

Universidad de las Ciencias Informáticas

Facultad 2



Diseño de Sistema de Monitoreo de SMDR.

Trabajo de Diploma para optar por el título de
Ingeniero Informático

Autores:

Yilian Monier Leyva
Karen Planas Peña

Tutores:

Ing. Yenlys Guerra Dávila
Ing. Ailín Alarcón Ferrá

Ciudad de la Habana, 30 de Junio de 2008
"Año 50 de la Revolución"

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los 30 días del mes de Junio del año 2008.

Yilian Monier Leyva

Firma del Autor

Karen Planas Peña

Firma del Autor

Ailín Alarcón Ferrá

Firma del Tutor

Yenlys Guerra Dávila

Firma del Tutor

AGRADECIMIENTOS

A nuestras tutoras Ailín y Yenlys, por su ayuda y apoyo.

A mi familia por darme fuerzas para continuar y llegar al final.

A mis compañeros de carrera, especialmente a aquellos que siempre estuvieron cerca y estarán por mucho que nos distancien.

Yilian

En primer lugar a mi mamá que es la responsable número uno de que yo haya llegado hasta aquí.

A Roberto por apoyarnos incondicionalmente.

A mi papá y a Milagros.

Al Dr. Carlos Roberto por llegar en tan buen momento y por ser como es, gracias.

A Ailin y Yenly por dedicarnos pedacitos de su tiempo y por confiar en nosotros.

A Yilian mi insuperable compañera de tesis, por ser la mejor amiga.

A todos mis amigos, los de allá y los de aquí.

A los que ya no están.

Karen

DEDICATORIA

A mi familia y amigos, mi fuente de inspiración.

A mis abuelitos, mis padres y mi hermana por todo su amor y cariño.

A los años que pasé lejos de mi familia y del único verdadero amor de mi vida, tú mi D.

Yilían

A mis padres.

Karen

RESUMEN

Las pizarras telefónicas son equipamiento muy útil para las grandes entidades. Evitan conectar todos los teléfonos de una oficina de manera separada a la red de telefonía local pública (RTC), evitando a su vez que estos tengan una línea propia para llamadas salientes y por lo tanto cargos mensuales hacia la central telefónica. Además con ellas es posible realizar llamadas más rápidas con discado abreviado, mayor completamiento de llamadas y mayor ahorro económico por concepto de tráfico telefónico. Existen diversos modelos y marcas de pizarras telefónicas y cada una genera de manera distinta un registro SMDR (Station Message Detail Recording) que contiene los datos de la llamada. Con la información contenida en estos registros SMDR se obtienen los costos de las llamadas y demás detalles importantes a informar al cliente.

El objetivo de este trabajo es diseñar un sistema de monitoreo de SMDR, un software estándar en el que se puedan configurar el tipo de pizarra, registros u otras opciones, y además que extraiga la información contenida en los SMDR y la almacene en una base de datos.

En el transcurso de esta investigación se exponen detalladamente los pasos que se siguieron para desarrollar el sistema. Incluyendo el origen y la historia de las pizarras telefónicas así como sus características, funciones y las posibilidades que brindan. Se presentan algunos de los modelos de pizarras con los que trabajará inicialmente el software y se describen los SMDR, registros que contienen los detalles de las llamadas que se han realizado a través de la pizarra. Además se realiza el análisis y el diseño del sistema, y se estudia la factibilidad del mismo utilizando el método de estimación por puntos de casos de uso. Por último se reflejan las recomendaciones que deben ser consideradas para investigaciones próximas.

TABLA DE CONTENIDOS

AGRADECIMIENTOS	I
DEDICATORIA.....	II
RESUMEN	III
INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	3
1.1 Introducción.....	3
1.2 Pizarras telefónicas o PBX.....	3
1.2.1 Tipos de llamadas que se realizan a través de un PBX.....	4
1.2.2 Servicios que brinda un PBX.	5
1.2.3 Conexión del PBX.....	6
1.2.4 PBX en la actualidad.....	7
1.2.5 Pizarras telefónicas más comercializadas en Cuba.....	7
1.3 SMDR (Station Message Detail Recording).....	11
1.3.1 Ejemplos de registros SMDR.....	11
1.4 Softwares de monitoreo de PBX.....	13
1.4.1 Software PBX Call Tarifficator Pro.....	13
1.4.2 Software CPanax.....	15
1.4.3 Software NeuroTel v1.0.....	15
1.5 Metodología de desarrollo de software.....	16
1.5.1 Proceso Unificado de Desarrollo de Software (RUP).....	16
1.6 Lenguaje Unificado de Modelado (UML).....	17
1.7 Propuesta de herramientas a utilizar en el desarrollo del sistema.....	18
1.7.1 Plataforma de desarrollo Java.....	18
1.7.2 Gestor de Base de Datos.....	21
1.7.3 Microsoft Office Vicio.....	23
1.7.4 Visual Paradigm.....	23
1.8 Conclusiones.....	24
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....	25
2.1 Introducción.....	25
2.2 Solución propuesta del Sistema de Monitoreo de SMDR.....	25

2.3 Modelo de dominio.....	26
2.3.1 Diagrama de clases del dominio.....	26
2.3.2 Descripción de los conceptos del dominio.....	26
2.3.3 Descripción del modelo de dominio.....	27
2.4 Requisitos.....	27
2.4.1 Requisitos funcionales.....	27
2.4.2 Requisitos no funcionales.....	29
2.5 Modelo de Casos de Uso del sistema.....	30
2.5.1 Definición de los actores del sistema.....	30
2.5.2 Diagrama de Casos de Uso del sistema.....	31
2.5.3 Descripción de los casos de uso del sistema.....	31
2.6 Conclusiones.....	43
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.....	44
3.1 Introducción.....	44
3.2 Modelo de análisis.....	44
3.2.1 Diagramas de clases del análisis.....	45
3.3 Modelo de diseño.....	48
3.3.1 Diagrama de clases del diseño.....	48
3.3.2 Diagramas de interacción.....	54
3.3.3 Patrones de diseño.....	54
Patrones de diseño.....	55
3.4 Diseño de la base de datos.....	57
3.4.1 Modelo lógico de datos.....	57
3.4.2 Modelo físico de datos del sistema.....	58
3.5 Diagrama de despliegue.....	59
3.6 Diagrama de componentes.....	60
3.7 Conclusiones.....	61
CAPÍTULO 4: ESTUDIO DE FACTIBILIDAD.....	62
4.1 Introducción.....	62
4.2 Método de estimación por Puntos de Casos de Uso.....	62
4.3 Planificación.....	62

4.3.1 Cálculo de Puntos de Casos de Uso sin ajustar	62
4.3.2 Cálculo de Puntos de Casos de Uso ajustados	63
4.3.3 Calcular esfuerzo del flujo de trabajo Implementación.	65
4.3.4 Calcular el esfuerzo total de todo el proyecto.	66
4.4 Cálculo del costo	66
4.5 Beneficios tangibles e intangibles.....	67
4.6 Análisis de costo y beneficios.....	67
4.7 Conclusiones.....	67
CONCLUSIONES.....	68
RECOMENDACIONES	69
BIBLIOGRAFÍA.....	70
ANEXOS.....	72
Anexo1. Diagrama de clases del diseño.	72
Anexo 2. Diagramas de Secuencia del diseño.....	73
GLOSARIO.....	80

INTRODUCCIÓN

El servicio de venta de pizarras telefónicas privadas es ofrecido por la Empresa de Telecomunicaciones de Cuba S.A. (ETECSA) desde el año 1994, coincidiendo prácticamente con la fundación de la empresa, y está orientado a las pequeñas, medianas y grandes entidades que necesitan distribución telefónica para una solución más integral de sus comunicaciones. De entonces a la fecha ETECSA ha ido avanzando desde el punto de vista tecnológico, y las soluciones que hoy en día ofertan se caracterizan por ser completamente digitales, incluso están brindando soluciones integrales de Voz, Datos e Inalámbricas.

Las pizarras telefónicas resultan muy útiles para aquellos clientes que tienen una movilidad grande dentro de extensas áreas de trabajo, ya sea en universidades, naves de almacenaje, fábricas, e incluso edificios altos.

Dentro de los distintos proveedores mundiales de pizarras telefónicas se encuentran: MITEL, ALCATEL, PANASONIC, ERICSSON, INFINITY, TOSHIBA, LG, entre otras. Aunque las cuatro primeras son las más comercializadas en Cuba actualmente.

En la mayoría de los casos, las instituciones que compran las pizarras telefónicas solicitan a ETECSA un software para monitorear y controlar toda la información relacionada con las llamadas que se realizan a través de estas, por ejemplo: hora en que se realizó la llamada, tiempo de duración, costo de la llamada, extensión a la que llamó, desde que número llamó, etc. Estos datos se obtienen a través de un registro SMDR (*Station Message Detail Recording*) que genera la pizarra telefónica en tiempo real. Los registros SMDR son enviados por la pizarra telefónica al sistema como cadenas de caracteres que posteriormente se traducen y se guardan en la base de datos. Cada tipo¹ de pizarra telefónica genera uno o varios formatos SMDR, que pueden coincidir en algunas pizarras. Se necesita un software que interprete el registro SMDR y lo traduzca en formato lógico para guardar en la base de datos. Comprar y trabajar con un software distinto para cada tipo de pizarra resultaría costoso y complejo.

Actualmente existen softwares que obtienen la información de los SMDR y la gestionan, pero son propietarios y ETECSA sólo puede servir como intermediario entre los proveedores y el cliente, para que se efectúe la compra de los mismos. Por esta razón ETECSA plantea la necesidad de crear un sistema estándar, capaz de interpretar y traducir el registro SMDR correspondiente a la pizarra en cuestión.

¹ El término se refiere a la marca y al modelo de la pizarra.

El presente trabajo surge como la necesidad de dar solución a la situación anteriormente expuesta por lo que el **problema científico** sería:

¿Cómo diseñar un sistema que permita el monitoreo de registros SMDR generados por las pizarras telefónicas?

Se definen como **objeto de estudio** los sistemas de monitoreo de registros SMDR que realicen funciones similares a la del software a diseñar. De ahí que el **campo de acción** comprenda los sistemas de monitoreo de registros SMDR más comercializados en Cuba.

El **objetivo general** de este trabajo es analizar y diseñar un sistema de monitoreo de registros SMDR.

Y los **objetivos específicos** son:

- Diseñar un sistema que posea amplios parámetros de configuración.
- Diseñar un sistema capaz de capturar e interpretar los registros SMDR.
- Proporcionar información acerca del estado del sistema.

En el transcurso de nuestro trabajo se plantearon las siguientes **preguntas de investigación**:

- ¿Qué conceptos, tecnologías y software han sido empleados en el desarrollo de sistemas de monitoreo de registro de pizarras telefónicas?
- ¿Qué metodología utilizar para llevar a cabo el desarrollo exitoso del sistema?
- ¿Qué herramientas y plataforma utilizar para desarrollar el sistema?

Para poder cumplir los objetivos e interrogantes anteriormente expuestos se proponen las siguientes **tareas**:

- Investigar características y funcionamiento de las pizarras más comercializadas en Cuba.
- Analizar sistemas similares de monitoreo de registros SMDR.
- Seleccionar una metodología para el análisis y el diseño del sistema.
- Seleccionar las herramientas óptimas para el desarrollo del sistema.

El trabajo está estructurado en 4 capítulos: El **Capítulo 1** contiene la **Fundamentación Teórica**, donde se hace un estudio de los distintos tipos de pizarras telefónicas, y la forma en que funcionan; se muestran además las diferentes herramientas y metodología a utilizar. En el **Capítulo 2** se definen las **Características del Sistema**, se propone una solución a través de una descripción detallada de un modelo de dominio, y los requisitos funcionales y no funcionales del sistema. Los principales aspectos relacionados con la construcción de la solución propuesta se abordan en el **Capítulo 3, Análisis y diseño del sistema**, donde se modelan los diagramas de clases del análisis y el diseño, y se especifican los principios de la implementación. Y por último en el **Capítulo 4: Estudio de factibilidad** se muestra el costo estimado del proyecto y un análisis de los beneficios que reporta.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se presenta la fundamentación teórica de la tesis. Se exponen conceptos importantes relacionados con el tema de las pizarras telefónicas, como el origen y evolución de los PBX y en qué consisten, centrándonos fundamentalmente en los SMDR que son la base de la investigación. Se caracterizan además, algunos de los tipos de pizarras telefónicas más comercializados en Cuba actualmente. Se describen los aspectos fundamentales del trabajo, así como las herramientas y la metodología a utilizar.

1.2 Pizarras telefónicas o PBX.

Un **PBX** ó **PABX** (*Private Branch Exchange* y *Private Automatic Branch Exchange*, respectivamente) cuya traducción al español sería *Intercambiador automático de redes privadas*. Es una central telefónica que pertenece a una empresa pero no a la compañía que se la vende. El término se refiere a equipos de comunicaciones telefónicas destinados a establecer y mantener llamadas tanto internas (llamadas entre extensiones) como externas (llamadas realizadas a las líneas de la red pública de teléfono).

En los orígenes de la telefonía era necesario conectar manualmente cables para establecer la comunicación. Este sistema era conocido como PMBX (PBX Manual) con el que se requería un operador, para que realizase las funciones de conmutado de llamadas. Luego fue reemplazado por un dispositivo electromecánico automático y posteriormente con el avance de la electrónica de microprocesadores, por sistemas digitales de conmutación que se le llamó PABX desplazando al PMBX. A partir de ese momento PABX y PBX se convirtieron en sinónimos.

Los PBX de gran escala, funcionan como un dispositivo físico que puede automatizar los procesos de tráfico de llamadas de una oficina, eliminando en algunos casos la necesidad de que el recepcionista atienda la totalidad de las llamadas entrantes, gracias a las contestadoras automáticas que interactúan con el llamante mediante el teclado del teléfono.

Un PBX requiere poco mantenimiento y tiene un promedio de 10 años de duración, para el cual se habría vuelto obsoleto, defectuoso, o simplemente la capacidad no daría abasto para el crecimiento de la compañía. Este último problema se ha solucionado con la capacidad de expansión que tienen los

PBX; es decir, se colocarían en ranuras destinadas para ello, tarjetas de expansión que contienen puertos con conectores telefónicos para aumentar el número de líneas troncales conectadas al PBX y/o más extensiones internas.

Un PBX mantiene tres funciones esenciales: establecer llamadas (internas o externas) entre dos o más usuarios, mantener la comunicación durante el tiempo que lo requiera el usuario y proveer información para contabilidad y/o facturación de llamadas.

Existen varios tipos de llamadas que se realizan a través de un PBX, las que se realizan entre sus propias extensiones y las que son recibidas por estas desde líneas de la Red de telefonía pública.

1.2.1 Tipos de llamadas que se realizan a través de un PBX.

Existen tres grupos de llamadas fundamentales:

Llamadas entrantes

Se tiende hoy en día a dejar configurado el acceso de llamadas entrantes al PBX por dos métodos principales:

- **Acceso por número único**

El usuario contrata n cantidad de líneas conectadas en los puertos de entrada del PBX. Generalmente, a esta n cantidad de líneas se les asocia un *único número* con el fin de evitar tener n números distintos. Por ejemplo, que una compañía tenga como número único el 555000. Al llamar desde el exterior a este número, podrá contestar una grabación de bienvenida indicando que marque el número de extensión; o por otro lado contestar directamente una operadora, quien transferirá la llamada a la extensión correspondiente tras la petición del que llama.

- **Acceso directo a extensiones**

Se puede contratar con la compañía una serie telefónica, de manera que cada extensión pueda recibir llamadas mediante un número directo. Por ejemplo, en una compañía todos los números pueden empezar con 555xxxx y los dígitos completados con x representarían la extensión marcada. Si se desea marcar la extensión 1234, uno marcaría desde fuera 5551234. No obstante, los usuarios con esta modalidad siempre tienen como número principal el de la operadora, entonces la otra forma de llamar podría ser marcar 5550000 y al contestar la locución de bienvenida, marcar a continuación 1234 o marcando directamente 5551234.

Llamadas salientes

Por lo general se marca un código de acceso, que es 9 con el estándar del continente americano, y 0 en Europa. Marcar este código abrirá y conmutará una línea externa al azar con el usuario desde cualquier extensión. Una vez marcado, se escuchará otro tono de marcado que corresponde a la línea externa o troncal. En los PBX con enlace digital no se abre ninguna línea, el PBX simula otro tono para posteriormente enviar la solicitud al procesador central de llamadas de la RTC.

Desde algunos teléfonos internos, se puede configurar la opción de línea directa, que tan solo al abrir la extensión, conmutará o dará tono de alguna línea externa, simulando esta extensión no estar conectada a la central.

Desde ciertos teléfonos, aparte de la posibilidad de marcar un código de acceso, se puede presionar una tecla con un indicador, que sirve para abrir una línea específica, por lo que no se marca 9 ó 0.

