trama.
16. Buscar el Archivo.txt de la fecha actual.
17. Si el archivo no existe se emite una señal de alarma, y se genera un nuevo archivo.
18. Si el archivo existe se escribe la cadena dentro del archivo.
19. Convertir la cadena en Evento = char[7].
20. Validar variable ocp.
21. Si no es valida Emitir señal de alerta y asignarle un valor por defecto.
22. Se asigna una carga un valor textual en Evento[]

23. Consultar en la BD Tipo\_Evento (codigoevento).
24. Si no esta registrado el evento se emite una señal de alarma y se carga un valor por defecto en Evento[].
25. Crear un hilo para la reproducción de sonido. Y para la actualización de Estados.
26. Consultar en la BD Cuenta (cuenta)
27. Si no esta registrado el evento se emite una señal de alarma y se carga un valor por defecto en Evento[].
28. Guardar el valor en Evento[] y Verificar si es un evento de cambio de estado Apertura/Cierre.
29. Si no es, Consultar en la BD Zonas\_Cuenta (zonausuario).
30. Si no esta registrada la zona se emite una señal de alarma y se carga un valor por defecto en Evento[].
31. Guardar el valor en Evento[].
32. Consultar en la BD Usuario y Usuario\_Cuenta (zonausuario)
33. Si no esta registrado el usuario se emite una señal de alarma y se carga un valor por defecto en Evento[].
34. Guardar el valor en Evento[]. Y terminar de llenar Evento[] con los valores restantes #cuenta y zonausuario.
35. Mostrar el evento en la ventana de Monitoreo.

#### **b) Calcular la complejidad ciclomática**

La complejidad ciclomática es una métrica del software que proporciona una medida cuantitativa de la complejidad lógica de un programa. En el contexto del método de prueba del camino básico, el valor de la complejidad ciclomática define el número de caminos independientes de dicho programa, y por lo tanto, el número de casos de prueba a realizar.

Existen varias formas de calcular la complejidad ciclomática de un programa a partir de un grafo de flujo, una de las formas es:

$$V(G) = \text{Nodos Predicado} + 1$$

$$V(G) = \text{Nodos Predicado} + 1 = 10 + 1 = 11$$

Por tanto se determina que el número de casos de prueba que deben ejecutarse es igual a once, de tal manera que se pueda garantizar que todas las sentencias del módulo de Recepción de Eventos se han ejecutado al menos una vez.



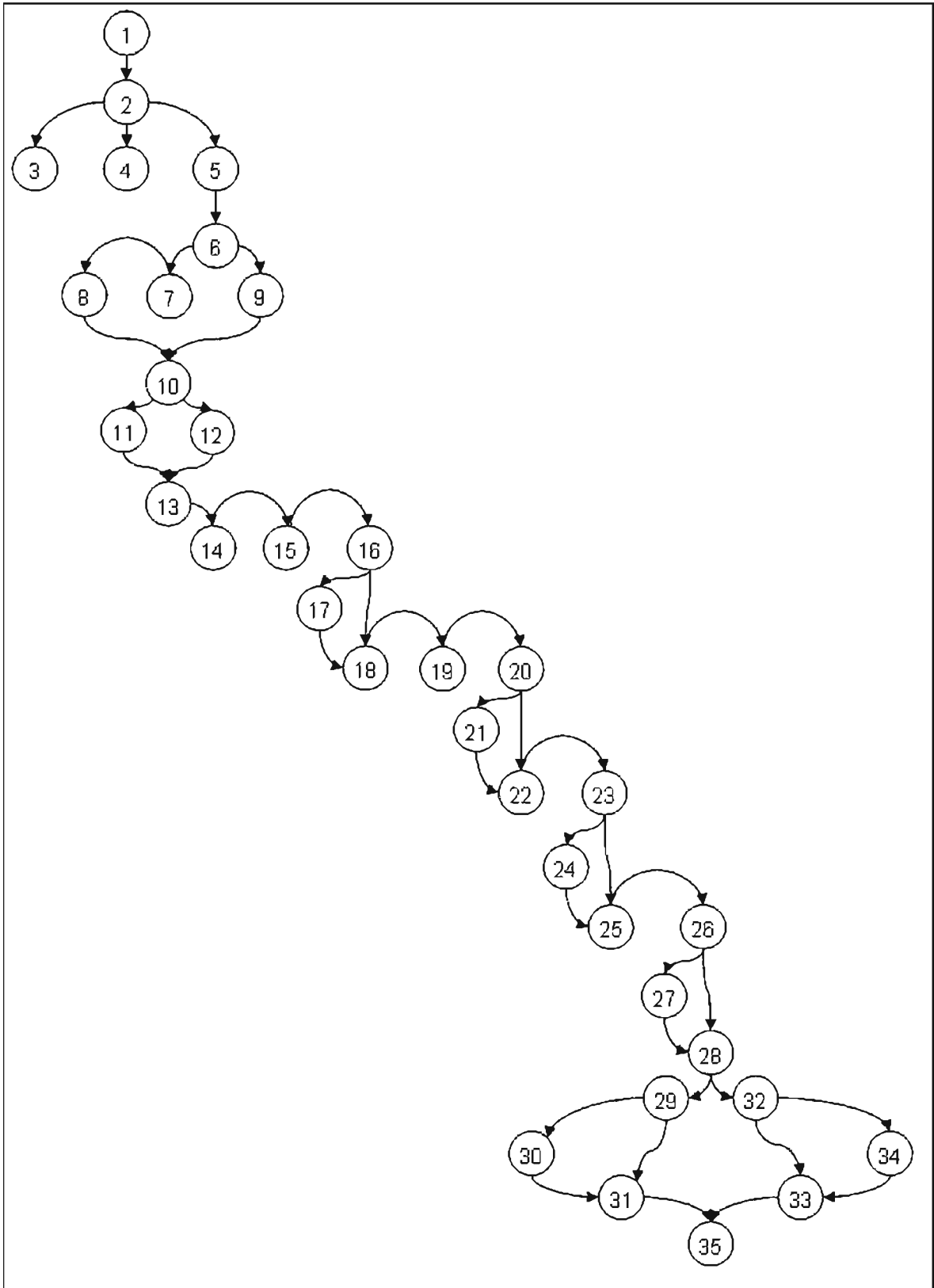


Fig. 4.12. Grafo de Flujo  
Fuente: Elaboración propia

c) Determinar el conjunto básico de caminos independientes

Un camino independiente es cualquier camino del programa que introduce, por lo menos, un nuevo conjunto de sentencias de proceso o una condición, respecto a los caminos existentes. En términos de diagrama de flujo, un camino independiente está construido por lo menos por una arista que no haya sido recorrida anteriormente a la definición del camino.

Entonces del paso anterior se obtuvo  $V(G) = 11$ , por tanto se realiza las pruebas que obliguen la ejecución de cada camino del conjunto básico.

Se hizo las pruebas de los doscientos cincuenta y nueve caminos de manera automática generando tramas con el Simulador D6600 y fueron enviadas por el puerto serial al sistema WOLF (ver Anexo A5)

**d) Derivar los casos de prueba que fuerzan la ejecución de cada camino.**

Por último debemos construir los casos de prueba que fuerzan la ejecución de cada camino. El conjunto de casos de prueba se muestra a continuación.

## **4.4. METRICAS DE CALIDAD**

### **4.4.1. Punto Función**

Las métricas del software orientadas a la función utilizan una medida de funcionalidad entregada por la aplicación como un valor de normalización. Ya que la “funcionalidad” no se puede medir directamente mediante otras medidas. Los puntos función se derivan de una relación empírica según las medidas contables (directas) del dominio de información del software y las evaluaciones de complejidad del software [Presman,2005]

Los puntos función se calculan completando la Tabla 4.3. Se determinan 5 características de dominios de información:

1. *Número de entradas de usuario.* Se cuenta cada entrada de usuario que proporciona diferentes datos orientados a la aplicación.



2. *Número de salidas de usuario.* Se cuenta cada salida que proporciona al usuario información orientada a la aplicación (Informes, pantallas, mensajes de error, etc.).
3. *Número de peticiones de usuario.* Una petición se define como una entrada interactiva que produce la generación de alguna respuesta del software inmediata en forma de salida interactiva. Se cuenta cada petición por separado.
4. *Número de archivos.* Se cuenta cada archivo maestro lógico.
5. *Número de interfaces externas.* Se cuentan todas las interfaces legibles por la maquina que se utilizan para transmitir información a otro sistema.

Parámetros de medida	Cuenta	Factor de peso			Total
		Simple	Medio	Complejo	
Número de entradas de usuario	11	3	4	6	33
Número de salidas de usuario	35	4	5	7	149
Número de peticiones al usuario	8	3	4	6	32
Número de archivos	9	7	10	15	63
Número de interfaces externas	2	5	7	10	14
<b>Total</b>					<b>291</b>

Tabla 4.1.: Calculo de punto función  
Fuente: Elaboración propia

Para calcular el punto función se utiliza la siguiente relación:

$$PF = TOTAL * (0.65 + 0.06 * \sum(Fi))$$

En donde cuenta-total es la suma de todas las entradas PF obtenidas en la Tabla 4.4.

Fi(i= 1 a 14) son "valores de ajuste de la complejidad" según las respuestas a las siguientes preguntas de acuerdo a la escala con rangos descritos en la Tabla 4.5.:

Factor	Escala
1. ¿Requiere el sistema copias de seguridad?	2
2. ¿Se requiere comunicación de datos?	5
3. ¿Existen funciones de procesamiento distribuido?	0
4. ¿Es crítico el rendimiento?	3
5. ¿Será ejecutado el sistema en un entorno operativo existente frecuentemente utilizado?	4

6. ¿Requiere el sistema entrada de datos interactivo?	1
7. ¿Requiere la entrada de datos iterativo que las transiciones de entrada se lleven acabo sobre múltiples pantallas o variadas operaciones?	4
8. ¿Se actualiza los archivos maestros en forma iterativa?	2
9. ¿Son complejos las entradas, las salidas, los archivos o las peticiones?	3
10. ¿Es compleja el procesamiento interno?	4
11. ¿Sea diseñado el código para ser reutilizable?	4
12. ¿Están incluidas en el diseño la conversión y la instalación?	3
13. ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	2
14. ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizados por el usuario?	5

Tabla 4.2.: Tabla de valores de ajuste de complejidad  
Fuente: Elaboración propia

0	1	2	3	4	5
Sin influencia	Incidental	Moderado	Medio	Significativo	Esencial

Tabla 4.3: valores de ajuste de la complejidad

Reemplazando en la ecuación de punto función se tiene:

$$PF = 291 * (0.65 + 0.06*(42))$$

$$PF = 291 * 2.52 = 922,47$$

Este resultado se debe ajustar a la curva normal:

$$\text{Ajuste} = \text{Cuenta Total} * \text{Valor Máximo} = 1024.32$$

$$\text{Funcionalidad} = PF/\text{Ajuste} = 922,47/1024.32 = 0.90 * 100\%$$

Se concluye que la funcionalidad o utilidad del sistema es de 90% de productividad con respecto al tamaño y calidad del sistema.

## CAPITULO V

### CONCLUSIONES Y RECOMENDACIONES

#### 5.1. CONCLUSIONES

El Sistema de Monitoreo WOLF, después de las pruebas piloto realizadas con el simulador del Receptor D6600 (Anexo A4) en primera instancia y posteriormente con el Receptor D6600 real, se puede apreciar los beneficios que aporta al departamento de monitoreo de la empresa CIT Seguridad en cumplimiento con los objetivos planteados en el Capítulo I.

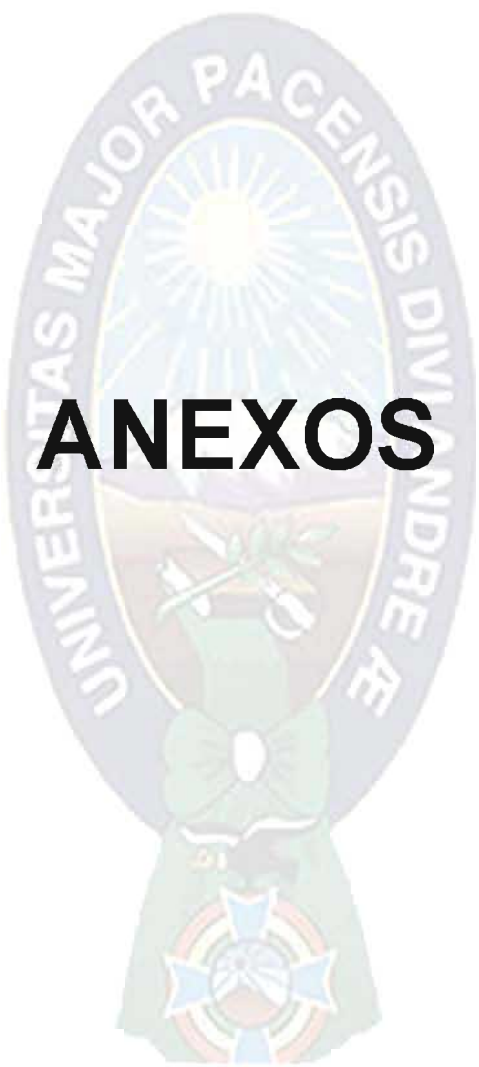
- Se construyó una interfaz de monitoreo en la cual podemos observar todos los eventos percibidos por el Receptor D6600 enlazados con una base de datos, de modo que el evento percibido pueda ser identificado claramente por el operador de la central de monitoreo.
- Dentro de la misma interfaz se generan alertas sonoras únicas para cada tipo de evento entrante que necesite un procedimiento de respuesta por parte del operador y en adición alertas visuales llamativas en la IU Monitoreo y la IU Estados.
- Se automatizó el proceso de control de Aperturas y Cierres de las cuentas que tienen este servicio, de modo que el sistema puede detectar el desactivado de una alarma (apertura) fuera de horario como un evento de alarma.
- El sistema permite la búsqueda de la información necesaria para comunicarse con el cliente (Datos de la cuenta, usuarios, zonas y reportes anteriores) gracias a la ventana "Datos de la Cuenta", de manera que el operador no tendrá que buscar manualmente los datos para realizar su procedimiento de respuesta.
- El sistema permite la elaboración y generación de reportes diarios, semanales y mensuales por cada cuenta y tipo de cuenta.
- La elección de la metodología RUP se ajustó de manera aceptable para la realización de este tipo de proyecto por ser iterativo e incremental. Y juntamente con la herramienta UML Mast se construyó un modelo lógico óptimo que permitió la implementación del sistema en el lenguaje C#.

Finalmente, al medir la calidad del software se obtuvo que la utilidad del sistema que es de 90% de productividad con respecto al tamaño y calidad del sistema. Esto significa que el sistema es eficiente y confiable en su aplicación.

## 5.2. RECOMENDACIONES

De acuerdo a la delimitación del proyecto de grado, y después del análisis de pruebas y en vista de las necesidades de la empresa CIT Seguridad en cuanto a la administración de la del sistema son las siguientes:

- El personal a cargo del departamento de monitoreo debe incorporar normas internas para el manejo del sistema a fin de protegerlo.
- Para alcanzar al máximo las bondades del sistema, será necesario registrar todos los atributos de cada tabla en la base de datos y además verificar si aun son validos, de modo que el sistema pueda desplegar toda la información necesaria y actual al operador.
- Identificar los horarios de sobrecarga de señales de manera que:
  - Se pueda establecer un horario para realizar cambios en la base de datos a través del módulo de administración de la base de datos.
  - Se pueda establecer el mejor horario para el envío de Test Periódicos por parte de los paneles de alarma.
  - Se pueda identificar las cuentas problemáticas y su impacto en la carga de señales en horarios de sobrecarga.
  - Se pueda estimar el flujo futuro de señales para tomar medidas preventivas.



# ANEXOS

## ANEXO A1

### LISTA DE EVENTOS

## D6600

### Appendix A: Contact-ID Event Code Classifications

Contact ID reporting takes the following format: CCCC Q EEE GG ZZZ

CCCC = customer (subscriber account number)

Q = event qualifier, E = new event, R = restore

EEE = event code

GG = partition number, 00-08 (always 00 for non-partitioned panels)

ZZZ = zone ID number reporting the alarm (001 -099), or user number for open/close reports.

\*System status messages (i.e. AC Loss, Low Battery) contain zeros in the ZZZ location.

SUBSCRIBER ID #	EVENT QUALIFIER	EVENT CODE	PARTITION #	ZONE OR USER #
XXXX	Event or Restore E or R	000	00	C000 or U000

The Event codes have been grouped according to the type of event, as described below.

MEDICAL:		
100	Medical	Emerg-Personal Emergency-#
101	Pendant Transmitter	Emerg-Personal Emergency-#
102	Fail to report in	Emerg-Fail to check in-#

FIRE ALARMS:		
110	FIRE	Fire-Fire Alarm-#
111	SMOKE w/VERIFICATION	Fire-Fire Alarm-#
112	Combustion	Fire-Combustion-#
113	WATERFLOW	Fire-Water Flow-#
114	Heat	Fire-Heat Sensor-#
115	Pull Station	Fire-Pull Station-#
116	Duct	Fire-Duct Sensor-#
117	Flame	Fire-Flame Sensor-#
118	Near Alarm	Fire-Near Alarm-#

PANIC ALARMS:		
120	Panic Alarm	Panic-Panic-#
121	DURESS	Panic-Duress-User-#
122	SILENT	Panic-Silent Panic-#
123	AUDIBLE	Panic-Audible Panic-#
124	Duress-Access Granted	Panic-Duress Access Grant-#
125	Duress-Egress Granted	Panic-Duress Egress Grant-#

BURGLAR ALARMS:		
130	Burglary	Burg-Burglary-#
131	PERIMETER	Burg-Perimeter-#
132	INTERIOR	Burg-Interior-#
133	24 HR BURG (AUX)	Burg-24 Hour-#
134	ENTRY/EXIT	Burg-Entry/Exit-#
135	DAY/NIGHT	Burg-Day/Night-#
136	Outdoor	Burg-Outdoor-#
137	TAMPER	Burg-Tamper-#
138	Near Alarm	Burg-Near Alarm-#
139	Intrusion Verifier	Burg-Intrusion Verifier-#