Algunos PBX más modernos, como los que trabajan con VoIP, no requieren de un código de acceso para hacer una llamada externa porque reconocen automáticamente cuando el usuario marca pocos dígitos para hacer una llamada interna, y si son más de éstos entenderá automáticamente que el número saldrá de la central por una línea externa.

Llamadas internas

Son llamadas gratuitas, ya que es la propia compañía la dueña de los dispositivos. El usuario marca directamente la extensión deseada sin pasar ésta por ninguna línea externa.

1.2.2 Servicios que brinda un PBX.

Existen los denominados servicios adicionales, la mayoría de ellos atribuibles también a cualquier central telefónica moderna:

- Marcado Automático.
- Contestador automático.
- Distribuidor automático de tráfico de llamadas.
- Servicio de directorio automatizado (usuarios pueden ser ruteados a la extensión deseada tecleando o diciendo verbalmente las iniciales o el nombre del empleado).

- Contestar llamadas desde otra extensión.
- Transferencia de llamadas.
- Llamada en espera. (permite escuchar y atender una llamada entrante, teniendo otra en línea).
- Aviso mediante timbre cuando una línea externa/extensión está libre.
- Conferencia entre tres o más usuarios (permite establecer comunicación entre varias líneas al mismo tiempo).
- Mensaje de Bienvenida.
- Marcación Abreviada.
- Marcado de una extensión desde el exterior del sistema.
- No-Molestar (DND).
- Sígame (programar desvío de llamadas de una extensión a otra).
- Música en espera.
- Contestador automático de buzón de voz.
- Anuncio por altavoces.

1.2.3 Conexión del PBX.

El método de conexión para pequeñas y medianas empresas es realizado a través de líneas comunes de la compañía telefónica, utilizando cuantas líneas quiera tener el usuario, y éstas a su vez conectadas al PBX.

Las nuevas tecnologías de telefonía IP, VoIP (*Voice Over Internet Protocol*) permiten la conmutación de voz vía Internet o redes informáticas privadas, siendo éste a veces el sistema de conexión de el PBX con la Red de Telefonía Local Pública (RTC).

Las extensiones del PBX suelen ser líneas sencillas conectadas a teléfonos simples, con características similares a una línea de la RTC en cuanto a tensión y señales eléctricas, por lo que son perfectamente compatibles. Otros teléfonos pueden ser exclusivamente para utilizarse con la marca del PBX, siendo compatibles sólo con estos.

Se puede incluso conectar una extensión de un PBX a un puerto línea externa de otro PBX, para lograr comunicación interna gratuita, pudiendo compartir llamadas internas y hasta líneas troncales, como si se tratara de una sola central.

1.2.4 PBX en la actualidad.

En el mundo del software libre se están desarrollando programas que realizan las funciones de una central PBX bajo Linux, tal es el caso del programa *Asterisk*, un software que instalado a dispositivos controladores de líneas telefónicas, como módems, es capaz de simular ciertas funciones de un PBX. Con estos sistemas es posible integrar esta y más funciones en un solo computador que brinda comunicación telefónica, Internet, fax, etc. Tanto como el fax, el módem, grupos de teléfonos, u otros dispositivos de comunicación pueden ser conectados a un PBX. Generalmente estos dispositivos se relacionan como extensiones.

Las compañías de teléfono pueden ofrecer por medio de sus líneas telefónicas el servicio simulado de PBX virtual, útil para empresas temporales o que se trasladan constantemente. La compañía telefónica que ofrece el servicio configura, mediante software, la central telefónica de la zona de tal forma que las líneas que lleguen, a la oficina que contrata el servicio, aparenten funcionar como un PBX con casi todas las mismas características. Cabe indicar que por cada extensión es un cableado por separado con la central regional de la compañía telefónica, por lo que es un poco ineficiente en caso de que sean muchas de ellas.

1.2.5 Pizarras telefónicas más comercializadas en Cuba.

A nivel mundial existen numerosos proveedores que ofrecen equipos para PBX: Agfeo, Avaya, Alcatel, Cisco, Ericsson, Fujitsu, InterTel, KARELLG, Mitel, Mitto, NEC, Nortel, Panasonic, Samsung, Siemens AG, ShoreTel, TELIP, Toshiba, Vertical, Vodavi, entre otros.

Hoy, las más comercializadas en Cuba por ETECSA son: ERICSSON, PANASONIC, MITEL y ALCATEL. Aunque no son las únicas que existen en el país, como es el caso de la pizarra telefónica INFINITY instalada en la Universidad de las Ciencias Informáticas. A continuación se muestra una serie de características generales de las pizarras anteriormente mencionadas.

Pizarras telefónicas ERICSSON

Ericsson Business Phone, la sencillez de la comunicación:

- Ideal para pequeñas y medianas empresas.
- Diseño escalable desde 8 hasta 250 extensiones.
- Telefonía IP para convergencia en las comunicaciones de voz y de datos.

- Funcionalidad de Call Center.
- Aplicaciones hospitalidad para soporte a hoteles, centros de conferencias y otras empresas con este tipo de requerimiento.
- Facilidades de movilidad para usuarios inalámbricos y personas que trabajan de forma remota.

Ericsson MX-ONE, diseñado para que su empresa siga avanzando:

- Ideal para medianas y grandes empresas.
- Más de 500 funciones y suficientes prestaciones para conectar hasta 100 000 usuarios.
- Sistema de comunicaciones, con gran escalabilidad, basado en un servidor.
- Basado en tecnología IP.
- Integración de las funciones de IP y voz brindando excelente movilidad y fiabilidad.

Pizarras telefónicas PANASONIC

Modelos **KX-TES824** y **KX-TEM824**:

- Son ideales para negocios pequeños u oficinas en casas que requieren de un sistema flexible con un alto grado de sofisticación.
- La **KX-TES824** acepta 3 líneas CO (truncos) y 8 extensiones. Y Con tarjetas opcionales puede fácilmente expandirse hasta 8 líneas CO y 24 extensiones.
- La **KX-TEM824** acepta 6 líneas CO y 16 extensiones. Y Con tarjetas opcionales se pueden fácilmente incrementar la capacidad del sistema hasta 8 líneas CO y 24 extensiones dependiendo del aumento de las necesidades.
- Se pueden conectar una gran variedad de equipos de telecomunicaciones como: teléfonos inalámbricos, máquinas contestadoras, módems, verificadores de tarjetas de créditos, máquinas de fax y cualquier otro equipo que trabaje con líneas de teléfonos convencionales.

Modelos **IP-PBX**, **KX-TDA100** y **KX-TDA200**:

- **KX-TDA100** con capacidad hasta 108 puertos, de los cuales 64 pueden ser líneas CO y hasta 64 teléfonos inalámbricos.
- **KX-TDA200** con capacidad hasta 216 puertos, de los cuales 128 pueden ser líneas CO y hasta 128 teléfonos inalámbricos.

- Sistema híbrido que integra funciones de PBX con la tecnología IP, resultando un sistema de comunicaciones de grandes prestaciones orientado a diferentes tipos de negocios que brinda avanzadas soluciones en telefonía y mensajería, comunicaciones flexibles y eficientes, movilidad inalámbrica, transmisión de voz en IP y la integración continua de sus computadoras mediante conexión directa a puertos USB.

Modelo KX-TDA600:

- Ideal para bancos, hospitales, Call Centers y hoteles.
- Capacidad de hasta 640 líneas CO ó 640 extensiones (688 puertos máximos). Puede llegar a una capacidad máxima de 1088 puertos.
- Capacidad de transmisión de voz utilizando protocolo IP (VoIP).
- Permite la mensajería unificada con la ayuda de correo de voz.

Pizarras telefónicas MITEL

Posibilita:

- Capacidad variable, puede configurar desde 96 hasta 1200 puertos.
- Compatibilidad para tecnología Red Digital de Servicios Integrados (RDSI).
- Sistema telefónico con prestaciones de avanzada y fácil de utilizar.
- Recepción automática de mensajes.
- Distribución automática de llamadas.
- Transferencia de llamadas.
- Llamadas en conferencia.
- Recuperación de llamadas en espera.
- Información visual de la línea que llama, estado de la llamada, hora y fecha, mensajes.
- Detección de tono de fax.

Además oferta la ejecución de proyectos de: servicios de instalación, asesoría técnica, adiestramiento de operadoras y servicio post venta.

Pizarras telefónicas ALCATEL

Las pizarras ALCATEL son originales de Francia. La ALCATEL modelo OXO (*Omni PCX Office*) posibilita:

- Capacidad para 236 extensiones con 84 puertos de datos.

- Servicio electrónico de comunicaciones.
- Incluyendo:

Parte de voz	Parte de datos	Parte de Internet
-Telefonía avanzada. - Mensajería de voz. - Operador automático. - Servidor CTI (Computer Telephone Integration). - VoIP. - Movilidad.	- LAN Switch. - Router y Firewall. - Servidor DHCP y DNS. - Información compartida.	- Servidor PROXY. - Servidor Caché. - Servidor e-mail. - Servidor ETHELNET.

Tabla 1.1 Funcionalidades de la pizarra ALCATEL

- La pizarra es modular y ampliable. Modular porque tiene todo incluido en una CPU única y ampliable porque se puede adaptar a las necesidades del cliente incluyéndole ranuras de expansión.

Pizarra telefónica INFINITY

INFINITY TX-2400L/XL

- Tiene capacidad para 3.480 puertos.
- Capacidad máxima de soporte de llamadas de 138 000 por segundo.
- Capacidad para 100 horas y 32 canales de mensajes de voz.
- Servicio de certificación de llamadas para todas las líneas.
- Servicio de espera de mensajes para todas la líneas.
- Servicio de despliegue de alarma de: sensor de temperatura y cierre termal.
- Servicio de administración, monitoreo y control.
- El sistema de la **TX-2400L** está disponible con un paquete de software estándar que es configurable para varias aplicaciones.
- IVM (Integrated Voice Mail) es una tarjeta que se utiliza en aplicaciones de voz, que consultan los mensajes de voz almacenados.
- Capacidad de grabación de voz en todas las consolas.
- Puede configurarse la grabación automática de voz de cada llamada.
- Respuesta interactiva de voz (IVR).
- Facilidad de mantenimiento remoto.

- Múltiple operación de la consola ISDN Operators.
- ACD (Automatic Call Distribution) ofrece una distribución uniforme de las llamadas entrantes para un grupo de estación de usuarios.

Por cada llamada realizada a través de estas pequeñas centrales telefónicas o PBX, es generado un registro llamado SMDR.

1.3 SMDR (Station Message Detail Recording).

Los SMDR son registros, generados por las pizarras telefónicas, que contienen detalles de las llamadas que se han realizado a través de ellas. Cada registro SMDR ocupa una línea de 85 caracteres (88 si incluye un identificador del sistema de 3 dígitos). En un registro de llamada puede incluirse información como:

Información	Descripción
Fecha	Fecha inicio de la llamada.
Tiempo	Hora inicio de la llamada.
Extension	Extensión que realiza la llamada.
Número Marcado	Número marcado para llamadas salientes.
Duración	Duración de la llamada.
Tronco	Número del tronco utilizado en la llamada.
Cuenta	Código de la cuenta insertado para la llamada.
CD	Tipo de llamada: (<i>TR</i> - transferida, <i>FW</i> - reenviada, <i>NA</i> - no contestada)
Dirección	Dirección de la llamada – Entrante o saliente.
Tiempo de timbrado	Tiempo de espera hasta concretar la llamada o desistir de esta.

Tabla 1.2 Información que contiene un SMDR.

1.3.1 Ejemplos de registros SMDR.

A continuación se muestran una serie de ejemplos de registros SMDR generados por distintos tipos de llamadas.

Llamada saliente desde una extensión a un número de la red de telefonía pública.

```

1      2      3      4      5      6      7      8      9
1234567890123456789012345678901234567890123456789012345678901234567890
-06/13 11:42 00:08:29 214 9 16135562122 ART054 000

```

Fig. 1.1 Ejemplo1 de SMDR, llamada saliente.

La llamada se realizó el 13 de Junio a las 11:42a.m. Se obtuvo el número de tronco 54 y se marcó el número 1-613-555-2122. La supervisión de respuesta fue dada. La conversación duró 8 minutos y 29 segundos.

Llamada saliente para acceder a una extensión de otro PBX.

```

1           2           3           4           5           6           7           8           9
12345678901234567890123456789012345678901234567890123456789012345678901234567890
05/17 10:51 00:01:52 213 9 201                                APR082                                000

```

Fig. 1.2 Ejemplo2 de SMDR, llamada saliente.

La llamada se realizó el 17 de mayo a las 10:51a.m. La extensión 213 accedió al grupo troncal, después marcó la extensión 201 para acceder a otro PBX. EL otro PBX dio la supervisión de respuesta y la conversación duró 1 minuto con 52 segundos. El número del tronco es 082.

Llamada saliente a través de operadora.

```

1           2           3           4           5           6           7           8           9
1234567890123456789012345678901234567890123456789012345678901234567890
+01/30 15:27 00:35:11 201 *9 1655666951                                ART052                                000

```

Fig. 1.3 Ejemplo3 de SMDR, llamada saliente.

El 30 de Enero, la extensión 201 marcó el número de la operadora y pidió llamar a una línea externa. La operadora marcó el número 1-654-555-6951. A las 3:27p.m. fue contestada la llamada. La conversación duró 35 minutos y 11 segundos. El tronco utilizado fue el 52.

Llamada entrante.

```

1           2           3           4           5           6           7           8           9
1234567890123456789012345678901234567890123456789012345678901234567890
01/30 15:10 00:02:22 T102 008 201                                201                                000

```

Fig. 1.4 Ejemplo1 de SMDR, llamada entrante.

La llamada entró el 30 de Enero a las 3:10p.m., a través del tronco número 102 hacia la extensión 201. Fue contestada después de 8 segundos de timbrado. La conversación duró 2 minutos y 22 segundos.

Llamada entrante a través de operadora.

	1	2	3	4	5	6	7	8	9
1234567890123456789012345678901234567890123456789012345678901234567890									
04/05 13:42 00:00:31 T090 009						1234			000

Fig. 1.5 Ejemplo2 de SMDR, llamada entrante.

El 5 de Abril a la 1:42p.m., el tronco 90 timbró a la operadora. Después de 9 segundos fue contestada la llamada en la extensión 1234. La conversación duró 31 segundos.

Para realizar la traducción y obtención de estos datos obtenidos en los SMDR, es necesario utilizar Softwares específicos encargados de monitorear estos registros.

1.4 Softwares de monitoreo de PBX.

En la actualidad existen varios casos de softwares que se encargan de realizar el monitoreo de los registros generados por las pequeñas centrales telefónicas. Estos en su mayoría presentan características y comportamientos similares al sistema que se propone diseñar en este trabajo. Todos tienen las mismas funcionalidades básicas, solo se diferencian en los servicios propios adicionales que cada uno pueda brindar. La diferencia fundamental entre ellos y el sistema propuesto radica en que los softwares que serán mostrados a continuación, son capaces de realizar, además de la traducción y el almacenamiento de los SMDR, la tarificación de llamadas.

1.4.1 Software PBX Call Tarifficator Pro.

PBX Call Tarifficator Pro es un software de gestión que ayuda a administrar el tráfico telefónico realizado desde y hacia la empresa que tiene la PBX. Puede: obtener información de todas las llamadas, locales, nacionales e internacionales, calcular el costo de las llamadas, crear, imprimir y exportar detallados reportes de los costos y cuadros de estadísticas. Consiste en dos módulos:

Logger Module

Este módulo sería similar al sistema que se propone diseñar y consiste en recibir continuamente la información de los SMDR del sistema PBX. Los registros válidos SMDR se convierten en formato de base de datos y se guardan en el disco. El módulo Logger no requiere interacción con un operador.

Muestra el estado de la conexión, el número de llamadas realizadas desde que empezó a correr el sistema, y los mensajes de errores.

Calls Browser

La ventana principal del Calls Browser muestra una lista de registros de llamadas. Cada registro puede contener información básica o ampliada. Se puede elegir qué columnas y en qué orden mostrar la información.

Columna	Descripción
Fecha	Fecha inicio de la llamada.
Tiempo	Hora inicio de la llamada.
Extension	Extensión que realizó la llamada.
Número Marcado	Número marcado para llamadas salientes.
Destino	País y/o ciudad para largas distancias y llamadas internacionales.
Duración	Duración de la llamada.
Costo	Calcula el costo de la llamada.
Empleado	Empleado al que se le debe cargar la llamada.
Departamento	Departamento al que pertenece el empleado.
Contacto	Nombre de contacto asignado al número de teléfono.
Tronco	Número del tronco utilizado en la llamada.
Grupo Troncal	Grupo al que pertenece dicho tronco.
Cuenta	Código de la cuenta insertado para la llamada.
CD	Tipo de llamada: (<i>TR</i> - transferida, <i>FW</i> - reenviada, <i>NA</i> - no contestada)
Dirección	Dirección de la llamada – Entrante o saliente.
Tipo	Tipo de llamada - Local, Larga distancia o Internacional
Tiempo de timbrado	Tiempo de espera hasta concretar la llamada o desistir de esta.
Proyecto	Nombre del proyecto asignado para el código de la cuenta de la llamada.

Tabla 1.3 Información obtenida por PBX Call Tarifficator Pro de un SMDR.

Las listas de llamadas pueden ser actualizadas automáticamente a medida que un nuevo registro de llamada llegue del sistema PBX. La lista de llamadas se puede ordenar y agrupar por cualquier columna, ascendente o descendente. Por ejemplo se puede agrupar las llamadas por empleados ordenadas por fecha.

1.4.2 Software CPanax.

CPANAX es un software diseñado para lograr un estricto control del gasto telefónico, y la optimización de los sistemas de comunicaciones en una empresa. Esta aplicación captura las llamadas realizadas y recibidas desde la central telefónica, almacenándolas en un computador, posteriormente, a través de un poderoso conjunto de reportes, ofrece a la gerencia toda la información necesaria para el análisis y distribución interna de gastos.

El módulo de análisis de tráfico realiza un estudio del tráfico telefónico generando información que le permite redistribuir las capacidades de su central telefónica, adicionalmente monitorea el uso de enlaces digitales (satélites y microondas) simulando el costo de utilización.

La información puede ser seleccionada por llamadas personales, teléfono destino, número más marcado, llamadas superiores a N minutos. Los diferentes criterios y parámetros no son mutuamente excluyentes, por lo tanto se pueden realizar las combinaciones de selección que el usuario desee, adaptando así la información generada a sus necesidades particulares. Muestra una a una las llamadas realizadas, incluyendo fecha, hora, teléfono marcado, extensión interna, costo de la llamada, ciudad destino, nombre del contacto, entre otros datos. Además tiene reportes resumidos y genera una gran variedad de gráficos que permite visualizar de una mejor forma la distribución del gasto y la proyección de los mismos.

1.4.3 Software NeuroTel v1.0

NeuroTel V1.0 puede capturar la información SMDR desde su Sistema PBX en cuanto la llamada se complete, analizar esta información, convertir a un formato de registro de base de datos y almacenar en el disco duro. Esto requiere una PC con un puerto serial disponible, el cual es conectado al puerto SMDR de su Sistema PBX. El software mantiene una base de datos con el registro de cada llamada, disponible para revisar, realizar consultas selectivas, imprimir reportes, exportar, etc. El software permite realizar consultas por Anexo, Numero Discado, Fecha, Inicio, Duración y Area. Cada consulta es creada por el usuario a través de una interfaz gráfica.

Linea	Anexo	Usuarios	Numero	DestinoOrigen	Fecha	Inicio	Duracion	tipo	Costo
CO6	21	Betsy Ramirez	004934850182	Alemania Berlin	03/01/2003	10:34:40	00:01:08	Saliente	2
CO6	20	Freddy Marquez	064818121	Junin	03/02/2003	10:45:00	00:02:00	Saliente	4
CO6	22	Julio M.V.	99374758		03/03/2003	18:16:41	00:03:29	Saliente	0,4
CO6	21	Betsy Ramirez	004934850182	Alemania Berlin	03/04/2003	10:34:40	00:01:08	Saliente	2
CO6	20	Freddy Marquez	064818121	Junin	03/05/2003	10:45:00	00:02:00	Saliente	0
CO6	21	Betsy Ramirez	004934850182	Alemania Berlin	02/06/2003	10:34:40	00:01:08	Saliente	0
CO6	22	Julio M.V.	99374758		03/06/2003	18:16:41	00:03:29	Saliente	0,4
CO5	22	Julio M.V.	99374758		04/06/2003	18:16:41	00:03:29	Saliente	0,4
CO6	20	Freddy Marquez	99374758		04/06/2003	18:16:41	00:03:29	Saliente	0
CO6	22	Julio M.V.	99374758		04/06/2003	18:16:41	00:03:29	Saliente	0
CO6	22	Julio M.V.	99384758		04/06/2003	18:16:41	00:03:29	Saliente	0,4
CO7	22	Julio M.V.	99374758		04/06/2003	18:16:41	00:03:29	Saliente	0,4
CO5	21	Betsy Ramirez	004934850182	Alemania Berlin	06/06/2003	10:34:40	00:01:08	Saliente	2
CO6	21	Betsy Ramirez	004934850182	Alemania Berlin	06/06/2003	10:34:40	00:01:08	Saliente	0
CO6	21	Betsy Ramirez	004934850182	Alemania Berlin	06/06/2003	10:34:40	00:09:08	Saliente	10
CO6	22	Julio M.V.	004934850182	Alemania Berlin	06/06/2003	10:34:40	00:01:08	Saliente	2
CO7	21	Betsy Ramirez	004934850182	Alemania Berlin	06/06/2003	10:34:40	00:01:08	Saliente	2
CO5	20	Freddy Marquez	064818121	Junin	06/06/2003	10:45:00	00:02:00	Saliente	4
CO6	20	Freddy Marquez	054818121	Arequipa	06/06/2003	10:45:00	00:02:00	Saliente	0
CO6	20	Freddy Marquez	084818121	Cusco	06/06/2003	10:45:00	00:02:00	Saliente	0
CO6	21	Betsy Ramirez	064818121	Junin	06/06/2003	10:45:00	00:02:00	Saliente	4
CO7	20	Freddy Marquez	064818121	Junin	06/06/2003	10:45:00	00:02:00	Saliente	4
CO6	21	Betsy Ramirez			07/06/2003	10:34:40	00:01:08	Entrante	0,4
CO6	21	Betsy Ramirez			07/06/2003	10:34:40	00:01:18	Entrante	0,4
CO6	21	Betsy Ramirez	004934850182	Alemania Berlin	07/06/2003	10:34:40	00:01:08	Saliente	0
CO6	20	Freddy Marquez			08/06/2003	10:45:00	00:02:00	Entrante	0,4
CO6	20	Freddy Marquez			08/06/2003	10:45:00	00:02:10	Entrante	0,4

Fig.1.6 Ventana Principal de Neurotel V1.0

1.5 Metodología de desarrollo de software.

En la Industria de Software, hay tendencia al crecimiento del volumen y complejidad de los productos, se quiere un software que esté mejor adaptado a nuestras necesidades y en el menor tiempo posible. Sin embargo, la mayoría de los desarrolladores hoy en día hacen sus sistemas con los mismos métodos de años atrás donde muchos proyectos fracasaban, sin percatarse que el elemento que hace que un software triunfe es tener un proceso bien definido y bien gestionado. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. Durante este proceso se sigue una metodología que no es más que un conjunto de procedimientos, técnicas, y herramientas que ayudarán a obtener el producto. Además se genera una documentación, que indicará a los desarrolladores del software qué es lo que hay que obtener en cada iteración.

1.5.1 Proceso Unificado de Desarrollo de Software (RUP).

Para elaborar este sistema se decidió utilizar *Rational Unified Process* (RUP). Una propuesta para el desarrollo de software basada en el desarrollo iterativo y el modelamiento visual. Usando UML como lenguaje de modelado para describir el sistema, lo cual permite incorporar al proceso de desarrollo de software un mejor control de los requerimientos. Posibilita además la distribución del trabajo en diversos frentes de forma simultánea. RUP proporciona el modo en que el equipo de proyecto puede

trabajar de una forma más conjunta con los clientes; lo que favorece a una mayor organización y entendimiento de lo que realmente el cliente necesita.

Características de RUP

- Dirigido por Casos de Uso.

Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. Los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de estos.

- Centrado en la arquitectura.

La arquitectura en un sistema software se describe mediante diferentes vistas del sistema en construcción. Este concepto incluye los aspectos estáticos y dinámicos más significativos del sistema. Esta se refleja en los casos de uso pues cada producto tiene tanto una función como una forma, ninguna es suficiente por sí sola.

- Iterativo e incremental.

Resulta práctico dividir el trabajo en partes más pequeñas o mini-proyectos, los cuales no son más que iteraciones que resultan en un incremento. Una iteración es una secuencia de actividades con un plan establecido y criterios de evaluación, cuyo resultado es una versión del software. Entre los beneficios de la iteración se encuentran: reducir el coste del riesgo al coste de un solo incremento, y lograr menos riesgo de sacar el producto al mercado en una fecha que acelere el ritmo de desarrollo. Las necesidades del usuario y correspondientes requisitos no puedan definirse completamente al principio, se requieren iteraciones sucesivas.

1.6 Lenguaje Unificado de Modelado (UML).

Lenguaje Unificado de Modelado (UML, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

UML es un "lenguaje" que se utiliza para definir un sistema de software, detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el *RUP*), pero no especifica en sí mismo qué metodología o proceso usar. UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas.

- Diagrama de Casos para Uso para modelar los procesos del negocio.
- Diagrama de Secuencia para modelar el paso de mensajes entre objetos.
- Diagrama de Colaboración para modelar interacciones entre objetos.
- Diagrama de Estado para modelar el comportamiento de los objetos en el sistema.
- Diagramas de Actividad para modelar el comportamiento de los Casos de Uso.
- Diagrama de Clases para modelar la estructura estática de las clases en el sistema.
- Diagrama de Objetos para modelar la estructura estática de los objetos en el negocio.
- Diagramas de Componentes para modelar componentes.
- Diagrama de Implementación para modelar la distribución del sistema.

Este lenguaje no es, por tanto, un método, sino varios. Se trata de una estandarización o consolidación de muchas notaciones y modelos usados anteriormente.

1.7 Propuesta de herramientas a utilizar en el desarrollo del sistema.

Para el desarrollo del sistema se realizó un estudio sobre las posibles herramientas a utilizar en su construcción. Teniendo en cuenta las tendencias actuales que existen en la universidad de migrar a software libre, la mayoría de las herramientas que seleccionadas pertenecen a software libre, así como la plataforma de desarrollo.

1.7.1 Plataforma de desarrollo Java.

Java es un entorno o plataforma de desarrollo creada por SunMicroSystems. Se ejecuta sobre otra plataforma hardware/software y provee:

- El lenguaje de programación Java.
- La Máquina Virtual Java (JVM).
- La Interfaz de Programación de Aplicaciones (API).

Esta plataforma Java se ha convertido en una de las más usadas por los desarrolladores. Tiene como principal ventaja que su entorno de desarrollo es independiente de la plataforma sobre la que se trabaje, es decir, sus aplicaciones son funcionales tanto en Linux, Unix, Solaris, Power/Mac como en Windows, brindando una compatibilidad a nivel de fuentes, código intermedio y bibliotecas de clases.

La implementación en dicha plataforma proporciona un compilador de Java para traducir código fuente al conjunto de instrucciones de la máquina virtual de Java, un intérprete para ejecutar las instrucciones de la máquina virtual y una implementación de la API de Java. Actualmente existen distintas ediciones o especificaciones de la plataforma Java como por ejemplo:

- J2ME (Java 2 Edición Micro): Orientada a entornos de limitados recursos, como teléfonos móviles, PDAs (Personal Digital Assistant), etc.
- J2EE (Java 2 Edición Empresarial): Orientada a entornos distribuidos empresariales o de Internet.
- J2SE (Java 2 Edición Estándar): Para entornos de gama media y estaciones de trabajo. Aquí se sitúa al usuario medio en una PC de escritorio.

1.7.1.1 Lenguaje de programación Java.

Java es un lenguaje de programación independiente de la plataforma, que alcanzó su madurez con la popularización de Internet y es en cierta manera el heredero legítimo de C++. Elimina la mayoría de sus complejidades como por ejemplo: la herencia múltiple y la creación de punteros. Java presenta características que lo convierten en un lenguaje seguro, estándar y de alto nivel. Algunas de las principales características se muestran a continuación:

Orientado a Objetos: Basado en C++ con algunas mejoras y elimina algunas cosas para mantener el objetivo de la simplicidad del lenguaje. Soporta las tres características propias de la orientación a objetos: encapsulamiento, herencia y polimorfismo.

Distribuido: Java se ha construido con extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como http y ftp. La característica de ser distribuido es debido a que proporciona las librerías y herramientas para que los programas puedan distribuirse, es decir, que se corran en varias máquinas, interactuando.

Interpretado: Se traduce el código fuente a un código intermedio (bytecode), que es interpretado por La Máquina Virtual de Java, lo cual permite que se pueda ejecutar en cualquier sistema operativo.

Robusto: Java realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos en Java ayuda a detectar errores, lo antes posible, en el ciclo de desarrollo.

Seguro: No se permite el acceso ilegal a memoria ya que no se trabaja con punteros.

Portabilidad: Una de las principales características de Java es su portabilidad. En lugar de compilarse a código nativo de la máquina, los programas de Java se traducen al formato bytecode que es el mismo en cualquier sistema operativo. Estos bytecodes son procesados directamente por La Máquina Virtual de Java que es dependiente de la máquina en uso. La portabilidad radica en que se pueden programar las aplicaciones solo una vez y ejecutar en cualquier plataforma. Por ejemplo, cuando se compila un programa Java en una plataforma Windows/Intel, se obtiene la misma salida compilada (o los mismos bytecodes) que en un sistema Macintosh o Unix.

Altas prestaciones: No se pierde tiempo optimizando código que no se ejecutará.

Multihilo: Permite la ejecución de varias tareas a la vez.

Dinámico: No conecta todos los módulos que comprende una aplicación hasta el tiempo de ejecución.

La mayoría de los lenguajes de programación se caracterizan por ser interpretados o compilados, lo que determina la manera en cómo serán ejecutados en una computadora. Java tiene la característica de ser al mismo tiempo compilado e interpretado. El compilador es el encargado de convertir el código fuente de un programa en bytecodes que son independientes de la plataforma en que se trabaje y que es ejecutado por el intérprete de Java que forma parte de la Máquina Virtual de Java. La programación en Java, permite el desarrollo de aplicaciones tanto bajo el esquema Cliente/Servidor, como de aplicaciones distribuidas, lo que lo hace capaz de conectar dos o más computadoras, ejecutando tareas simultáneamente, y de esta forma logra distribuir el trabajo a realizar.

1.7.2 Gestor de Base de Datos.

Existen múltiples situaciones en la vida cotidiana en las que de una manera u otra, se utilizan bases de datos con varias tablas para obtener información: en una biblioteca, en un banco, en cualquier empresa de servicios o a nivel personal. Existen diversas formas de manejar dichas bases de datos: con gestores como Oracle, SQL Sever, MySql, entre muchos otros.

Los sistemas de gestión de base de datos (SGBD) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Los gestores se componen de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. El propósito general de los sistemas de gestión de base de datos es manejar de manera clara, sencilla y ordenada un conjunto de información.

1.7.2.1 PostgreSQL

Como SGBD se propone utilizar PostgreSQL por las razones expuestas a continuación.

Es un proyecto *open source* (código abierto) lo que por definición significa que se puede obtener el código fuente, usar el programa, y modificarlo libremente, sin las limitaciones del software propietario, para adaptarlo a las necesidades particulares.

Postgre es un sistema de gestión de bases de datos Objeto-Relacionales **ORDBMS** (*Object-Relational Database Management System*).

Objeto-Relacional

Postgre aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transacciones, optimización de consultas, herencia, y arrays (arreglos).

Orientado a Objetos

Un ODBMS (*Object Database Management System*) representa la información mediante objetos y hace que estos aparezcan como objetos del lenguaje de programación al que dé soporte.

Una ventaja de utilizar un modelo de base de datos orientado a objetos es que proporcionan los costes de desarrollo más bajos y el mejor rendimiento gracias a que almacenan objetos y tienen una integración transparente con el programa escrito en un lenguaje de programación orientado a objetos.

Relacional

Una base de datos relacional RDBMS (*Relational Database Management System*) es un conjunto de una o más tablas estructuradas en registros (líneas) y campos (columnas), que se vinculan entre sí por un campo en común, en ambos casos posee las mismas características como por ejemplo el nombre de campo, tipo y longitud; a este campo generalmente se le denomina ID, identificador o clave.

Entre las ventajas de este modelo se encuentra que:

- Garantiza herramientas para evitar la duplicidad de registros, a través de campos claves o llaves.
- Garantiza la integridad referencial: Así al eliminar un registro elimina todos los registros relacionados dependientes.
- Favorece la normalización por ser más comprensible y aplicable.

El hecho de que PostgreSQL sea un gestor de base de datos orientado a objetos lo hace compatible con el lenguaje de programación escogido: Java.

Postgre además soporta **integridad referencial**, la cual es utilizada para garantizar la validez de los datos de la base de datos.

La flexibilidad del **API** de Postgre ha permitido proporcionar soporte al desarrollo fácilmente para el RDBMS PostgreSQL. Estas interfaces incluyen Object Pascal, Python, Perl, PHP, ODBC, Java/JDBC, Ruby, TCL, C/C++, y Pike.

MVCC (*Multi-Version Concurrency Control* - Control de Concurrency Multi-versión) es la tecnología que Postgre utiliza para evitar bloqueos innecesarios. MVCC posibilita que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos, ya que mantiene una ruta para todas las transacciones realizadas por los usuarios de la base de datos. Postgre es capaz entonces de manejar los registros sin necesidad de que estén disponibles. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases de datos, eliminando la necesidad del uso de bloqueos explícitos.