GENERAL ALARMS:		
140	General Alarm	Alarm-General Alarm-#
141	Polling Loop Open	Alarm-Polling Loop Open
142	POLLING LOOP SHORT (AL)	Alarm-Polling Loop Short
143	EXPANSION MOD FAILURE	Alarm-Exp. Module Tamper-#
144	Sensor Tamper	Alarm-Sensor Tamper-#
145	Expansion Module Tamper	Alarm-Exp. Module Tamper-#
146	SILENT BURG	Alarm-Silent Burglary-#

24 HOUR NON-BURGLARY:		
150	24 HOUR (AUXILIARY)	Alarm-24 Hr. Non-Burg-#
151	Gas Detected	Alarm-Gas Detected-#
152	Refrigeration	Alarm-Refrigeration-#
153	Loss of Heat	Alarm-Heating System-#
154	Water Leakage	Alarm-Water Leakage-#
155	Foil Break	Trouble-Foil Break-#
156	Day Trouble	Trouble-Day Zone-#
157	Low Bottled Gas Level	Alarm-Low Gas Level-#
158	High Temp	Alarm-High Temperature-#
159	Low Temp	Alarm-Low Temperature-#
161	Loss of Air Flow	Alarm-Air Flow-#
162	Carbon Monoxide Detected	Alarm-Carbon Monoxide-#
163	Tank Level	Trouble-Tank Level-#

FIRE SUPERVISORY:		
200	FIRE SUPERVISORY	Super.-Fire Supervisory-#
201	Low Water Pressure	Super.-Low Water Pressure-#
202	Low CO2	Super.-Low CO2-#
203	Gate Valve Sensor	Super.-Gate Valve-#
204	Low Water Level	Super.-Low Water Level-#
205	Pump Activated	Super.-Pump Activation-#
206	Pump Failure	Super.-Pump Failure-#



## ANEXO A1

### Appendix A

SYSTEM TROUBLES:		
300	System Trouble	Trouble-System Trouble
301	AC LOSS	Trouble-AC Power
302	LOW SYSTEM BATT	Trouble-Low Battery
303	RAM Checksum Bad	Trouble-Bad RAM Checksum (Restore not applicable)
304	ROM Checksum Bad	Trouble-Bad ROM Checksum (Restore not applicable)
305	SYSTEM RESET	Trouble-System Reset (Restore not applicable)
306	PANEL PROG CHANGE	Trouble-Programming Changed (Restore not applicable)
307	Self-Test Failure	Trouble-Self Test Failure
308	System Shutdown	Trouble-System Shutdown
309	Battery Test Fail	Trouble-Battery Test Failure
310	GROUND FAULT	Trouble-Ground Fault-#
311	Battery Missing	Trouble-Battery Missing
312	Power Supply Overcurrent	Trouble-Pwr. Supp. Overcurr.-#
313	Engineer Reset	Status-Engineer Reset-User # (Restore not applicable)

SOUNDER/RELAY TROUBLES:		
320	SOUNDER/RELAY	Trouble-Sounder/Relay-#
321	BELL 1	Trouble-Bell/Siren #1
322	BELL 2	Trouble-Bell/Siren #2
323	Alarm Relay	Trouble-Alarm Relay
324	Trouble Relay	Trouble-Trouble Relay
325	Reversing Relay	Trouble-Reversing Relay
326	Notification Appliance CKT. #3	Trouble-Notification Appl. Ckt#3
327	Notification Appliance CKT. #4	Trouble-Notification Appl. Ckt#4

SYSTEM PERIPHERAL TROUBLES:		
330	System Peripheral	Trouble-Sys. Peripheral-#
331	Polling Loop Open	Trouble-Polling Loop Open
332	POLLING LOOP SHORT	Trouble-Polling Loop Short
333	Exp. Module Failure	Trouble-Exp. Module Fail-#
334	Repeater Failure	Trouble-Repeater Failure-#
335	Local Printer Paper Out	Trouble-Printer Paper Out
336	Local Printer Failure	Trouble-Local Printer
337	EXP. MOD. DC LOSS	Trouble-Exp. Mod. DC Loss-#
338	EXP. MOD. LOW BAT	Trouble-Exp. Mod. Low Batt-#
339	EXP. MOD. RESET	Trouble-Exp. Mod. Reset-#
341	EXP. MOD. TAMPER	Trouble-Exp. Mod. Tamper-#
342	Exp. Module AC Loss	Trouble-Exp. Module AC Loss-#
343	Exp. Module Self Test Fail	Trouble-Exp. Self-Test Fail-#
344	RF Revr Jam Detect-#	Trouble-RF Revr Jam Detect-#

COMMUNICATION TROUBLES:		
350	Communication	Trouble-Communication Failure
351	TELCO 1 FAULT	Trouble-Phone Line #1
352	TELCO 2 FAULT	Trouble-Phone Line #2
353	LR Radio Xmitter Fault	Trouble-Radio Transmitter
354	FAILURE TO COMMUNICATE	Trouble-Fail to Communicate
355	Loss of Radio Superv.	Trouble-Radio Supervision
356	Loss of Central Polling	Trouble-Central Radio Polling
357	LRR XMTR. VSWR	Trouble-Radio Xmitter. VSWR-#

PROTECTION LOOP:		
370	Protection Loop	Trouble-Protection Loop-#
371	Protection Loop Open	Trouble-Prot. Loop Open-#
372	Protection Loop Short	Trouble-Prot. Loop Short-#
373	FIRE TROUBLE	Trouble-Fire Loop-#
374	EXIT ERROR (BY USER)	Alarm-Exit Error-#
375	Panic Zone Trouble	Trouble-PA Trouble-#
376	Hold-Up Zone Trouble	Trouble-Hold-Up Trouble-#

SENSOR:		
380	SENSOR TRBL-GLOBAL	Trouble-Sensor Trouble-#
381	LOSS OF SUPERVISION	Trouble-RF Sensor Super.-#
382	LOSS OF SUPERVSN	Trouble-RPM Sensor Super.-#
383	SENSOR TAMPER	Trouble-Sensor Tamper-#
384	RF LOW BATTERY	Trouble-RF Sensor Battery-#
385	SMOKE HI SENS.	Trouble-Smoke Hi Sens.-#
386	SMOKE LO SENS.	Trouble-Smoke Lo Sens.-#
387	INTRUSION HI SENS.	Trouble-Intrusion HI Sens.-#
388	INTRUSION LO SENS.	Trouble-Intrusion Lo Sens.-#
389	DET. SELF TEST FAIL	Trouble-Sensor Test Fail-#
391	Sensor Watch Failure	Trouble-Sensor Watch Fail-#
392	Drift Comp. Error	Trouble-Drift Comp. Error-#
393	Maintenance Alert	Trouble-Maintenance Alert-#

## ANEXO A1

### Appendix A

OPEN/CLOSE:		
400	Open/Close	Opening/Closing
401	OPEN/CLOSE BY USER	Opening-User #/Closing-User #
402	Group O/C	Closing-Group User #
403	AUTOMATIC OPEN/CLOSE	Opening-Automatic/Closing-Automatic
404	Late to O/C	Opening-Late/Closing-Late
405	Deferred O/C	Event & Restore Not Applicable
406	CANCEL (BY USER)	Opening-Cancel
407	REMOTE ARM/DISARM	Opening-Remote/Closing-Remote
408	QUICK ARM	Event not applicable for opening/closing-Quick Arm
409	KEYSWITCH OPEN/CLOSE	Opening-Keyswitch/Closing-Keyswitch
441	Armed Stay	Opening-Armed Stay/Closing-Armed Stay
442	Keyswitch Armed Stay	Opening-Keysw. Arm Stay/Opening-Keysw. Arm Stay
450	Exception O/C	Opening-Exception/Closing-Exception
451	Early O/C	Opening-Early/Closing-Early-User #
452	Late O/C	Opening-Late/Closing-Late-User #
453	Failed to Open	Trouble-Fail to open (Restore not applicable)
454	Failed to Close	Trouble-Fail to Close (Restore not applicable)
455	Auto-Arm Failed	Trouble-Auto Arm Failed (Restore not applicable)
456	Partial Arm	Closing-Partial Arm-User #
457	Exit Error (User)	Closing-Exit Error-User #
458	User on Premises	Opening-User on Prem. -User #
459	Recent Close	Trouble-Recent Close-User # (Restore not applicable)
461	Wrong Code Entry	Access-Wrong Code entry (Restore not applicable)
462	Legal Code Entry	Access-Legal Code entry-user # (Restore not applicable)
463	Re-arm after Alarm	Status-Re Arm After Alarm-User # (restore not applicable)
464	Auto Arm Time Extended	Status-Auto Arm Time ExtL-User # (Restore not applicable)
465	Panic Alarm Reset	Status-PA Reset (Restore not applicable)