La característica de PostgreSQL conocida como **WAL** (*Write Ahead Logging*) incrementa la dependencia de la base de datos al registro de cambios antes de que estos sean escritos en la base de datos. Esto garantiza que en el hipotético caso de que la base de datos tenga fallos, existirá un registro de las transacciones a partir del cual podremos restaurar la base de datos. Esto puede ser enormemente beneficioso, ya que cualquier cambio que no fue escrito en la base de datos pueda ser recuperado usando el dato que fue previamente registrado. Una vez que el sistema ha quedado restaurado, un usuario puede continuar trabajando desde el punto en que lo dejó cuando falló la base de datos.

1.7.3 Microsoft Office Vicio

Microsoft Office Visio 2003 es una solución para la creación de gráficos y diagramas de aspecto profesional, que ayudan a comprender, documentar, transformar, analizar y transmitir información, además de diseñar sistemas y procesos complejos de una forma rápida y clara. Presenta un entorno amigable y flexible, que brinda facilidades como exportar diagramas y guardar en formato de página Web. Microsoft Office Visio 2003 brinda componentes que facilitan la elaboración de diagramas orientados al desarrollo de software, como es el caso del diseño de las interfaces de usuario del sistema.

1.7.4 Visual Paradigm

Visual Paradigm for UML es una herramienta CASE muy potente, de modelado visual UML, multiplataforma y fácil de utilizar. Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientado a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y documentación. Visual Paradigm además, importa a Rational Rose y porta una excelente interoperabilidad con otras herramientas CASE.

1.8 Conclusiones

En este capítulo se hizo un estudio sobre el origen y la evolución de las pizarras telefónicas. Se expusieron las características generales y las funcionalidades principales, de las pizarras más comercializadas en Cuba. Se describieron los SMDR y casos de softwares que muestran funciones similares a las del sistema que se diseñará.

Se hizo un análisis detallado de las herramientas a utilizar, definiendo a RUP como metodología de desarrollo y a UML como lenguaje de modelado. Se seleccionaron además el Visual Paradigm para desarrollar los diagramas UML y el Microsoft Office Visio para obtener los prototipos de interfaz de usuario del sistema. Se propone PostgreSQL como gestor de base da datos y Java como plataforma de desarrollo para la implementación del sistema.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción

En este capítulo se presenta la propuesta del sistema de monitoreo de registro de pizarras telefónicas (SMRPT). Se tratarán las características generales que fueron recogidas en el proceso de captura de requisitos funcionales y no funcionales. Se identifican además Actores y Casos de Uso (CU), obteniendo como resultado el Modelo de CU del sistema.

2.2 Solución propuesta del Sistema de Monitoreo de SMDR.

Se propone diseñar un sistema configurable en dependencia de la pizarra telefónica a la que se conecte, y los parámetros de esta. Después de establecida la configuración, el sistema deberá estar recibiendo constantemente los SMDR generados por la pizarra. Los registros SMDR válidos serán traducidos y almacenados en la base de datos. El sistema también debe ofrecer mensajes ante eventos no deseados, y mostrar el estado (error/sin error) en el que se está ejecutando la aplicación.

El sistema será diseñado utilizando el patrón de arquitectura *Layers* (capas) definido en cuatro capas: capa de presentación, capa Lógica, capa de acceso a datos y capa de datos; donde la capa inferior proporciona servicios a la superior. Las capas se tratarán de forma independiente, pues cada una encapsula un aspecto concreto del sistema. Este patrón es importante porque simplifica la comprensión y la organización del desarrollo del sistema. Para una mejor comprensión de la propuesta se muestra la siguiente figura.

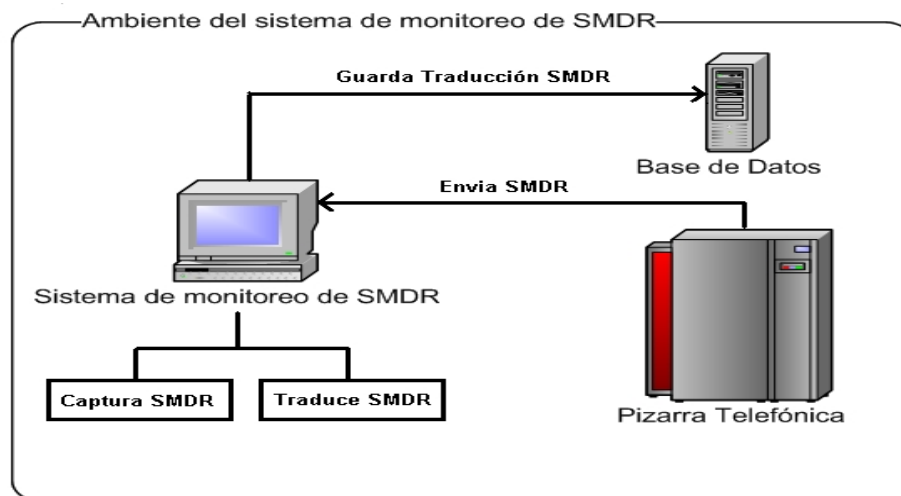


Fig.2.1 Ambiente del Sistema de Monitoreo de SMDR.

2.3 Modelo de dominio.

Después de haber realizado un análisis exhaustivo, se llegó a la conclusión de que no se modelaría el negocio a través del diagrama de casos de uso del negocio, debido a que en el sistema que se desea realizar no se identifican de forma clara los actores, ni los trabajadores, ni los procesos del negocio. Por lo que se decide realizar un modelo de dominio el cual es un subconjunto del modelo de objeto del negocio donde se representan los conceptos y eventos fundamentales que se expresan como clases con la cardinalidad que existe entre ellas. En el siguiente modelo de dominio se especifican las relaciones que existen entre los principales conceptos que interactúan en el sistema.

2.3.1 Diagrama de clases del dominio.

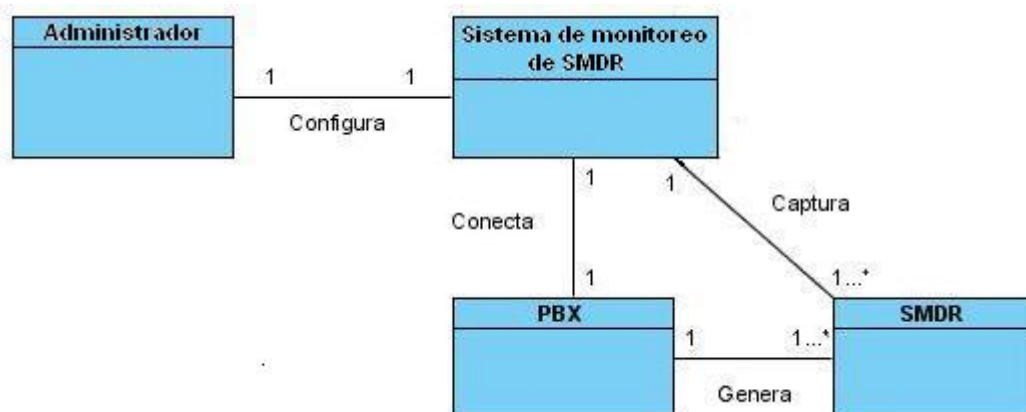


Fig.2.21 Modelo de dominio del Sistema de Monitoreo de SMDR.

2.3.2 Descripción de los conceptos del dominio.

A continuación se muestra una descripción de los conceptos representados en el diagrama del Modelo de dominio, con el objetivo de comprender cada una de las relaciones que se establecen entre estas entidades.

Administrador (Administrador de la pizarra telefónica): Es la persona autorizada a trabajar con el sistema, el encargado además de instalarlo y configurarlo.

PBX (Private Branch Exchange): Es una central telefónica privada que permite establecer y mantener llamadas tanto internas (llamadas entre extensiones) como externas (llamadas realizadas entre las extensiones y las líneas de la red pública de teléfono).

SMDR (Station Message Detail Recording): Registros generados por las pizarras telefónicas, que almacenan los diferentes parámetros de las llamadas que se realizan a través de estas.

Sistema de monitoreo de SMDR: Sistema encargado del monitoreo de los registros generados por las pizarras telefónicas, instalado en una computadora.

2.3.3 Descripción del modelo de dominio.

El administrador de la pizarra telefónica, única persona autorizada a interactuar con el sistema, se encarga de instalar y configurar el Sistema de monitoreo de registros SMDR de pizarras telefónicas. La aplicación será instalada en una computadora conectada a la pizarra. Esta genera los SMDR que contienen la información de las llamadas realizadas y el sistema captura estos registros para su posterior traducción y almacenamiento en la base de datos.

2.4 Requisitos.

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Ellos no alteran la funcionalidad del producto, esto quiere decir que los requerimientos funcionales se mantienen invariables sin importar con qué propiedades o cualidades se relacionen.

2.4.1 Requisitos funcionales.

RF1- Configurar sistema

- RF1.1- Seleccionar modelo de PBX.
- RF1.2- Seleccionar formato de SMDR.
- RF1.3- Seleccionar formato de fecha/hora.
- RF1.4- Definir parámetros del puerto serie.

RF2- Mostrar estado

- RF2.1- Mostrar ícono de estado.
- RF2.2- Reconocer el tipo de error.
- RF2.3- Mostrar mensaje de error producido.

RF3- Gestionar SMDR

- RF3.1- Capturar SMDR.
- RF3.2- Comprobar que el SMDR coincida con la configuración de registros.

RF3.3- Traducir SMDR.

RF3.4- Almacenar SMDR traducido.

RF4- Mostrar información resumen

RF4.1- Mostrar fecha/hora de lanzamiento de la aplicación.

RF4.2- Mostrar fecha/hora de la última llamada registrada.

RF4.3- Mostrar total de llamadas registradas desde el lanzamiento de la aplicación.

RF4.4- Mostrar número de errores de SMDR inválidos recibidos.

RF4.5- Mostrar último error que se produjo.

RF5- Configurar registros

RF5.1- Seleccionar el tipo de llamadas (salida y/o entrada) a registrar.

RF5.2- Seleccionar omitir o no llamadas de duración cero.

RF5.3- Seleccionar omitir o no llamadas no contestadas (si la PBX lo permite).

RF5.4- Seleccionar la ruta destino de la base de datos (BD).

RF5.4.1- Almacenar en la BD la información correspondiente a un año.

RF5.4.2- Nombrar fichero de la BD por año.

RF5.4.3- Generar automáticamente nueva BD al iniciarse un nuevo año.

RF5.5- Seleccionar salvar o no el registro SMDR en ficheros textos.

RF5.5.1- Seleccionar ruta destino para almacenar los ficheros.

RF5.5.2- Nombrar el fichero con el siguiente formato: AAAA_MM.TXT.

RF6- Administrar aplicación

RF6.1- Seleccionar ejecutar o no la aplicación al iniciar la PC.

RF6.2- Seleccionar establecer o no una llave de acceso.

RF7- Mostrar ventana Terminal

RF7.1- Mostrar datos recibidos desde la PBX.

RF7.1.1- Salvar datos del buffer hacia fichero TXT.

RF7.1.2- Limpiar pantalla.

RF8-Autenticar

RF9- Mostrar total de llamadas

RF9.1- Mostrar número de llamadas registradas hasta el momento.

2.4.2 Requisitos no funcionales.

Los requerimientos no funcionales son las propiedades o cualidades que el producto debe tener para que sea atractivo, usable, rápido y confiable.

- **Requisitos de usabilidad**

El administrador del sistema, único usuario de la aplicación, necesitará una preparación previa para operar la misma. Se requiere que posea un nivel medio o alto en conocimientos de computación, aunque el manejo de la aplicación es sencillo, pues la mayoría de las operaciones se realizan de forma automática. Las operaciones no automatizadas son de configuración, y visualización de reportes, las que son fácilmente comprensibles por cualquier usuario.

- **Requisitos de fiabilidad**

Es necesaria la ejecución constante de la aplicación para la actualización de la base de datos en tiempo real. Esta característica es fundamental debido a que continuamente se obtienen datos importantes en los registros SMDR generados por la pizarra. El sistema detecta y muestra los diferentes tipos de errores críticos que puedan surgir durante su ejecución.

- **Requisitos de eficiencia**

La eficiencia del producto estará determinada en gran medida por la velocidad que se logre en la transferencia de datos entre la PBX y la computadora en la que sea instalado el sistema.

Es necesario que la velocidad del puerto esté acorde con el volumen de información y la rapidez con que es generado por la pizarra, para lograr un mejor rendimiento del sistema. Esta velocidad varía en dependencia de las características de la pizarra telefónica.

- **Requisitos de diseño e implementación.**

Se desea construir un sistema para la plataforma Windows, por lo que fue necesario escoger un lenguaje de programación y un sistema gestor de base de datos (SGBD) que fueran compatibles con esta plataforma. Se recomienda utilizar PostgreSQL por ser considerado el gestor de base de datos de código abierto más avanzado del mundo. Y Java para el desarrollo de la aplicación debido a que es un lenguaje seguro, estándar y de alto nivel. El análisis y diseño se realizará usando UML y como apoyo a este lenguaje de modelación se utilizará la herramienta Visual Paradigm perteneciente a software libre.

- **Requisitos de interfaz externa.**

La aplicación propuesta poseerá una interfaz sencilla dirigida directamente al usuario del sistema. El diseño se realizará siguiendo las formalidades de las ventanas de Windows para mayor comodidad. Este software no intercambiará ningún tipo de información con un sistema mayor. Es un producto independiente, autónomo, que realizará sus propias funciones.

- **Requisitos de software y hardware.**

Se requiere que la computadora conectada a la pizarra telefónica permita la comunicación utilizando el protocolo de transferencia TCP/IP, y tenga al menos un puerto serie RS-232 disponible para establecer dicha comunicación directamente con el puerto SMDR de la pizarra; este puerto se encontrará todo el tiempo en modo de recepción de datos mientras se ejecute la aplicación.

2.5 Modelo de Casos de Uso del sistema.

El modelo de Casos de Uso del sistema representa las funcionalidades deseadas y el entorno del sistema a través de actores y casos de uso, sirve como un contrato entre clientes y desarrolladores, y sienta las bases necesarias para el desarrollo del análisis y el diseño del sistema.

2.5.1 Definición de los actores del sistema.

Los actores del sistema son terceros fuera del sistema que interactúan con él.

Actor	Justificación
AdministradorPT	Es el encargado de instalar y configurar la aplicación, así como velar por su correcto funcionamiento.
Pizarra telefónica	Es la encargada de generar los datos con los que trabajará la aplicación.

Tabla 2.1 Descripción de los actores del sistema.

2.5.2 Diagrama de Casos de Uso del sistema.

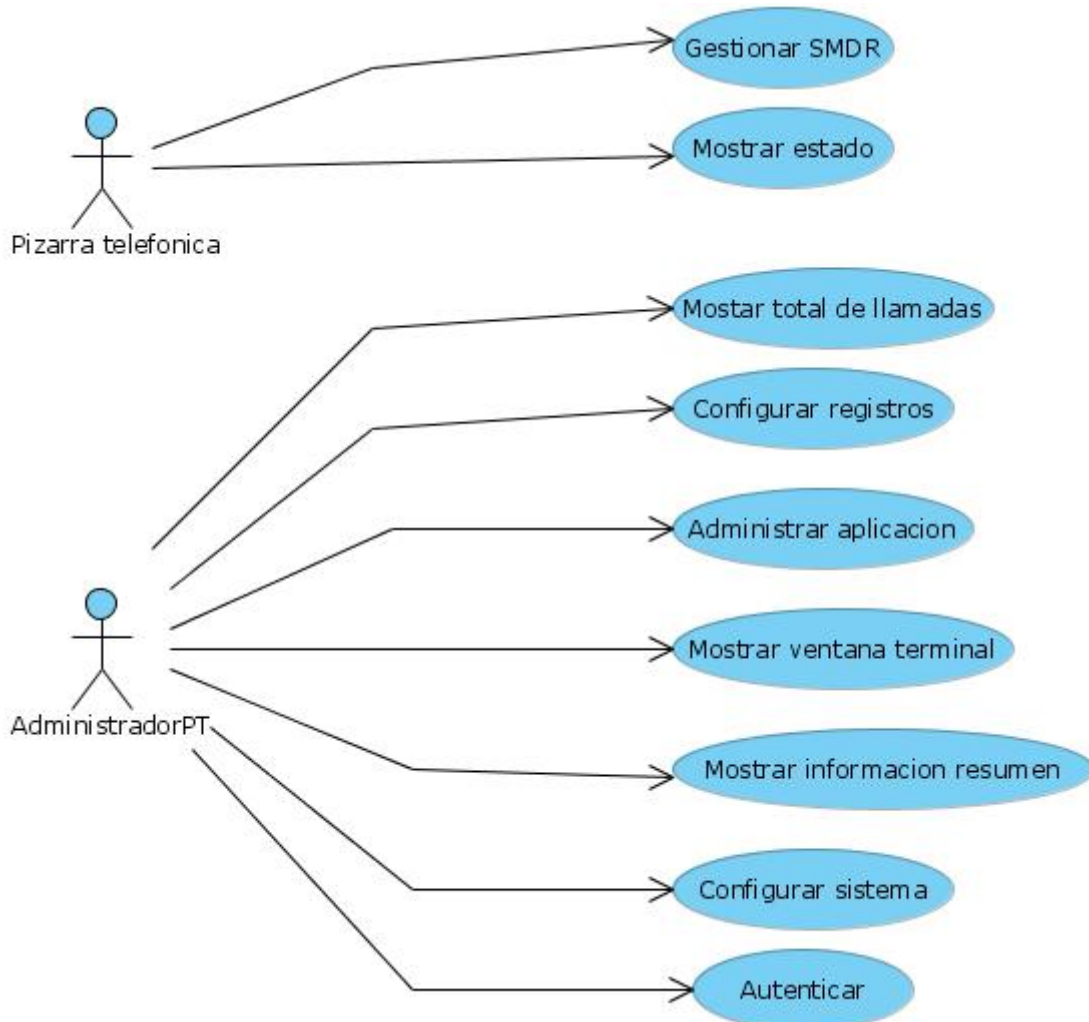


Fig.2.32 Diagrama de Casos de Uso del sistema.

2.5.3 Descripción de los casos de uso del sistema.

Los casos de uso son artefactos que describen, en forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario, las siguientes tablas presentan los casos de uso determinados para satisfacer los requisitos funcionales de sistema.

Tabla 2.2 Descripción del caso de uso, Configurar sistema.

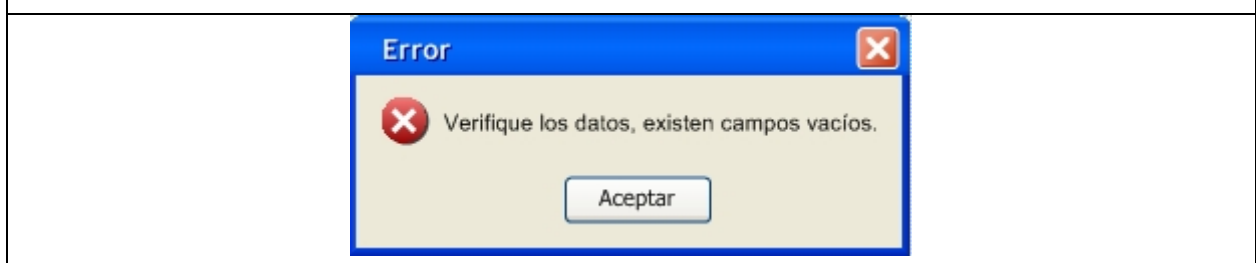
Caso de Uso:	Configurar sistema	
Actores:	AdministradorPT	
Resumen:	Cuando el AdministradorPT seleccione la opción "Configuración del sistema" se le mostrarán las opciones de configuración. Seleccionará el tipo de PBX que se ha conectado al sistema, se cargarán los posibles formatos SMDR que esta PBX genera. Seleccionará el formato SMDR, así como el formato de fecha/hora, que desee. Definirá los parámetros del puerto serie que se precisen.	
Precondiciones:	Conocer el tipo de PBX que se va a conectar al sistema. Que el Administrador del sistema esté autenticado.	
Referencias	RF1	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. Selecciona la opción "Configuración del sistema".	2. Carga la lista de PBX contenida en la base de datos.	
	3. Muestra una ventana con las opciones siguientes: Tipo de PBX. Formato de SMDR. Formato fecha/hora. Parámetros de puerto serie.	
4. Selecciona el tipo de PBX.	5. Carga de la BD los posibles formatos SMDR que esta PBX genera.	
	6. Muestra los formatos SMDR.	
7. Selecciona el formato SMDR.		
8. Selecciona el formato de fecha/hora.		
9. Selecciona los parámetros del puerto serie.		
10. Hace clic en el botón "Aceptar".	11. Verifica que no exista ningún campo vacío.	
	12. Guarda los cambios y cierra la ventana.	
Prototipo de Interfaz		



Flujo Alternativo 11a: Si existen campos vacíos.

Acción del Actor	Respuesta del Sistema
	11a.1 Indica que existen campos vacíos.
11a.2 Selecciona la opción "Aceptar".	11a.3 Cierra la ventana. Ir a la acción 2.

Prototipo de interfaz



Flujo Alternativo *a: Cancelar

Acción del Actor	Respuesta del Sistema
a.1 Da clic en el botón "Cancelar"	a.2 Cierra la ventana sin guardar los cambios.

Poscondiciones | Se actualiza la configuración de la pizarra telefónica en la BD.

Tabla 2.3 Descripción del caso de uso, Mostrar estado.

Caso de Uso:	Mostrar estado
Actores:	Pizarra telefónica



Resumen:	El caso de uso se inicia cuando la pizarra comienza a enviar información al sistema. La aplicación se ejecutará continuamente y su estado se mostrará a través de un icono en el tray-bar que será verde mientras no exista error. Al surgir un error el icono cambia a rojo y muestra el mensaje del error producido.	
Precondiciones:	Que la pizarra telefónica esté conectada al sistema.	
Referencias	RF2	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. La pizarra telefónica envía información al sistema.	2. Comprueba que no existan errores.	
	3. Muestra el icono de estado en el tray-bar, de color verde.	
Prototipo de Interfaz		
		
Flujos Alternos 2a: Error generado		
Acción del Actor	Respuesta del Sistema	
	2a.1 El color del icono cambia a rojo.	
	2a.2 Reconoce el tipo de error que se produjo.	
	2a.3 Muestra el mensaje de error en dependencia del tipo de error. Los posibles mensajes son: <ul style="list-style-type: none"> • Modelo de PBX seleccionado no corresponde con el modelo de PBX conectado al sistema. • Formato SMDR seleccionado no coincide con ninguno de los formatos que genera la PBX. • Formato de fecha/hora incorrecto. • Información recibida con errores. • Fallo en la conexión. Ir a la acción 2.	
Prototipo de interfaz		
		
Poscondiciones	Se muestra el estado de la aplicación.	

Tabla 2.4 Descripción del caso de uso, Gestionar SMDR.

Caso de Uso:	Gestionar SMDR	
Actores:	Pizarra telefónica	
Resumen:	El caso de uso se inicia cuando la pizarra comienza a enviar los SMDR al sistema. La aplicación captura los SMDR y los traduce, para posteriormente almacenarlos en la BD.	
Precondiciones:	Que la pizarra telefónica esté conectada al sistema.	
Referencias	RF3	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
1. La pizarra telefónica envía información al sistema.		2. Captura SMDR generado por la pizarra telefónica.
		3. Comprueba que el SMDR coincida con la configuración de registros.
		4. Traduce el SMDR.
		5. Almacena dicha traducción en la BD distribuida en los siguientes datos: <ul style="list-style-type: none"> • Fecha y hora inicio de la llamada. • Teléfono origen. • Teléfono destino. • Duración de la llamada. • Dirección (llamada entrante/saliente).
Prototipo de Interfaz: No aplica.		
Flujos Alternos 3a: No coincide SMDR con la configuración de registros.		
Acción del Actor		Respuesta del Sistema
		3a.1 Ir a la acción 2.
Poscondiciones	No aplica	

Tabla 2.5 Descripción del caso de uso, Mostrar información resumen.

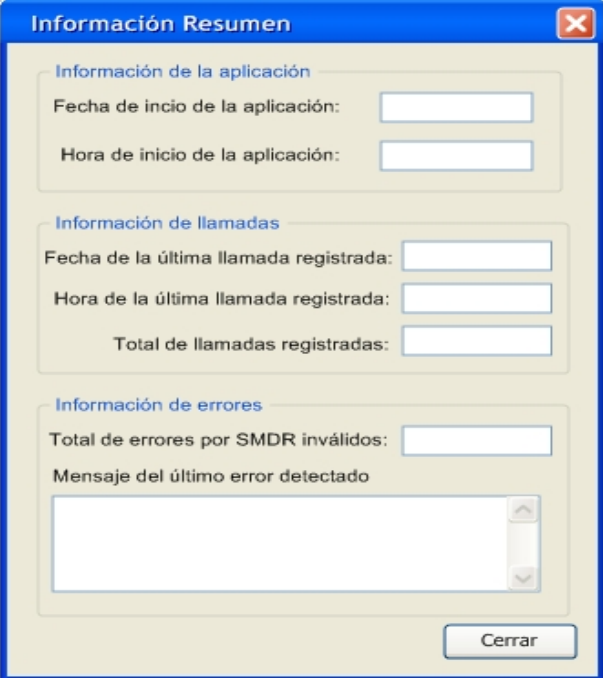
Caso de Uso:	Mostrar información resumen	
Actores:	AdministradorPT	
Resumen:	Cuando el AdministradorPT seleccione la opción "Información resumen" se mostrará una ventana con el resumen de informaciones generales del sistema.	
Precondiciones:	Que el Administrador del sistema esté autenticado.	
Referencias	RF4	
Prioridad	Secundario	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. Selecciona la opción "Información Resumen".	2. Obtiene de la BD la siguiente información: <ul style="list-style-type: none"> • Fecha/hora de inicio de la aplicación. • Fecha/hora de la última llamada registrada. • Total de llamadas registradas desde que inició la aplicación. • Total de errores por SMDR inválidos recibidos. • Mensaje del último error detectado. 	
	3. Muestra la ventana Información Resumen con los datos obtenidos de la BD.	
4. Hace clic en el botón "Cerrar".	5. Cierra la ventana.	
Prototipo de interfaz		
		
Poscondiciones	No aplica	

Tabla 2.6 Descripción del caso de uso, Configurar registros.

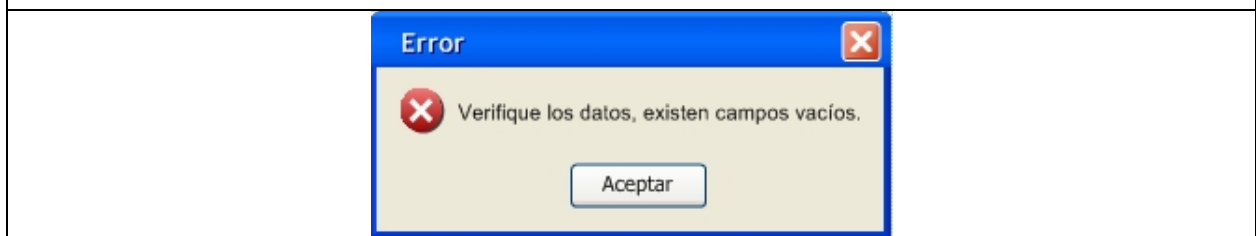
Caso de Uso:	Configurar registros.	
Actores:	AdministradorPT	
Resumen:	Cuando el AdministradorPT seleccione la opción "Configurar registros" se mostrará una ventana con las opciones para configurar la información que se desea recibir de la PBX, así como la opción para guardar los registros SMDR en ficheros TXT y hacer una salva de la información de la BD correspondiente al período de un año.	
Precondiciones:	Que el Administrador del sistema esté autenticado.	
Referencias	RF5	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. Selecciona la opción "Configurar registros".	2. Muestra una ventana con las opciones siguientes: <ul style="list-style-type: none"> • Tipo de llamadas a registrar (entrante/saliente). • Omitir llamadas de duración cero. • Omitir llamadas no contestadas. • Escoger la ruta destino de la BD. • Salvar registro SMDR a ficheros textos. • Escoger la ruta destino para estos ficheros. 	
3. Selecciona el tipo de llamada a registrar.		
4. Selecciona si desea omitir llamadas de duración cero.		
5. Selecciona si desea omitir llamadas no contestadas.		
6. Escoge la ruta destino donde se guardará la BD pasado un año.		
7. Selecciona si desea salvar los SMDR	8. Habilita la opción para que marque la ruta destino del SMDR.	
9. Selecciona la ruta destino donde se guardarán los SMDR por mes.		
10. Hace clic en el botón "Aceptar".	11. Verifica que no existan campos vacíos.	
	12. Guarda los cambios y cierra la ventana.	
Prototipo de interfaz		



Flujo Alternativo 11a: Existen campos vacíos.

Respuesta del Negocio	Respuesta del Sistema
	11a.1 Indica que existen campos vacíos.
11a.2 Selecciona la opción "Aceptar".	11a.3 Cierra la ventana. Ir a la acción 2.

Prototipo de interfaz




Flujo Alternativo *a: Cancelar

Acción del Actor	Respuesta del Sistema
a.1 Da clic en el botón "Cancelar"	a.2 Cierra la ventana sin guardar los cambios.

Poscondiciones	Se actualiza la configuración de registros en la BD.
-----------------------	--

Tabla 2.7 Descripción del caso de uso, Administrar aplicación.

Caso de Uso:	Administrar aplicación
Actores:	AdministradorPT
Resumen:	Cuando el AdministradorPT seleccione la opción "Administración" se mostrará una

	ventana donde podrá establecer una clave de acceso, y que el sistema se ejecute al inicio de sesión.	
Precondiciones:	Que el Administrador del sistema esté autenticado.	
Referencias	RF6	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. Selecciona la opción "Administración".	2. Muestra una ventana con las opciones siguientes: <ul style="list-style-type: none"> • Ejecutar la aplicación al iniciar la PC. • Establecer llave de acceso. 	
3. Selecciona si desea la opción de lanzar la aplicación al iniciarse la PC.		
4. Selecciona si desea, establecer contraseña.	5. El sistema habilita la opción que permite insertar una nueva clave de acceso.	
6. Inserta y confirma la clave.	7. Verifica que coincidan las claves.	
8. Hace clic en el botón "Aceptar".	9. Guarda los cambios y cierra la ventana.	
Prototipo de interfaz		
		
Flujo Alternativo 4a: No selecciona establecer contraseña.		
Acción del Actor	Respuesta del Sistema	
	4a.1 El sistema mantiene deshabilitada la opción que permite insertar una clave de acceso. Ir a la acción 8.	
Flujo Alternativo 7a: No coinciden las claves.		
Acción del Actor	Respuesta del Sistema	
	7a.1 Indica que las claves no coinciden.	

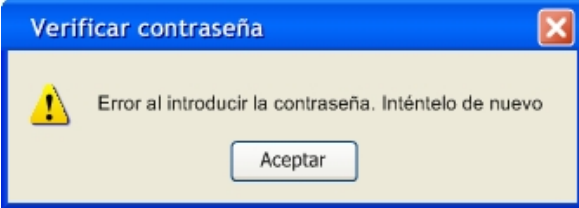
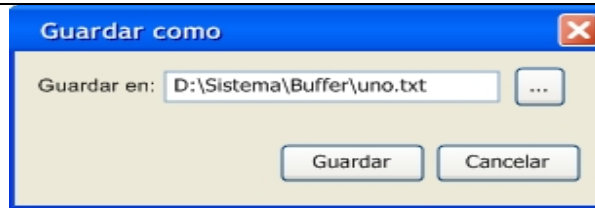
7a.2 Selecciona la opción "Aceptar".	7a.3 Cierra la ventana. Ir a la acción 5.
Prototipo de interfaz	
	
Flujo Alternativo *a: Cancelar	
Acción del Actor	Respuesta del Sistema
a.1 Da clic en el botón "Cancelar"	a.2 Cierra la ventana sin guardar los cambios.
Poscondiciones	Se actualizan los cambios en la BD.

Tabla 2.8 Descripción del caso de uso, Mostrar ventana Terminal.

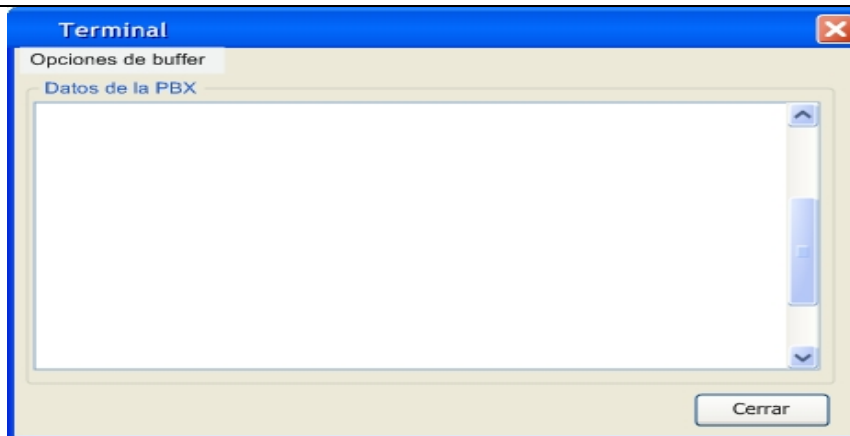
Caso de Uso:	Mostrar ventana Terminal
Actores:	AdministradorPT
Resumen:	Cuando el AdministradorPT seleccione la opción "Terminal" se mostrará una ventana con los datos crudos que se van recibiendo de la PBX y se van almacenando en un buffer temporal. Tendrá las opciones de limpiar pantalla y salvar buffer en un fichero TXT.
Precondiciones:	Que el Administrador del sistema esté autenticado.
Referencias	RF7
Prioridad	Secundario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Selecciona la opción "Terminal".	2. Muestra una ventana con los datos crudos que se van recibiendo de la PBX.
	3. Almacena estos datos en un buffer temporalmente.
4. Selecciona una de las Opciones de buffer: • Limpiar pantalla. • Salvar buffer.	5. Comprueba el tipo de selección que hizo el usuario. • Si es Limpiar pantalla ir a la sección "Limpiar pantalla". • Si es Salvar buffer ir a la sección "Salvar buffer".
6. Hace clic en el botón "Cerrar".	7. Cierra la ventana.
Sección: Limpiar pantalla.	
	8. Borra los datos guardados en el buffer hasta el momento. Limpia la pantalla.
Sección: Salvar buffer.	