REMOTE ACCESS:		
411	CALLBACK REQUESTED	Remote-Callback Requested (Restore not applicable)
412	Success-Download/access	Remote-Successful Access (Restore not applicable)
413	Unsuccessful Access	Remote-Unsuccessful Access (Restore not applicable)
414	System Shutdown	Remote-System Shutdown
415	Dialer Shutdown	Remote-Dialer Shutdown
416	Successful Upload	Remote-Successful Upload (Restore not applicable)

ACCESS CONTROL:		
421	Access Denied	Access-Access Denied-User # (Restore not applicable)
422	Access Report by User	Access-Access Granted-User # (Restore not applicable)
423	Forced Access	Panic-Forced Access-#
424	Egress Denied	Access-Egress Denied (Restore not applicable)
425	Egress Granted	Access-Egress Granted (Restore not applicable)
426	Access Door Propped Open	Access-Door Propped Open-#
427	Access Point DSM Trouble	Access-ACS Point DSM Trbl-#
428	Access Point RTE Trouble	Access-ACS Point RTE Trbl-#
429	Access Program Mode Entry	Access-ACS Prog. Entry-User # (Restore not applicable)
430	Access Program Mode Exit	Access-ACS Prog. Exit-User # (Restore not applicable)
431	Access Threat Level Change	Access-ACS Threat Level Chg.
432	Access Relay/Trigger Fail	Access-ACS Relay/Trig. Fail-#
433	Access RTE Shunt	Access-ACS RTE Shunt-#
434	Access DSM Shunt	Access-ACS DSM Shunt-#

SYSTEM DISABLES:		
501	Access Reader Disable	Disable-Access Rdr. Disable-#

SOUNDER/RELAY DISABLES:		
520	Sounder/Relay Disable	Disable-Sounder/Relay-#
521	Bell 1 Disable	Disable-Bell/Siren #1
522	Bell 2 Disable	Disable-Bell/Siren #2
523	Alarm Relay Disable	Disable-Alarm Relay
524	Trouble Relay Disable	Disable-Trouble Relay
525	Reversing Relay Disable	Disable-Reversing Relay
526	Notification Appliance Ckt #3	Disable-Notification Appl. Ckt #3
527	Notification Appliance Ckt #4	Disable-Notification Appl. Ckt #4

SYSTEM PERIPHERAL DISABLES:		
531	Module Added	Super.-Module Added (Restore not applicable)
532	Module Removed	Super.-Module Removed (Restore not applicable)

COMMUNICATION DISABLES:		
551	Dialer Disabled	Disable-Dialer Disable
552	Radio Xmitter Disabled	Disable-Radio Disable
553	Remote Upload/Download	Disable-Rem. Up/Download Disable

## ANEXO A1

### Appendix A

BYPASSES:		
570	ZONE/SENSOR BYPASS	Bypass-Zone Bypass-#
571	Fire Bypass	Bypass-Fire Bypass-#
572	24 Hour Zone Bypass	Bypass-24 Hour Bypass-#
573	Burg. Bypass	Bypass-Burg. Bypass-#
574	Group Bypass	Bypass-Group Bypass-User #
575	SWINGER BYPASS	Bypass-Swinger Bypass-#
576	Access Zone Shunt	Access-ACS Zone Shunt-#
577	Access Point Bypass	Access-ACS Point Bypass-#

TEST/MISC.:		
601	MANUAL TEST	Test-Manually Triggered (Restore not applicable)
602	PERIODIC TEST	Test-Periodic (Restore not applicable)
603	Periodic RF Xmission	Test-Periodic Radio (Restore not applicable)
604	PIRE TEST	Test-Fire Walk Test-User #
605	Status Report To Follow	Test-Fire Walk-User #
606	LISTEN-IN TO FOLLOW	Listen-Listen-In Active (Restore not applicable)
607	WALK-TEST MODE	Test-Walk Test Mode-User #
608	System Trouble Present	Test-System Trouble Present (Restore not applicable)
609	VIDEO XMTR ACTIVE	Listen-Video Xmitter Active (Restore not applicable)
611	POINT TESTED OK	Test-Point Tested OK-# (Restore not applicable)
612	POINT NOT TESTED	Test-Point Not Tested-# (Restore not applicable)
613	Intrusion Zone Walk Tested	Test-IntrnZone Walk Test-# (Restore not applicable)
614	Fire Zone Walk Tested	Test-Fire Zone Walk Test-# (Restore not applicable)
615	Panic Zone Walk Tested	Test-PA Zone Walk Test (Restore not applicable)
616	Service Request	Trouble-Service Request

EVENT LOG:		
621	EVENT LOG RESET	Trouble-Event Log Reset (Restore not applicable)
622	EVENT LOG 50% FULL	Trouble-Event Log 50% Full (Restore not applicable)
623	EVENT LOG 90% FULL	Trouble-Event Log 90% Full (Restore not applicable)
624	EVENT LOG OVERFLOW	Trouble-Event Log Overflow (Restore not applicable)
625	TIME/DATE RESET	Trouble-Time/Date Reset-User # (Restore not applicable)
626	TIME/DATE INACCURATE	Trouble-Time/Date Invalid
627	PROGRAM MODE ENTRY	Trouble-Program Mode Entry (Restore not applicable)
628	PROGRAM MODE EXIT	Trouble-Program Mode Exit (Restore not applicable)

SCHEDULING:		
630	Schedule Change	Trouble-Schedule Changed (Restore not applicable)
631	Exception Sched. Change	Trouble-Esc. Sched. Changed (Restore not applicable)
632	Access Schedule Changes	Trouble-Access Sched. Changed (Restore not applicable)

PERSONNEL MONITORING:		
641	Senior Watch Trouble	Trouble-Senior Watch Trouble
642	Latch-key Supervision	Status-Latch-key Super-User # (Restore not applicable)

SPECIAL CODES:	
750-789	These codes are used by Protection ONLY & can be assigned any unique non-standard Event code, which Pro 1 will be tracking.

ANEXO A2

TABLA DE DEMORAS

QLB POR DEMORA L-V												
FECHA ...../...../ 07												HORAS
CTA.	149	226	227									18:00
CTA.	211	231										18:30
CTA.	234	233										18:45
CTA.	129	177	223	238								19:00
CTA.	140	142	143	174	193	158	218	230	254	232	182	19:30
CTA.	252											19:40
CTA.	111	141	148	157	169	196	221	224	241			20:00
CTA.	228	229	225	256								
CTA.	106	109	130	131	132	135	136	137	144	146		20:30
CTA.	160	168	175	219	253	220	233					
CTA.	133	134	170	172	185	214	153					21:00
CTA.	123	147	161	164	166	235	236	258				21:30
CTA.	112	116	125	154	155	162	259					22:00
CTA.	150	197	209	145								22:30
CTA.	113	151	186	199	208	171	255	261				23:00
CTA.	120	163	167	192	206	122						23:30
CTA.	152	165	189	198	200	121	222					00:00
CTA.	117	138	183	257								00:30
CTA.	107											00:45
RESPONSABLE				VoBo								

## ANEXO A3

### ANALISIS DE UNA TRAMA VALIDA

[10]B4491Ca52290101[#0105]18340100001[[13]      [10]EE840B[9]52290101[[13]

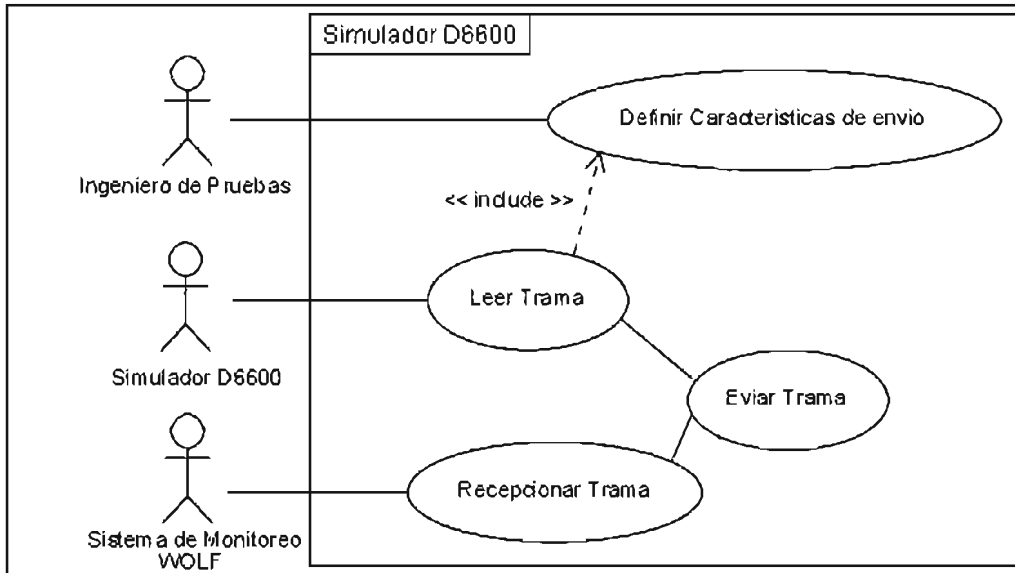
	A	B				C		D	E				F		G		Q	Q	H				Q	I		J		K				L		M				Q	N
d	[10]	B	4	4	9	1	C	a	5	2	2	9	0	1	0	1	[	#	0	1	0	5		1	8	3	4	0	1	0	0	0	0	0	1	]	[13]		
h	0A	42	34	34	39	31	43	61	35	32	32	39	30	31	30	31	5B	23	30	31	30	35	7C	31	38	33	34	30	31	30	30	30	30	31	5D	0D			
n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36			

	A	O				P	E				F		G		Q	Q	N		
d	[10]	E	E	8	4	0	B	[9]	5	2	2	9	0	1	0	1	[	]	[13]
h	0A	45	45	38	34	30	42	09	35	32	32	39	30	31	30	31	5B	5D	0D
n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

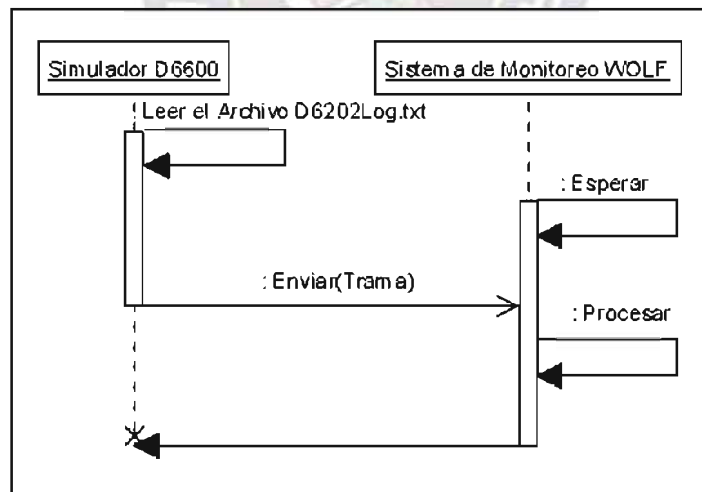
Donde d = trama en formato decimal, h = trama en formato hexadecimal y n = número de byte.

Casilla	Título	Descripción
A		Inicio de la trama.
B	LF	Carácter de inicio de línea estándar.
C	CRC	Numero verificador de redundancia cíclica.
D	Tipo de mensaje	a
E	Secuencia #	Numero de secuencia del mensaje. El rango valido de número de secuencia es de 0001 a 9999.
F	Receptor #	Receptor que envió el mensaje. Los dígitos validos son de 01 a 99 para el Receptor D6600.
G	Línea #	El numero de línea telefónica que envió el mensaje. Los dígitos validos son de 01 a 99. El D6600 usa Line# = 0 para los mensajes Receiver/Gateway.
H	Abonado #	Numero de abonado o cliente, 4 dígitos.
I	18	Formato numérico Ademco Contact-ID.
J	Calificador	Calificador, 1 = Nuevo Evento o Abierto, 3 = Nueva Restauración o Cierre, 6 = Evento Previo.
K	XYZ	Código de clase y código de evento (Anexo A1).
L	GG	Numero de grupo (Número de Partición).
M	CCC	Identificador de zona o de usuario.
N		Fin de trama.
O,P		No están especificados.
Q		Separadores

## ANEXO A4 SIMULADOR D6600



**Caso de uso del simulador D6600**



**Diagrama de secuencia del simulador D6600**



## ANEXO A5

### TABLA DE TRAMAS DE PRUEBA

Nro	Camino
1	1,2,3
2	1,2,4
3	1,2,5,6,9,10,12,13,14,15,16,18,19,20,22,23,25,26,28,32,34,33,35
4	1,2,5,6,9,10,12,13,14,15,16,18,19,20,22,23,25,26,28,29,30,31,35
5	1,2,5,6,9,10,12,13,14,15,16,18,19,20,21,22,23,25,26,28,32,34,33,35
6	1,2,5,6,9,10,12,13,14,15,16,18,19,20,21,22,23,25,26,27,28,29,30,31,35
7	1,2,5,6,9,10,12,13,14,15,16,18,19,20,21,22,23,24,25,26,28,32,34,33,35
8	1,2,5,6,9,10,12,13,14,15,16,17,18,19,20,22,23,25,26,28,32,34,33,35
9	1,2,5,6,9,10,12,13,14,15,16,17,18,19,20,22,23,25,26,28,29,31,35
10	1,2,5,6,9,10,11,13,14,15,16,18,19,20,22,23,25,26,28,32,34,33,35
11	1,2,5,6,7,8,10,12,13,14,15,16,18,19,20,22,23,25,26,28,32,34,33,35

Donde 9001 representa una cuenta registrada y 9999 representa una cuenta no registrada

Nro	Trama
1	[0]6FA410[9]51710101[NYB12][13] [10]E1D90B[9]51710101[13]
2	[10]A1D80B[9]51800100[13] [10]A1D80B[9]51800100[13]
3	[10B4491Ca52290101[#9001 xxx40100xxx][13] [10]EE840B[9]52290101[13]
4	[10B4491Ca52290101[#9001 xxx40100xxx][13] [10]EE840B[9]52290101[13]
5	[10B4491Ca52290101[#9001 18340000xxx][13] [10]EE840B[9]52290101[13]
6	[10B4491Ca52290101[#9999 18340000xxx][13] [10]EE840B[9]52290101[13]
7	[10B4491Ca52290101[#9001 183XX00xxx][13] [10]EE840B[9]52290101[13]
8	[10B4491Ca52290101[#9001 xxx13000xxx][13] [10]EE840B[9]52290101[13]
9	[10B4491Ca52290101[#9001 xxx13000001][13] [10]EE840B[9]52290101[13]
10	[10B4491Ca52290101[#9001 Xxxx30100xxx][13] [10]EE840B[9]52290101[13]
11	[10B4491Ca52290101[X9001 xxx44100xxx][13] [10]EE840B[9]52290101[13]

#### Tramas de pruebas de funcionalidad

Nro	Trama
1	[10B4491Ca52290101[#9001 18110000001][13] [10]EE840B[9]52290101[13]
2	[10B4491Ca52290101[#9001 18110100001][13] [10]EE840B[9]52290101[13]
3	[10B4491Ca52290101[#9001 18110200001][13] [10]EE840B[9]52290101[13]
4	[10B4491Ca52290101[#9001 18111000001][13] [10]EE840B[9]52290101[13]
5	[10B4491Ca52290101[#9001 18111200001][13] [10]EE840B[9]52290101[13]
6	[10B4491Ca52290101[#9001 18111300001][13] [10]EE840B[9]52290101[13]
7	[10B4491Ca52290101[#9001 18113600001][13] [10]EE840B[9]52290101[13]
8	[10B4491Ca52290101[#9001 18115200001][13] [10]EE840B[9]52290101[13]
9	[10B4491Ca52290101[#9001 18115300001][13] [10]EE840B[9]52290101[13]
10	[10B4491Ca52290101[#9001 18115500001][13] [10]EE840B[9]52290101[13]
11	[10B4491Ca52290101[#9001 18115700001][13] [10]EE840B[9]52290101[13]
12	[10B4491Ca52290101[#9001 18115900001][13] [10]EE840B[9]52290101[13]
13	[10B4491Ca52290101[#9001 18120200001][13] [10]EE840B[9]52290101[13]
14	[10B4491Ca52290101[#9001 18120300001][13] [10]EE840B[9]52290101[13]
15	[10B4491Ca52290101[#9001 18120500001][13] [10]EE840B[9]52290101[13]
16	[10B4491Ca52290101[#9001 18120600001][13] [10]EE840B[9]52290101[13]
17	[10B4491Ca52290101[#9001 18130300001][13] [10]EE840B[9]52290101[13]
18	[10B4491Ca52290101[#9001 18130500001][13] [10]EE840B[9]52290101[13]
19	[10B4491Ca52290101[#9001 18130600001][13] [10]EE840B[9]52290101[13]
20	[10B4491Ca52290101[#9001 18132000001][13] [10]EE840B[9]52290101[13]
21	[10B4491Ca52290101[#9001 18132700001][13] [10]EE840B[9]52290101[13]
22	[10B4491Ca52290101[#9001 18133000001][13] [10]EE840B[9]52290101[13]
23	[10B4491Ca52290101[#9001 18133200001][13] [10]EE840B[9]52290101[13]
24	[10B4491Ca52290101[#9001 18133300001][13] [10]EE840B[9]52290101[13]
25	[10B4491Ca52290101[#9001 18133500001][13] [10]EE840B[9]52290101[13]