	9. Muestra una ventana solicitando el destino para guardar el buffer en un fichero TXT.
10. Selecciona el destino y da clic en el botón "Guardar".	11. Verifica si el campo de la dirección está vacío.
	12. Guarda el fichero y cierra la ventana.

Prototipo de Interfaz. Sección: Salvar buffer



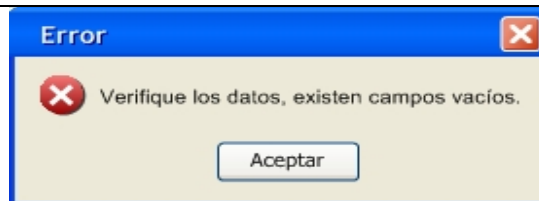
Prototipo de Interfaz: Mostrar ventana Terminal



Flujo Alterno 11a. Existen campos vacíos.

Acción del actor	Respuesta del Sistema
	11a.1 Indica que existen campos vacíos.
11a.2 Selecciona la opción "Aceptar".	11a.3 Cierra la ventana. Ir a la acción 10.

Prototipo de interfaz



Flujo Alterno *a: Cancelar

Acción del Actor	Respuesta del Sistema
a.1 Da clic en el botón "Cancelar"	a.2 Cierra la ventana sin guardar los cambios.

Poscondiciones	No aplica.
-----------------------	------------

Tabla 2.9 Descripción del caso de uso, Autenticar.



Caso de Uso:	Autenticar	
Actores:	AdministradorPT	
Resumen:	El caso de uso se inicia cuando el administrador solicita acceso a las opciones del sistema haciendo clic en el icono de la aplicación. El sistema pide insertar contraseña y verifica que sea válida para conceder el permiso.	
Precondiciones:	Que se haya establecido una contraseña.	
Referencias	RF8	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El administrador desea acceder al sistema y hace clic en el icono de estado de la aplicación.	2. Muestra una ventana de autenticación.	
3. Introduce contraseña y da clic en el botón Aceptar.	4. Comprueba que la contraseña sea válida.	
	5. Da acceso al sistema.	
Prototipo de Interfaz		
		
Flujos Alternos 4a: Contraseña incorrecta		
Acción del Actor	Respuesta del Sistema	
	4a.1 Indica que la contraseña es incorrecta Ir a la acción 2.	
Flujo Alternativo *a: Cancelar		
Acción del Actor	Respuesta del Sistema	
a.1 Da clic en el botón "Cancelar"	a.2 Cierra la ventana sin guardar los cambios.	
Poscondiciones	No aplica	

Tabla 2.10 Descripción del caso de uso, Mostrar total de llamadas.

Caso de Uso:	Mostrar total de llamadas.	
Actores:	AdministradorPT	
Resumen:	El caso de uso se inicia cuando el administrador de la pizarra telefónica posiciona el mouse sobre el icono de estado en el tray-bar. El sistema muestra un mensaje con el total de llamadas que se han registrado hasta ese momento.	
Precondiciones:	No aplica	
Referencias	RF9	
Prioridad	Secundario	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1 El administrador posiciona el Mouse sobre el icono de estado de la aplicación.	2 Consulta en la BD la cantidad de llamadas realizadas hasta el momento.	
	3. Muestra la cantidad de llamadas.	
	4. Pasado 15 segundos deja de mostrar el mensaje.	
Prototipo de Interfaz		
		
Poscondiciones	No aplica	

2.6 Conclusiones

En este capítulo fue descrita la propuesta de solución y las funcionalidades que el sistema debe cumplir a través de un Modelo de dominio. Se analizaron y describieron los requisitos funcionales y no funcionales. Además se identificaron y describieron los Actores y Casos de Uso, estableciéndose las relaciones correspondientes entre cada uno de ellos en el Diagrama de Casos de Uso del sistema.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 Introducción

Este capítulo tiene como objetivo principal realizar el modelo de análisis y diseño del sistema de monitoreo de registros de pizarras telefónicas. Se definen las entidades y las relaciones con los actores para poder efectuar los diagramas de clases del análisis y del diseño. Además se construyen los diagramas de secuencia. Se obtienen como resultado modelos imprescindibles para un mejor entendimiento del sistema. Este flujo de trabajo tiene una gran importancia en el ciclo de desarrollo, pues traduce los requisitos definidos a funcionalidades que debe realizar el producto. Se definen las clases persistentes y se obtiene el modelo de datos.

A pesar de que el diagrama de despliegue y el diagrama de componentes son artefactos que se generan en el flujo de trabajo de implementación, y que la investigación solo incluye hasta el diseño, para lograr un mejor resultado y continuidad del trabajo se desarrolló una propuesta de los mismos que se exponen al final del capítulo.

3.2 Modelo de análisis.

Un modelo de análisis estructura los requisitos de un modo que facilita su comprensión, su preparación, su modificación y en general, su mantenimiento. Puede considerarse como una primera aproximación al modelo de diseño. Las clases del análisis siempre encajan en uno de tres estereotipos básicos: de interfaz; de control o de entidad. Las clases de interfaz se utilizan para modelar la interacción entre el sistema y sus actores, las clases entidad se utilizan para modelar información que posee una vida larga y que es a menudo persistente y las clases de control representan coordinación, secuencia, transacciones y control de otros objetos.

3.2.1 Diagramas de clases del análisis.

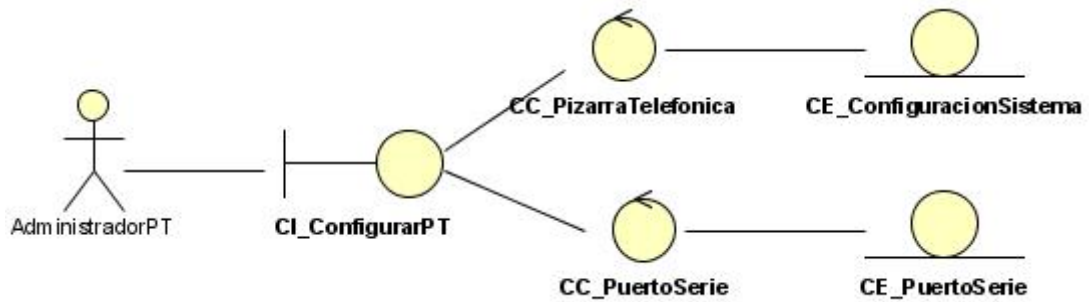


Fig.3.13 Diagrama de clases del análisis, Configurar sistema.

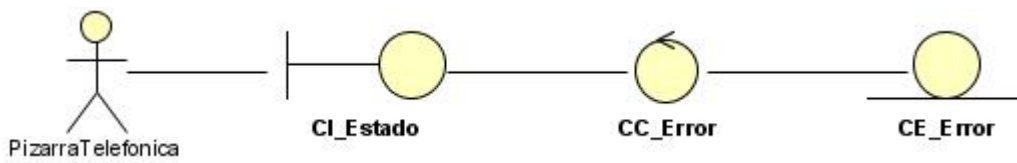


Fig.3.24 Diagrama de clases del análisis, Mostrar estado.

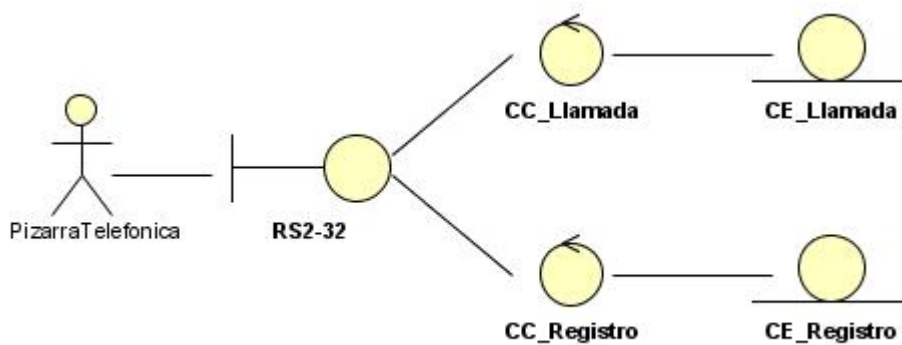


Fig.3.35 Diagrama de clases del análisis, Gestionar SMDR.

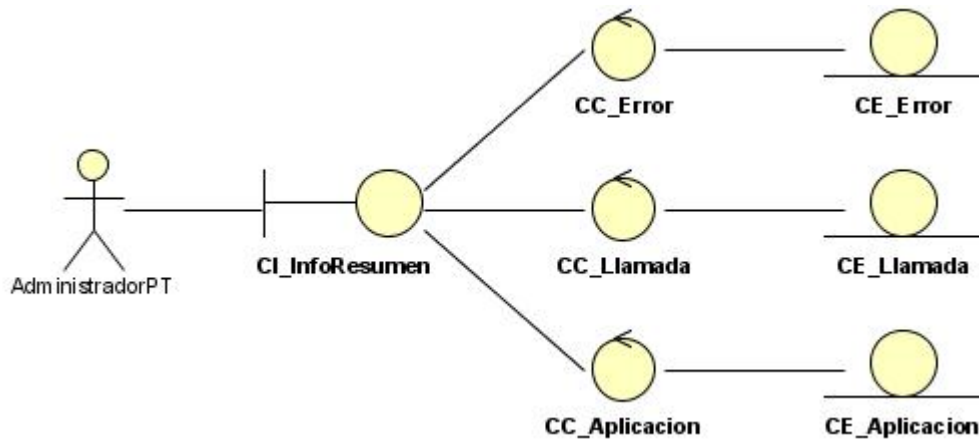


Fig.3.46 Diagrama de clases del análisis, Mostrar información resumen.

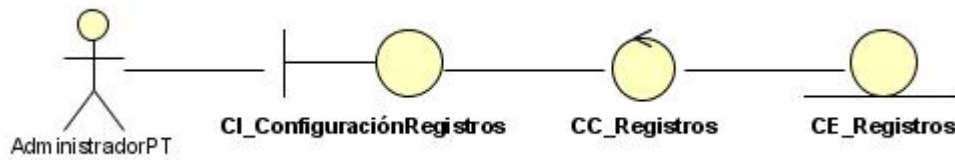


Fig.3.57 Diagrama de clases del análisis, Configurar registros.

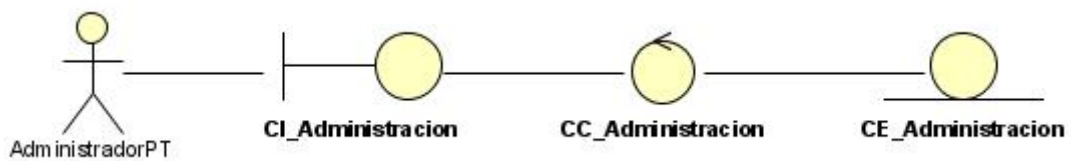


Fig.3.68 Diagrama de clases del análisis, Administrar aplicación.

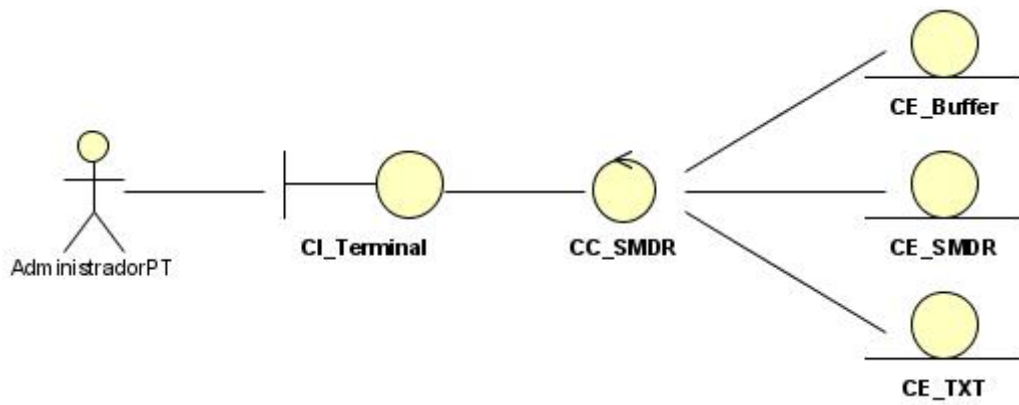


Fig.3.79 Diagrama de clases del análisis, Mostrar ventana Terminal.

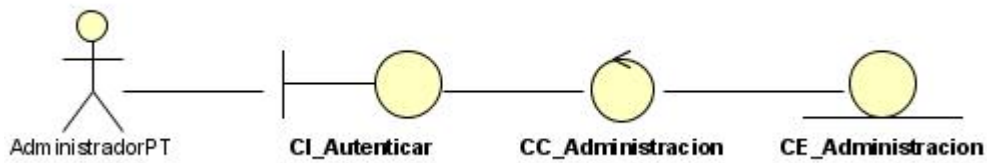


Fig.3.810 Diagrama de clases del análisis, Autenticar.



Fig.3.911 Diagrama de clases del análisis, Mostrar total de llamadas.

3.3 Modelo de diseño.

3.3.1 Diagrama de clases del diseño.

El diagrama de clases es una descripción de los modelos de objetos, contiene clases y las relaciones estructurales y de herencia existentes entre ellas. Este se obtiene como resultado del refinamiento del modelo conceptual y de los diagramas de secuencia. La definición de clase incluye definiciones para atributos y responsabilidades.

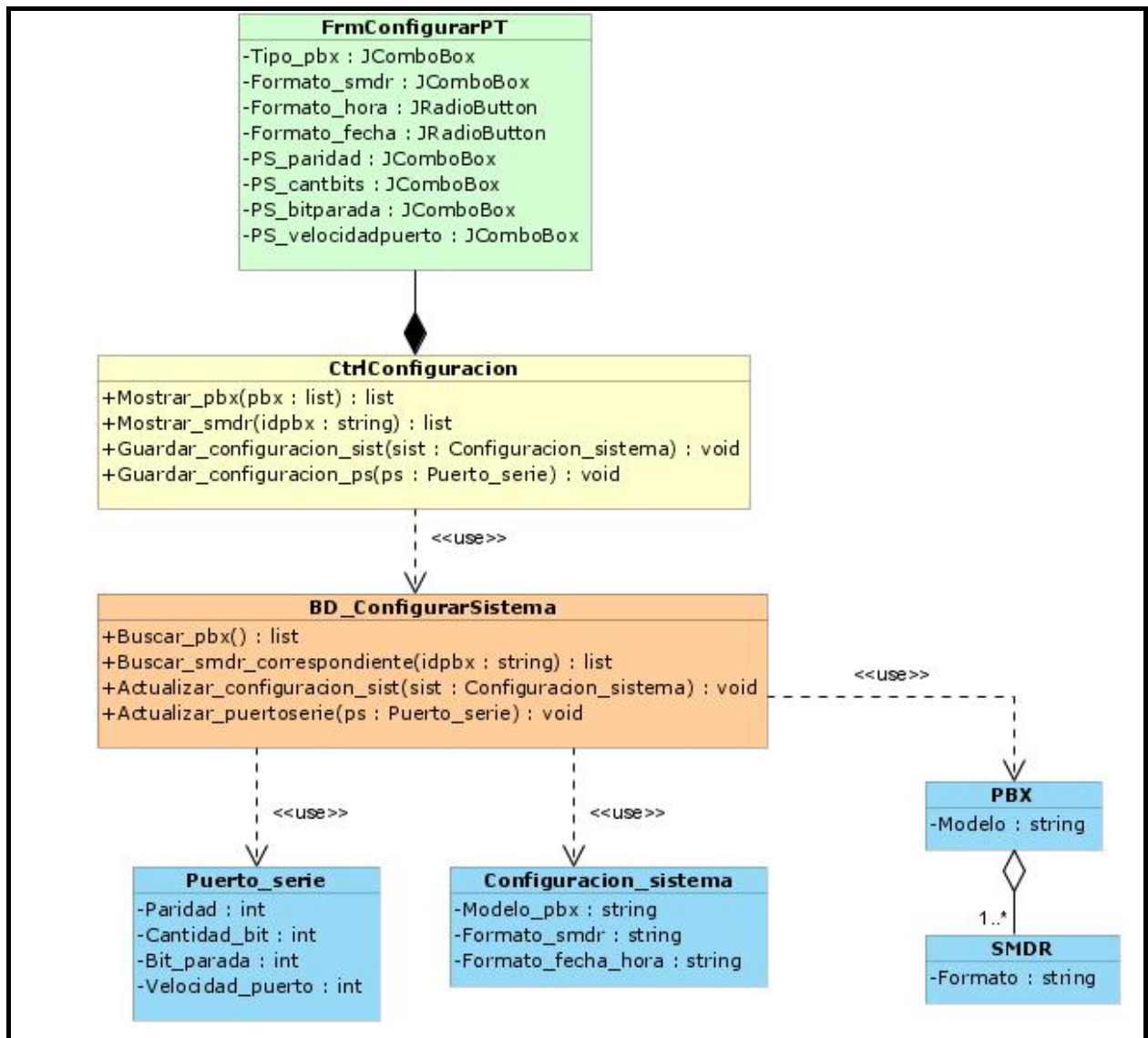


Fig.3.1012 Diagrama de clases del diseño, Configurar sistema.