91	[10B4491Ca52290101#9001 1831560000 1][13]	[10]EE840B[9]52290101[13]
92	[10B4491Ca52290101#9001 1834230000 1][13]	[10]EE840B[9]52290101[13]
93	[10B4491Ca52290101#9001 1834610000 1][13]	[10]EE840B[9]52290101[13]
Alerta 2		
94	[10B4491Ca52290101#9001 1813000000 1][13]	[10]EE840B[9]52290101[13]
95	[10B4491Ca52290101#9001 1813010000 1][13]	[10]EE840B[9]52290101[13]
96	[10B4491Ca52290101#9001 1813020000 1][13]	[10]EE840B[9]52290101[13]
97	[10B4491Ca52290101#9001 1813080000 1][13]	[10]EE840 B[9]52290101[13]
98	[10B4491Ca52290101#9001 1813090000 1][13]	[10]EE840B[9]52290101[13]
99	[10B4491Ca52290101#9001 1813100000 1][13]	[10]EE840B[9]52290101[13]
100	[10B4491Ca52290101#9001 1813110000 1][13]	[10]EE840B[9]52290 101[13]
101	[10B4491Ca52290101#9001 1813120000 1][13]	[10]EE840B[9]52290101[13]
102	[10B4491Ca52290101#9001 1813500000 1][13]	[10]EE840B[9]52290101[13]
103	[10B4491Ca52290101#9001 1813510000 1][13]	[10]EE840B[9]52290101[13]
104	[10B4491Ca52290101#9001 1813520000 1][13]	[10]EE840B[9]52290101[13]
105	[10B4491Ca52290101#9001 1813540000 1][13]	[10]EE840B[9]52290101[13]
106	[10B4491Ca52290101#9001 1833000000 1][13]	[10]EE840B[9]52290101[13]
107	[10B4491Ca52290101#9001 1833010000 1][13]	[10]EE840B[9]52290101[13]
108	[10B4491Ca52290101#9001 1833020000 1][13]	[10]EE840B[9]52290101[13]
109	[10B4491Ca52290101#9001 1833080000 1][13]	[10]EE840B[9]52290101[13]
110	[10B4491Ca52290101#9001 1833090000 1][13]	[10]EE840B[9]52290101[13]
111	[10B4491Ca52290101#9001 1833100000 1][13]	[10]EE840B[9]52290101[13]
112	[10B4491Ca52290101#9001 1833110000 1][13]	[10]EE840B[9]52290101[13]
113	[10B4491Ca52290101#9001 1833120000 1][13]	[10]EE840B[9]52290101[13]
114	[10B4491Ca52290101#9001 1833500000 1][13]	[10]EE840B[9]52290101[13]
115	[10B4491Ca52290101#9001 1833510000 1][13]	[10]EE840B[9]52290101[13]
116	[10B4491Ca52290101#9001 1833520000 1][13]	[10]EE840B[9]52290101[13]
117	[10B4491Ca52290101#9001 1833540000 1][13]	[10]EE840B[9]52290101[13]
Alerta 3		
118	[10B4491Ca52290101#9001 1814000000 1][13]	[10]EE840B[9]52290101[13]
119	[10B4491Ca52290101#9001 1814010000 1][13]	[10]EE840B[9]52290101[13]
120	[10B4491Ca52290101#9001 1814020000 1][13]	[10]EE840B[9]52290101[13]
121	[10B4491Ca52290101#9001 1814030000 1][13]	[10]EE840B[9]52290101[13]
122	[10B4491Ca52290101#9001 1814070000 1][13]	[10]EE840B[9]52290101[13]
123	[10B4491Ca52290101#9001 1814410000 1][13]	[10]EE840B[9]52290101[13]
124	[10B4491Ca52290101#9001 1834000000 1][13]	[10]EE840B[9]52290101[13]
125	[10B4491Ca52290101#9001 1834010000 1][13]	[10]EE840B[9]52290101[13]
126	[10B4491Ca52290101#9001 1834020000 1][13]	[10]EE840B[9]52290101[13]
127	[10B4491Ca52290101#9001 1834030000 1][13]	[10]EE840B[9]52290101[13]
128	[10B4491Ca52290101#9001 1834070000 1][13]	[10]EE840B[9]52290101[13]
129	[10B4491Ca52290101#9001 1834410000 1][13]	[10]EE840B[9]52290101[13]
Alerta 4		
130	[10B4491Ca52290101#9001 1815700000 1][13]	[10]EE840B[9]52290101[13]
131	[10B4491Ca52290101#9001 1815720000 1][13]	[10]EE840B[9]52290101[13]
132	[10B4491Ca52290101#9001 1815740000 1][13]	[10]EE840B[9]52290101[13]
133	[10B4491Ca52290101#9001 1835700000 1][13]	[10]EE840B[9]52290101[13]
134	[10B4491Ca52290101#9001 1835720000 1][13]	[10]EE840B[9]52290101[13]
135	[10B4491Ca52290101#9001 1835740000 1][13]	[10]EE840B[9]52290101[13]
Alerta5		
136	[10B4491Ca52290101#9001 1816020000 1][13]	[10]EE840B[9]52290101[13]
137	[10B4491Ca52290101#9001 1836020000 1][13]	[10]EE840B[9]52290101[13]
Alerta sin cuenta registrada		
138	[10B4491Ca52290101#9999 1834000000 1][13]	[10]EE840B[9]52290101[13]

Finalmente tenemos las pruebas de tiempo de respuesta. Para lo cual mostramos la siguiente tabla donde la columna T representa el tiempo de respuesta del sistema en función de su llegada con la hora del sistema (hh:mm:ss milisegundos) y la columna D representa la diferencia entre el tiempo de llegada y procesamiento de un evento  $T_i$  y un evento  $T_{i-1}$  tal que  $i > 0$ ; esta diferencia esta expresada en milisegundos. Para este propósito se enviaron los eventos uno detrás de otro sin tiempo de espera, utilizando el simulador D6600.

T	D
01 36 25 78	
01 36 25 62	16
01 36 24 937	125
01 36 24 937	0
01 36 24 937	0
01 36 24 937	0
01 36 24 921	16
01 36 24 921	0
01 36 24 921	0
01 36 24 921	0
01 36 24 906	15
01 36 24 906	0
01 36 24 906	0
01 36 24 906	0
01 36 24 890	16
01 36 24 890	0
01 36 24 890	0
01 36 24 890	0
01 36 24 875	15
01 36 24 875	0
01 36 24 875	0
01 36 24 875	0
01 36 24 859	16
01 36 24 859	0
01 36 24 859	0
01 36 24 859	0
01 36 24 843	16
01 36 24 843	0
01 36 24 843	0
01 36 24 828	15
01 36 24 828	0
01 36 24 828	0
01 36 24 812	16
01 36 24 796	16
01 36 24 781	15
01 36 24 781	0
01 36 24 765	16
01 36 24 750	15
01 36 24 750	0
01 36 24 734	16
01 36 24 734	0
01 36 24 734	0
01 36 24 718	16
01 36 24 718	0
01 36 24 718	0
01 36 24 703	15
01 36 24 703	0
01 36 24 703	0
01 36 24 687	16
01 36 24 687	0
01 36 24 687	0
01 36 24 687	0
01 36 24 671	16
01 36 24 671	0
01 36 24 671	0
01 36 24 656	15
01 36 24 656	0
01 36 24 656	0
01 36 24 656	0
01 36 24 640	16
01 36 24 640	0
01 36 24 640	0
01 36 24 640	0
01 36 24 625	15
01 36 24 625	0
01 36 24 625	0
01 36 24 609	16
01 36 24 609	0
01 36 24 609	0
01 36 24 609	0
01 36 24 593	16
01 36 24 593	0
01 36 24 593	0

01 36 24 593	0
01 36 24 578	15
01 36 24 578	0
01 36 24 578	0
01 36 24 578	0
01 36 24 562	16
01 36 24 562	0
01 36 24 562	0
01 36 24 546	16
01 36 24 546	0
01 36 24 546	0
01 36 24 546	0
01 36 24 531	15
01 36 24 531	0
01 36 24 531	0
01 36 24 531	0
01 36 24 515	16
01 36 24 515	0
01 36 24 515	0
01 36 24 515	0
01 36 24 500	15
01 36 24 500	0
01 36 24 484	16
01 36 24 468	16
01 36 24 453	15
01 36 24 453	0
01 36 24 437	16
01 36 24 421	16
01 36 24 406	15
01 36 24 406	0
01 36 24 390	16
01 36 24 375	15
01 36 24 359	16
01 36 24 359	0
01 36 24 343	16
01 36 24 328	15
01 36 24 312	16
01 36 24 312	0
01 36 24 296	16
01 36 24 281	15
01 36 24 265	16
01 36 24 265	0
01 36 24 250	15
01 36 24 234	16
01 36 24 218	16
01 36 24 203	15
01 36 24 203	0
01 36 24 187	16
01 36 24 171	16
01 36 24 156	15
01 36 24 156	0
01 36 24 140	16
01 36 24 125	15
01 36 24 109	16
01 36 24 93	16
01 36 24 93	0
01 36 24 78	15
01 36 24 62	16
01 36 24 46	16
01 36 24 46	0
01 36 24 31	15
01 36 24 15	16
01 36 24 0	15
01 36 24 0	0
01 36 23 984	16
01 36 23 968	16
01 36 23 953	15
01 36 23 953	0
01 36 23 937	16
01 36 23 921	16
01 36 23 906	15
01 36 23 906	0
01 36 23 890	16
01 36 23 875	15
01 36 23 859	16
01 36 23 859	0
01 36 23 843	16

01 36 23 828	15
01 36 23 812	16
01 36 23 796	16
01 36 23 796	0
01 36 23 781	15
01 36 23 765	16
01 36 23 750	15
01 36 23 750	0
01 36 23 734	16
01 36 23 718	16
01 36 23 703	15
01 36 23 703	0
01 36 23 687	16
01 36 23 671	16
01 36 23 656	15
01 36 23 656	0
01 36 23 640	16
01 36 23 625	15
01 36 23 609	16
01 36 23 609	0
01 36 23 593	16
01 36 23 578	15
01 36 23 562	16
01 36 23 562	0
01 36 23 546	16
01 36 23 531	15
01 36 23 515	16
01 36 23 515	0
01 36 23 500	15
01 36 23 484	16
01 36 23 468	16
01 36 23 453	15
01 36 23 453	0
01 36 23 437	16
01 36 23 421	16
01 36 23 406	15
01 36 23 406	0
01 36 23 390	16
01 36 23 375	15
01 36 23 359	16
01 36 23 359	0
01 36 23 343	16
01 36 23 343	0
01 36 23 343	0
01 36 23 343	0
01 36 23 328	15
01 36 23 328	0
01 36 23 312	16
01 36 23 312	0
01 36 23 312	0
01 36 23 312	0
01 36 23 296	16
01 36 23 296	0
01 36 23 281	15
01 36 23 265	16
01 36 23 250	15
01 36 23 250	0
01 36 23 234	16
01 36 23 234	0
01 36 23 218	16
01 36 23 218	0
01 36 23 218	0
01 36 23 218	0
01 36 23 203	15
01 36 23 203	0
01 36 23 203	0
01 36 23 187	16
01 36 23 187	0
01 36 23 171	16
01 36 23 171	0
01 36 23 171	0
01 36 23 156	15
01 36 23 156	0
01 36 23 156	0
01 36 23 140	16
01 36 23 140	0
01 36 23 140	0

01 36 23 140	0
01 36 23 125	15
01 36 23 125	0
01 36 23 125	0
01 36 23 125	0
01 36 23 109	16
01 36 23 109	0
01 36 23 109	0
01 36 23 93	16
01 36 23 93	0
01 36 23 93	0
01 36 23 93	0
01 36 23 78	15
01 36 22 984	84
01 36 22 984	0
01 36 22 984	0
01 36 22 968	16
01 36 22 968	0
01 36 22 968	0
01 36 22 968	0
01 36 22 953	15
01 36 22 953	0
01 36 22 953	0
01 36 22 937	16
01 36 22 937	0
01 36 22 937	0
01 36 22 937	0
01 36 22 921	16
01 36 22 921	0
01 36 22 921	0
01 36 22 921	0
01 36 22 906	15
01 36 22 906	0
01 36 22 906	0
01 36 22 906	0
01 36 22 890	16
01 36 22 890	0
01 36 22 875	15
01 36 22 859	16
01 36 22 859	0
01 36 22 843	16
01 36 22 828	15
01 36 22 812	16
01 36 22 812	0
01 36 22 796	16
01 36 22 781	15
01 36 22 765	16
01 36 22 765	0
01 36 22 750	15
01 36 22 734	16
01 36 22 718	16
01 36 22 718	0
01 36 22 703	15
01 36 22 687	16
01 36 22 687	0
01 36 22 671	16
01 36 22 671	0
01 36 22 656	15
01 36 22 640	16
01 36 22 640	0
01 36 22 625	15
01 36 22 609	16
01 36 22 593	16
01 36 22 578	15
01 36 22 578	0
01 36 22 562	16
01 36 22 546	16
01 36 22 531	15
01 36 22 531	0
01 36 22 515	16
01 36 22 500	15
01 36 22 484	16
01 36 22 484	0
01 36 22 468	16
01 36 22 453	15

01 36 22 437	16
01 36 22 437	0
01 36 22 421	16
01 36 22 406	15
01 36 22 390	16
01 36 22 390	0
01 36 22 375	15
01 36 22 359	16
01 36 22 343	16
01 36 22 343	0
01 36 22 328	15
01 36 22 312	16
01 36 22 296	16
01 36 22 296	0
01 36 22 281	15
01 36 22 265	16
01 36 22 250	15
01 36 22 250	0
01 36 22 234	16
01 36 22 218	16
01 36 22 218	0
01 36 22 203	15
01 36 22 187	16
01 36 22 171	16
01 36 22 156	15
01 36 22 156	0
01 36 22 140	16
01 36 22 125	15
01 36 22 125	0
01 36 22 109	16
01 36 22 93	16
01 36 22 78	15
01 36 22 78	0
01 36 22 62	16
01 36 22 46	16
01 36 22 31	15
01 36 22 31	0
01 36 22 15	16
01 36 21 0	15
01 36 21 984	16
01 36 21 984	0
01 36 21 968	16
01 36 21 953	15
01 36 21 937	16
01 36 21 937	0
01 36 21 921	16
01 36 21 906	15
01 36 21 890	16
01 36 21 890	0
01 36 21 875	15
01 36 21 859	16
01 36 21 859	0
01 36 21 843	16
01 36 21 828	15
01 36 21 812	16
01 36 21 812	0
01 36 21 796	16
01 36 21 781	15
01 36 21 765	16
01 36 21 765	0
01 36 21 765	0
01 36 21 765	0
01 36 21 750	15
01 36 21 750	0
01 36 21 750	0
01 36 21 734	16
01 36 21 734	0
01 36 21 734	0
01 36 21 734	0
01 36 21 718	16
01 36 21 718	0
01 36 21 703	15
01 36 21 687	16
01 36 21 671	16
01 36 21 671	0
01 36 21 656	15
01 36 21 656	0
01 36 21 640	16

01 36 21 640	0
01 36 21 640	0
01 36 21 640	0
01 36 21 625	15
01 36 21 625	0
01 36 21 625	0
01 36 21 625	0
01 36 21 625	0
01 36 21 625	0
01 36 21 609	16
01 36 21 609	0
01 36 21 609	0
01 36 21 609	0
01 36 21 593	16
01 36 21 593	0
01 36 21 593	0
01 36 21 593	0
01 36 21 578	15
01 36 21 578	0
01 36 21 578	0
01 36 21 578	0
01 36 21 578	0
01 36 21 562	16
01 36 21 562	0
01 36 21 562	0
01 36 21 562	0
01 36 21 546	16
01 36 21 546	0
01 36 21 546	0
01 36 21 546	0
01 36 21 546	0
01 36 21 531	15
01 36 21 531	0
01 36 21 531	0
01 36 21 531	0
01 36 21 531	0
01 36 21 515	16
01 36 21 515	0
01 36 21 515	0
01 36 21 515	0
01 36 21 515	0
01 36 21 500	15
01 36 21 500	0
01 36 21 500	0
01 36 21 500	0
01 36 21 500	0
01 36 21 484	16
01 36 21 484	0
01 36 21 484	0
01 36 21 484	0
01 36 21 484	0
01 36 21 468	16
01 36 21 468	0
01 36 21 468	0
01 36 21 468	0
01 36 21 468	0
01 36 21 453	15
01 36 21 453	0
01 36 21 453	0
01 36 21 453	0
01 36 21 453	0
01 36 21 437	16
01 36 21 421	16
01 36 21 421	0
01 36 21 406	15
01 36 21 390	16
01 36 21 375	15
01 36 21 375	0
01 36 21 359	16
01 36 21 343	16
01 36 21 343	0
01 36 21 328	15
01 36 21 312	16
01 36 21 296	16
01 36 21 296	0
01 36 21 281	15
01 36 21 265	16
01 36 21 265	0
01 36 21 250	15
01 36 21 234	16
01 36 21 234	0
01 36 21 218	16

01 36 21 218	0
01 36 21 203	15
01 36 21 187	16
01 36 21 171	16
01 36 21 171	0
01 36 21 156	15
01 36 21 140	16
01 36 21 140	0
01 36 21 125	15
01 36 21 109	16
01 36 21 93	16
01 36 21 93	0
01 36 21 78	15
01 36 21 62	16
01 36 21 46	16
01 36 21 46	0
01 36 21 31	15
01 36 21 15	16
01 36 21 15	0
01 36 21 0	15
01 36 20 984	16
01 36 20 968	16
01 36 20 968	0
01 36 20 953	15
01 36 20 937	16
01 36 20 937	0
01 36 20 921	16
01 36 20 906	15
01 36 20 890	16
01 36 20 890	0
01 36 20 875	15
01 36 20 859	16
01 36 20 859	0
01 36 20 843	16
01 36 20 828	15
01 36 20 812	16
01 36 20 812	0
01 36 20 796	16
01 36 20 781	15
01 36 20 781	0
01 36 20 765	16
01 36 20 750	15
01 36 20 734	16
01 36 20 734	0
01 36 20 718	16
01 36 20 703	15
01 36 20 703	0
01 36 20 687	16
01 36 20 671	16
01 36 20 656	15
01 36 20 656	0
01 36 20 640	16
01 36 20 625	15
01 36 20 625	0
01 36 20 609	16
01 36 20 593	16
01 36 20 578	15
01 36 20 578	0
01 36 20 562	16
01 36 20 546	16
01 36 20 546	0
01 36 20 531	15
01 36 20 515	16
01 36 20 500	15
01 36 20 500	0
01 36 20 484	16
01 36 20 468	16
01 36 20 468	0
01 36 20 453	15
01 36 20 437	16
01 36 20 421	16
01 36 20 421	0
01 36 20 406	15
01 36 20 390	16
01 36 20 390	0
01 36 20 375	15
01 36 20 359	16
01 36 20 359	0