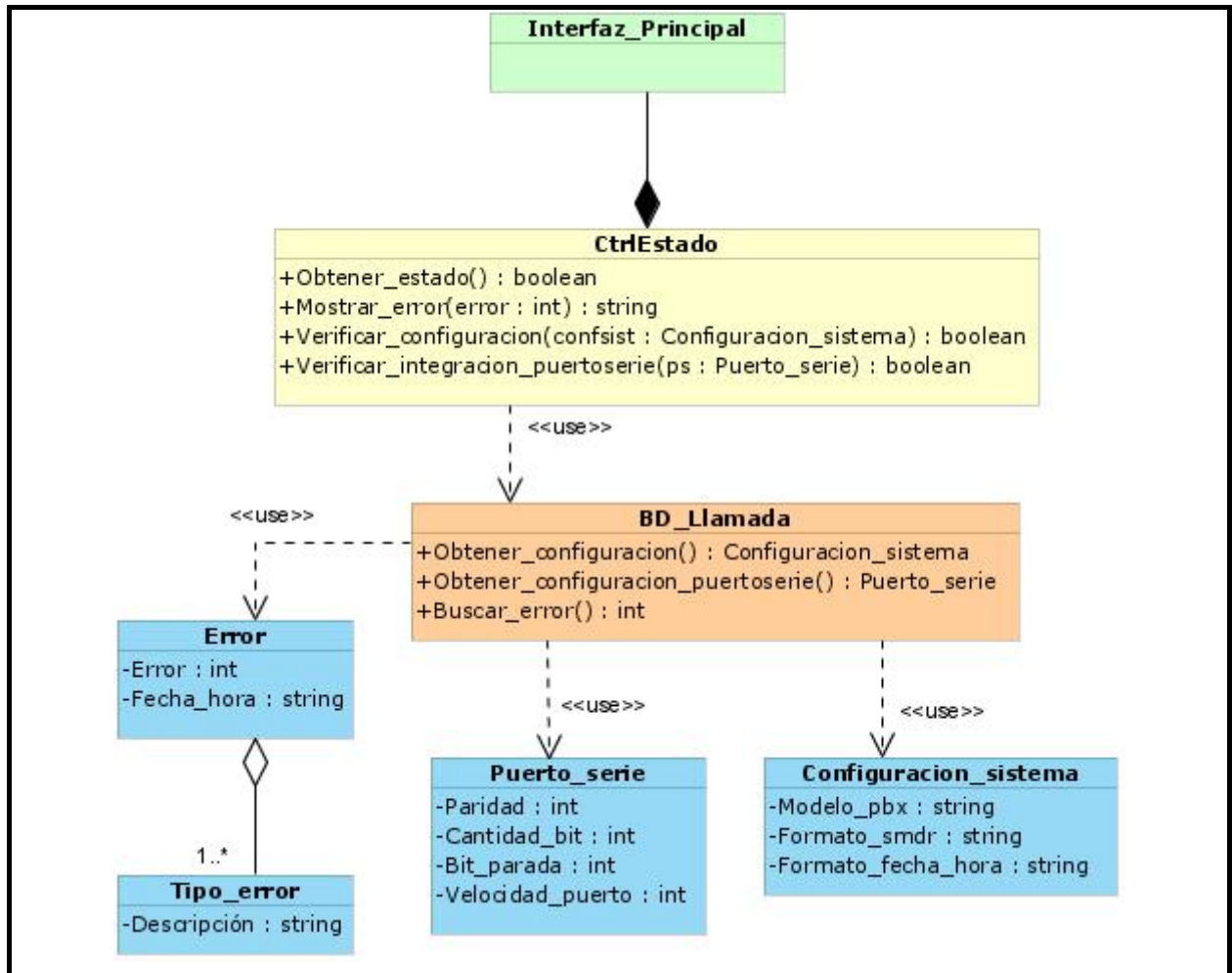


Fig.3.1113 Diagrama de clases del diseño, Mostrar estado.

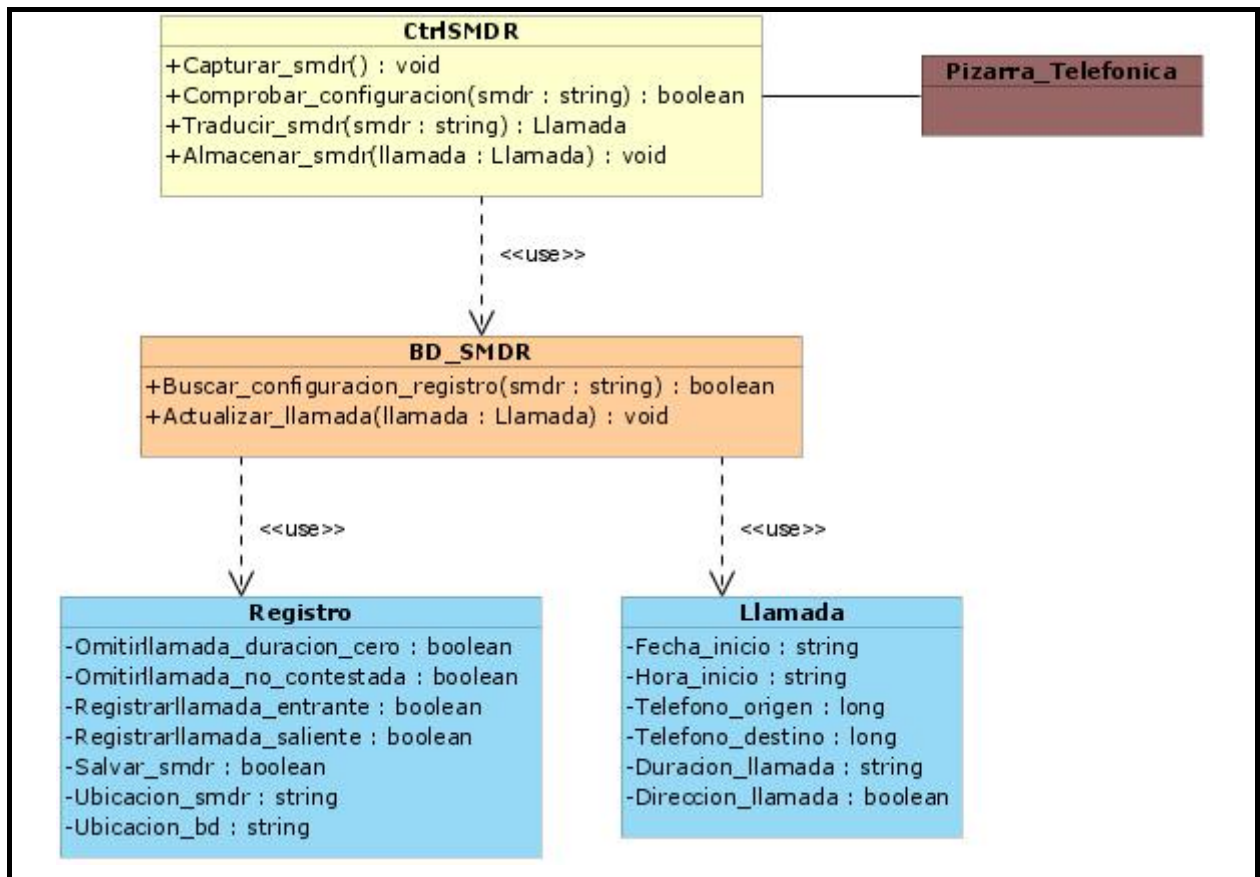


Fig.3.1214 Diagrama de clases del diseño, Gestionar SMDR.

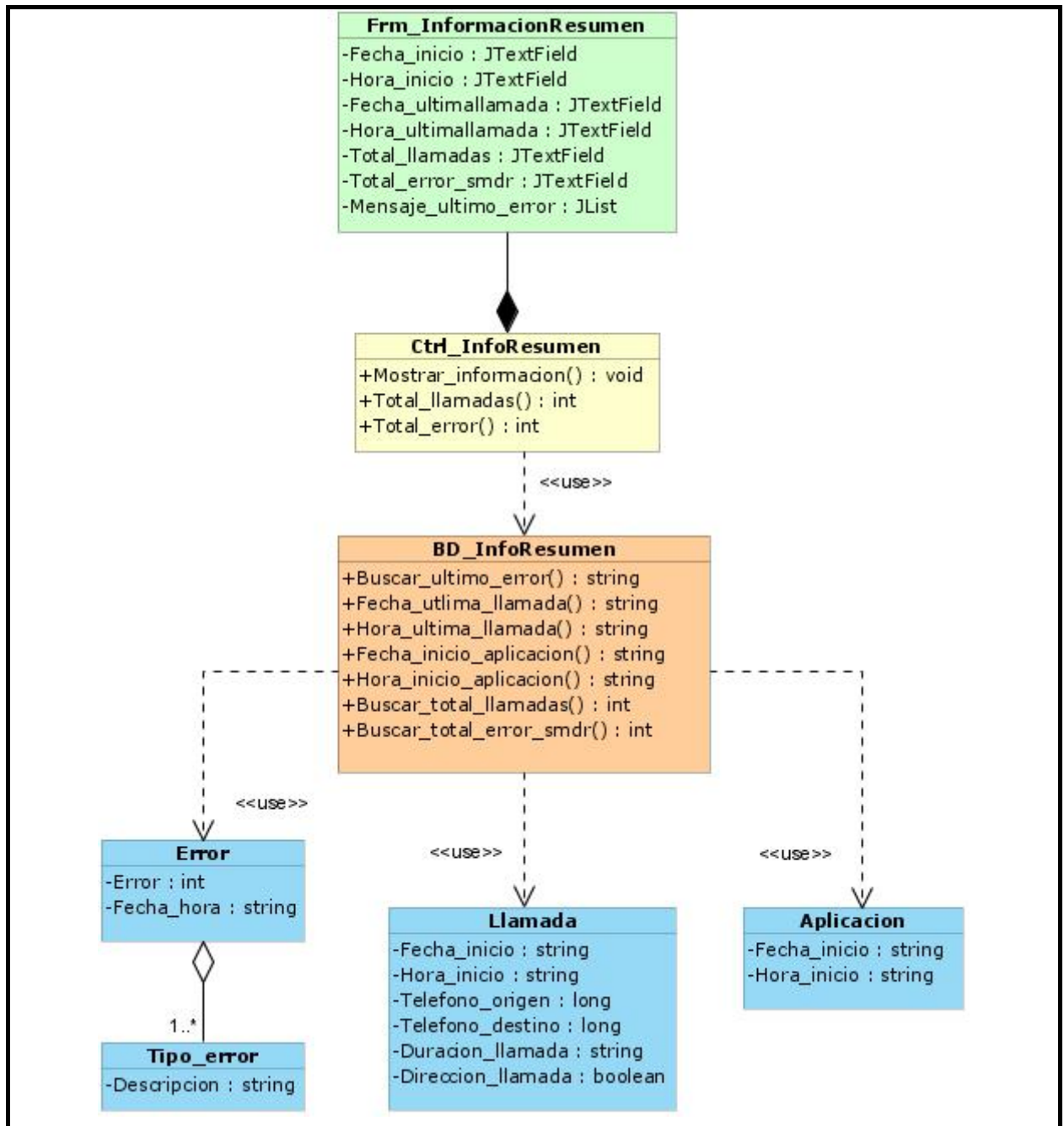


Fig.3.1315 Diagrama de clases del diseño, Mostrar información resumen.

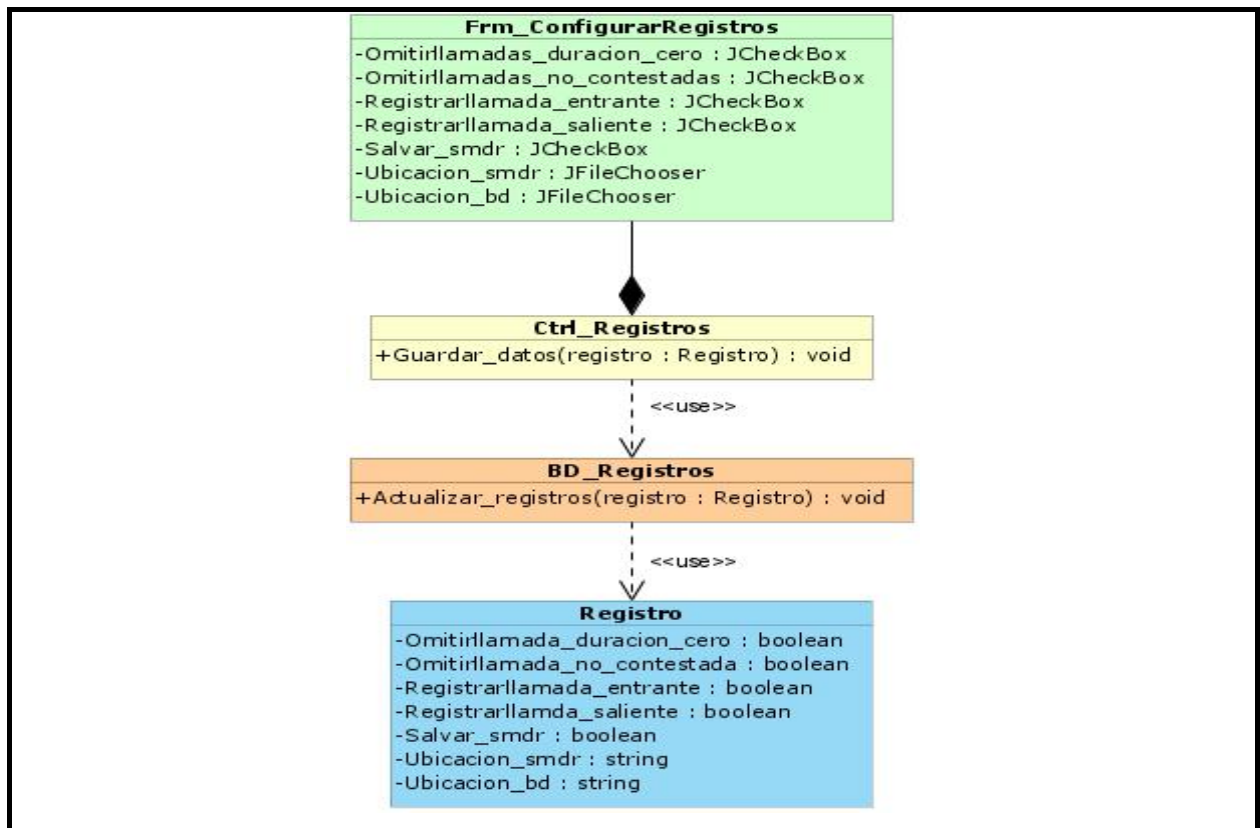


Fig.3.1416 Diagrama de clases del diseño, Configurar registro.