01 36 20 359	0
01 36 20 359	0
01 36 20 343	16
01 36 20 328	15
01 36 20 328	0
01 36 20 312	16
01 36 20 312	0
01 36 20 312	0
01 36 20 312	0
01 36 20 296	16
01 36 20 296	0
01 36 20 281	15
01 36 20 265	16
01 36 20 265	0
01 36 20 250	15
01 36 20 234	16
01 36 20 234	0
01 36 20 234	0
01 36 20 218	16
01 36 20 218	0
01 36 20 218	0
01 36 20 218	0
01 36 20 218	0
01 36 20 203	15
01 36 20 203	0
01 36 20 203	0
01 36 20 203	0
01 36 20 187	16
01 36 20 187	0
01 36 20 187	0
01 36 20 187	0
01 36 20 187	0
01 36 20 171	16
01 36 20 171	0
01 36 20 171	0
01 36 20 171	0
01 36 20 156	15
01 36 20 156	0
01 36 20 156	0
01 36 20 156	0
01 36 20 140	16
01 36 20 140	0
01 36 20 140	0
01 36 20 140	0
01 36 20 125	15
01 36 20 125	0
01 36 20 125	0
01 36 20 125	0
01 36 20 125	0
01 36 20 109	16
01 36 20 109	0
01 36 20 109	0
01 36 20 109	0
01 36 20 109	0
01 36 20 93	16
01 36 20 93	0
01 36 20 93	0
01 36 20 93	0
01 36 20 93	0
01 36 20 78	15
01 36 20 78	0
01 36 20 78	0
01 36 20 78	0
01 36 20 78	0
01 36 20 62	16
01 36 20 62	0
01 36 20 62	0
01 36 20 62	0
01 36 20 62	0
01 36 20 46	16
01 36 20 46	0
01 36 20 46	0
01 36 20 46	0
01 36 20 31	15
01 36 20 15	16

01 36 20 15	0
01 36 19 0	15
01 36 19 984	16
01 36 19 984	0
01 36 19 968	16
01 36 19 953	15
01 36 19 953	0
01 36 19 937	16
01 36 19 921	16
01 36 19 921	0
01 36 19 906	15
01 36 19 890	16
01 36 19 875	15
01 36 19 875	0
01 36 19 859	16
01 36 19 843	16
01 36 19 843	0
01 36 19 828	15
01 36 19 812	16
01 36 19 812	0
01 36 19 812	0
01 36 19 796	16
01 36 19 781	15
01 36 19 765	16
01 36 19 765	0
01 36 19 750	15
01 36 19 734	16
01 36 19 734	0
01 36 19 718	16
01 36 19 703	15
01 36 19 687	16
01 36 19 687	0
01 36 19 671	16
01 36 19 656	15
01 36 19 656	0
01 36 19 640	16
01 36 19 625	15
01 36 19 625	0
01 36 19 609	16
01 36 19 593	16
01 36 19 578	15
01 36 19 578	0
01 36 19 562	16
01 36 19 546	16
01 36 19 546	0
01 36 19 531	15
01 36 19 515	16
01 36 19 515	0
01 36 19 500	15
01 36 19 484	16
01 36 19 468	16
01 36 19 468	0
01 36 19 453	15
01 36 19 437	16
01 36 19 437	0
01 36 19 421	16
01 36 19 406	15
01 36 19 390	16
01 36 19 390	0
01 36 19 375	15
01 36 19 359	16
01 36 19 359	0
01 36 19 343	16
01 36 19 328	15
01 36 19 312	16
01 36 19 312	0
01 36 19 296	16
01 36 19 281	15
01 36 19 265	16
01 36 19 265	0
01 36 19 250	15
01 36 19 234	16
01 36 19 234	0
01 36 19 218	16
01 36 19 203	15
01 36 19 140	63
01 36 19 31	109
01 36 19 15	16
01 36 19 0	15

01 36 19 0	0
01 36 18 984	16
01 36 18 968	16
01 36 18 953	15
01 36 18 953	0
01 36 18 937	16
01 36 18 921	16
01 36 18 921	0
01 36 18 906	15
01 36 18 890	16
01 36 18 875	15
01 36 18 875	0
01 36 18 859	16
01 36 18 843	16
01 36 18 843	0
01 36 18 828	15
01 36 18 812	16
01 36 18 812	0
01 36 18 812	0
01 36 18 812	0
01 36 18 796	16
01 36 18 796	0
01 36 18 796	0
01 36 18 781	15
01 36 18 781	0
01 36 18 781	0
01 36 18 781	0
01 36 18 781	0
01 36 18 765	16
01 36 18 750	15
01 36 18 734	16
01 36 18 734	0
01 36 18 718	16
01 36 18 703	15
01 36 18 703	0
01 36 18 703	0
01 36 18 703	0
01 36 18 687	16
01 36 18 687	0
01 36 18 687	0
01 36 18 687	0
01 36 18 671	16
01 36 18 671	0
01 36 18 671	0
01 36 18 671	0
01 36 18 656	15
01 36 18 656	0
01 36 18 656	0
01 36 18 656	0
01 36 18 640	16
01 36 18 640	0
01 36 18 640	0
01 36 18 640	0
01 36 18 640	0
01 36 18 625	15
01 36 18 625	0
01 36 18 625	0
01 36 18 625	0
01 36 18 625	0
01 36 18 609	16
01 36 18 609	0
01 36 18 609	0
01 36 18 609	0
01 36 18 609	0
01 36 18 593	16
01 36 18 593	0
01 36 18 593	0
01 36 18 593	0
01 36 18 578	15
01 36 18 578	0
01 36 18 578	0
01 36 18 578	0
01 36 18 578	0
01 36 18 562	16

01 36 18 562	0
01 36 18 562	0
01 36 18 562	0
01 36 18 562	0
01 36 18 562	0
01 36 18 546	16
01 36 18 546	0
01 36 18 546	0
01 36 18 546	0
01 36 18 546	0
01 36 18 531	15
01 36 18 531	0
01 36 18 531	0
01 36 18 531	0
01 36 18 515	16
01 36 18 515	0
01 36 18 500	15
01 36 18 484	16
01 36 18 468	16
01 36 18 453	15
01 36 18 453	0
01 36 18 437	16
01 36 18 421	16
01 36 18 421	0
01 36 18 406	15
01 36 18 390	16
01 36 18 390	0
01 36 18 375	15
01 36 18 359	16
01 36 18 359	0
01 36 18 343	16
01 36 18 328	15
01 36 18 312	16
01 36 18 312	0
01 36 18 296	16
01 36 18 281	15
01 36 18 281	0
01 36 18 265	16
01 36 18 250	15
01 36 18 234	16
01 36 18 234	0
01 36 18 218	16
01 36 18 203	15
01 36 18 187	16
01 36 18 187	0
01 36 18 171	16
01 36 18 156	15
01 36 18 156	0
01 36 18 140	16
01 36 18 125	15
01 36 18 109	16
01 36 18 93	16
01 36 18 93	0
01 36 18 78	15
01 36 18 62	16
01 36 18 62	0
01 36 18 46	16
01 36 18 31	15
01 36 17 984	47
01 36 17 843	141
01 36 17 843	0
01 36 17 828	15
01 36 17 812	16
01 36 17 812	0
01 36 17 796	16
01 36 17 781	15
01 36 17 781	0
01 36 17 765	16
01 36 17 750	15
01 36 17 750	0
01 36 17 734	16
01 36 17 718	16
01 36 17 703	15
01 36 17 703	0
01 36 17 687	16
01 36 17 671	16

01 36 17 671	0
01 36 17 656	15
01 36 17 640	16
01 36 17 640	0
01 36 17 625	15
01 36 17 609	16
01 36 17 609	0
01 36 17 593	16
01 36 17 578	15
01 36 17 562	16
01 36 17 546	16
01 36 17 546	0
01 36 17 531	15
01 36 17 515	16
01 36 17 500	15
01 36 17 500	0
01 36 17 484	16
01 36 17 468	16
01 36 17 468	0
01 36 17 453	15
01 36 17 437	16
01 36 17 437	0
01 36 17 421	16
01 36 17 406	15
01 36 17 390	16
01 36 17 390	0
01 36 17 375	15
01 36 17 359	16
01 36 17 359	0
01 36 17 343	16
01 36 17 328	15
01 36 17 328	0
01 36 17 312	16
01 36 17 296	16
01 36 17 296	0
01 36 17 281	15
01 36 17 281	0
01 36 17 281	0
01 36 17 203	78
01 36 17 187	16
01 36 17 187	0
01 36 17 187	0
01 36 17 109	78
Tiempo promedio de respuesta (milisegundos)	8,85
Tiempo de testeo (Segundos)	7,9
Cantidad de eventos enviados	900