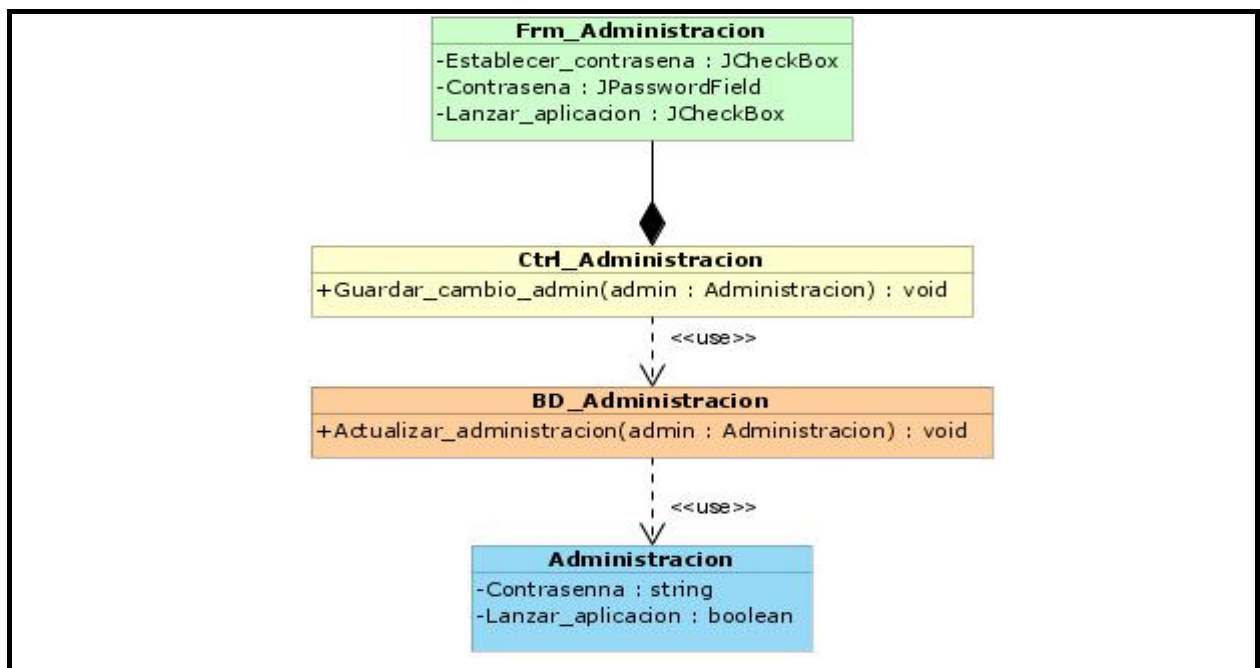


Fig.3.1517 Diagrama de clases del diseño, Administrar aplicación.

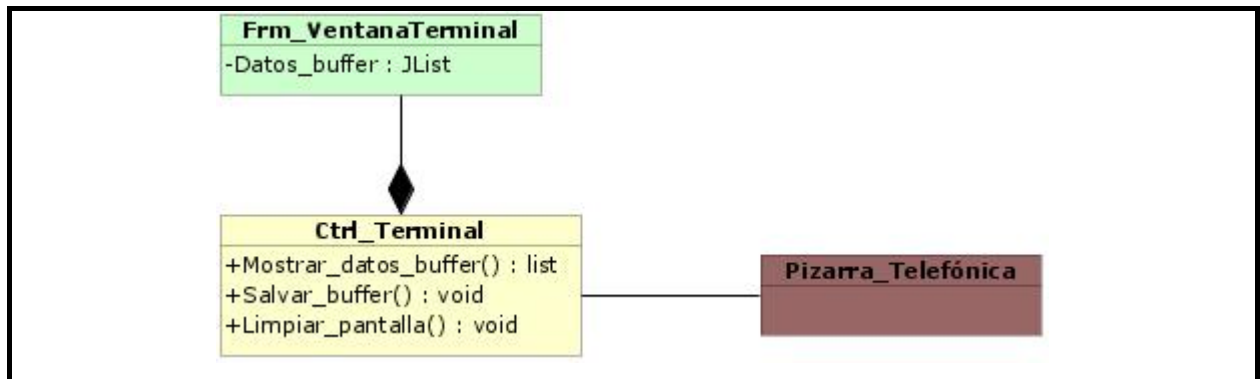


Fig.3.1618 Diagrama de clases del diseño, Mostrar ventana Terminal.

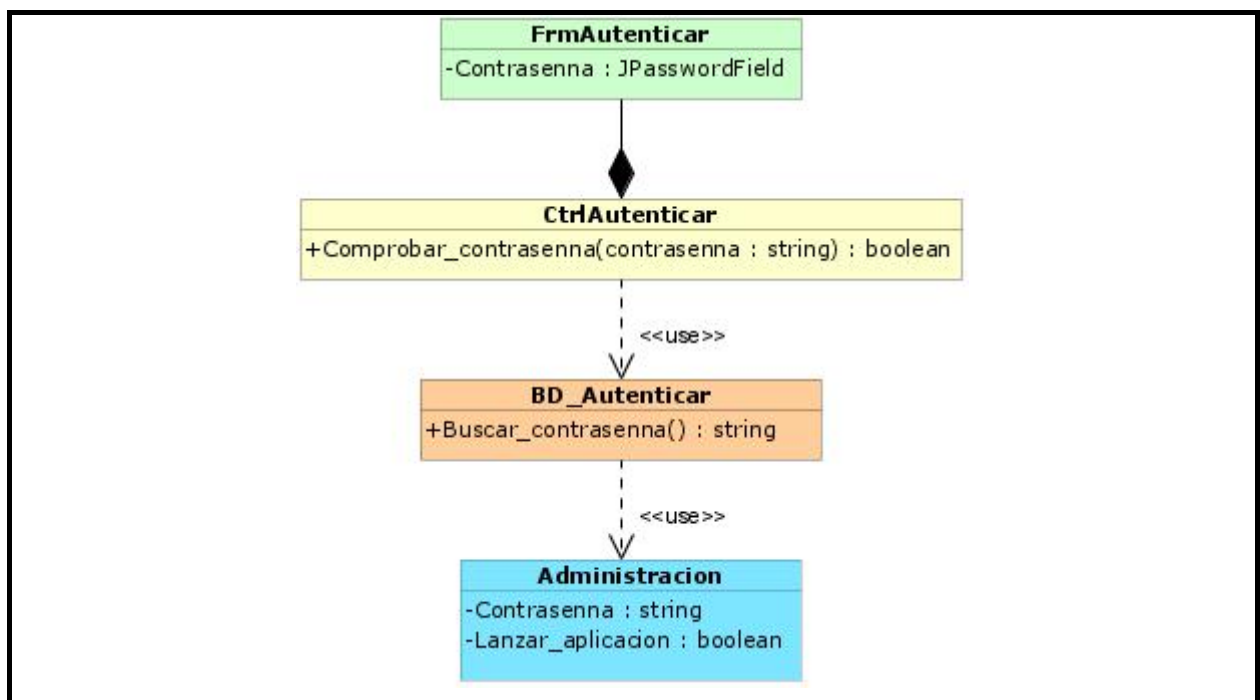


Fig.3.1719 Diagrama de clases del diseño, Autenticar.

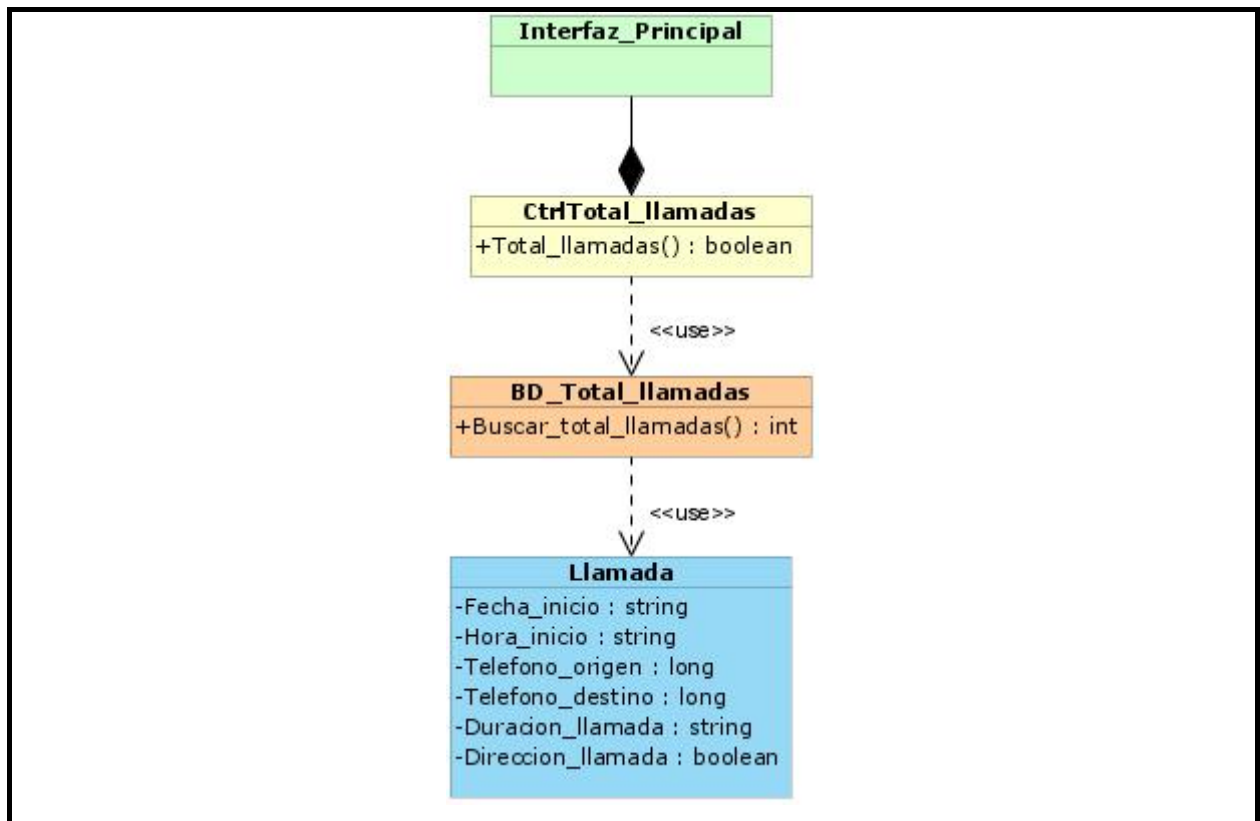


Fig.3.1820 Diagrama de clases del diseño, Mostrar total de llamadas.

Para una mejor comprensión del diseño, se muestra en un diagrama de clases del diseño general, que incluye a cada una de las clases utilizadas en el diseño (**Anexo 1. Diagrama de clases del diseño**).

3.3.2 Diagramas de interacción.

Los diagramas de interacción describen la forma en que cada operación detectada en los diagramas de secuencia lleva a cabo sus responsabilidades y modifica el estado del sistema, mostrando el modo en que los objetos interactúan a través de mensajes. En UML los diagramas de interacción pueden representarse a través de los diagramas de colaboración y/o de los diagramas de secuencia. Los diagramas de secuencia muestran interacciones entre objetos basadas en el tiempo.

(**Anexo 2. Diagramas de Secuencia del diseño**).

3.3.3 Patrones de diseño.

Un patrón es una descripción de un problema y su solución, que recibe un nombre y que puede emplearse en otros contextos. En el diseño del sistema se utilizaron los siguientes patrones.

Patrones de arquitectura

Los patrones arquitectónicos son la descripción de un problema particular y recurrente de diseño, que aparece en contextos de diseño específicos. Expresan el esquema de organización estructural fundamental para sistemas de software.

Un patrón de arquitectura provee un conjunto de subsistemas predefinidos, especifica sus responsabilidades e incluye reglas y pautas para la organización de las relaciones entre ellos. Propone plantillas para arquitecturas de software concretas, que especifican las propiedades estructurales de una aplicación con amplitud de todo el sistema y tienen un impacto en la arquitectura de subsistemas. La selección de un patrón arquitectónico es, por lo tanto, una decisión fundamental de diseño en el desarrollo de un sistema de software.

Patrón de arquitectura propuesto

El patrón arquitectónico aplicado es el de capas (*Layers*), pues el sistema estará estructurado específicamente en cuatro capas: la capa de presentación, capa lógica, capa de acceso a datos y la capa de datos. En esta estructura la capa inferior proporciona servicios a la superior. Las capas se tratarán de forma independiente, pues cada una encapsula un aspecto concreto del sistema.

Este patrón es importante porque simplifica la comprensión y la organización del desarrollo del sistema, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores.

Patrones de diseño

Los patrones de diseño son aquellos que expresan esquemas para definir estructuras de diseño (o sus relaciones) con las que construir sistemas de software. Estos identifican: clases, instancias, roles, colaboraciones y la distribución de responsabilidades.

Patrones GRASP

GRASP es un acrónimo que significa General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades). Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. GRASP contiene una lista de cinco patrones básicos que se refieren a cuestiones y aspectos fundamentales del diseño:

- **Experto**

Asigna responsabilidades al experto en información, o sea, las clases que tienen la información necesaria para cumplir con la responsabilidad.

- **Creador**

Guía la asignación de responsabilidades relacionadas con la creación de objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento.

La creación de objetos es una de las actividades más frecuentes en un sistema orientado a objetos. En consecuencia, conviene contar con un principio general para asignar las responsabilidades concernientes a ella. El diseño, bien asignado, puede soportar un bajo acoplamiento, una mayor claridad, el encapsulamiento y la reutilizabilidad.

- **Bajo Acoplamiento**

El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Una clase con bajo (o débil) acoplamiento no depende de muchas otras, mientras que una clase con alto (o fuerte) acoplamiento recurre a muchas otras.

- **Alta Cohesión**

Es asignar una responsabilidad de modo que la cohesión siga siendo alta. En la perspectiva del diseño orientado a objetos, la cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas y que no realicen un trabajo enorme.

- **Controlador**

Un Controlador es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema y define además el método de su operación. Un evento del sistema es un evento de alto nivel generado por un actor externo; es un evento de entrada externa. Se asocia a operaciones del sistema: las que emite en respuesta a los eventos del sistema.

3.4 Diseño de la base de datos.

3.4.1 Modelo lógico de datos.

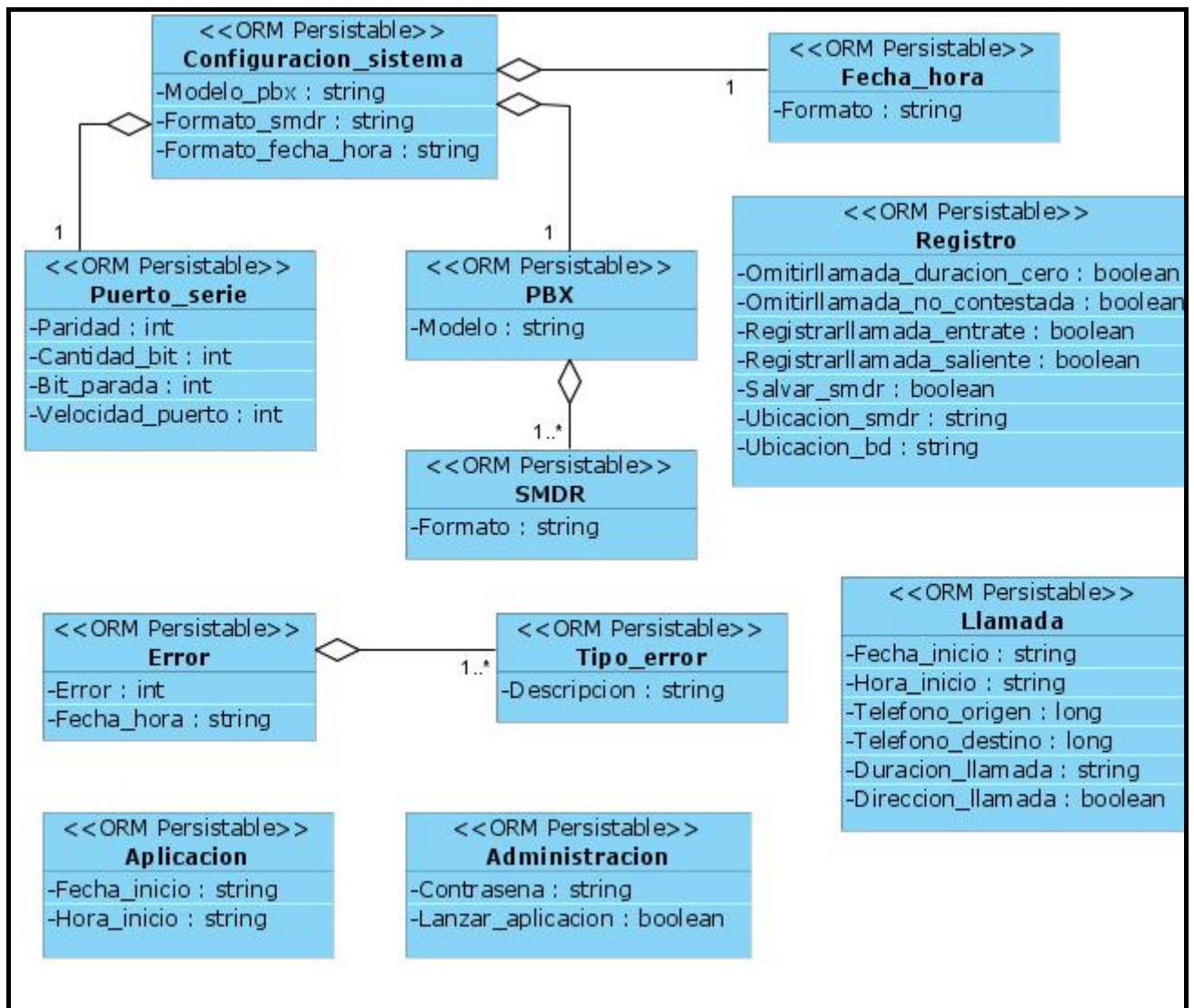


Fig.3.1921 Diagrama de clases persistentes del sistema.

3.4.2 Modelo físico de datos del sistema.

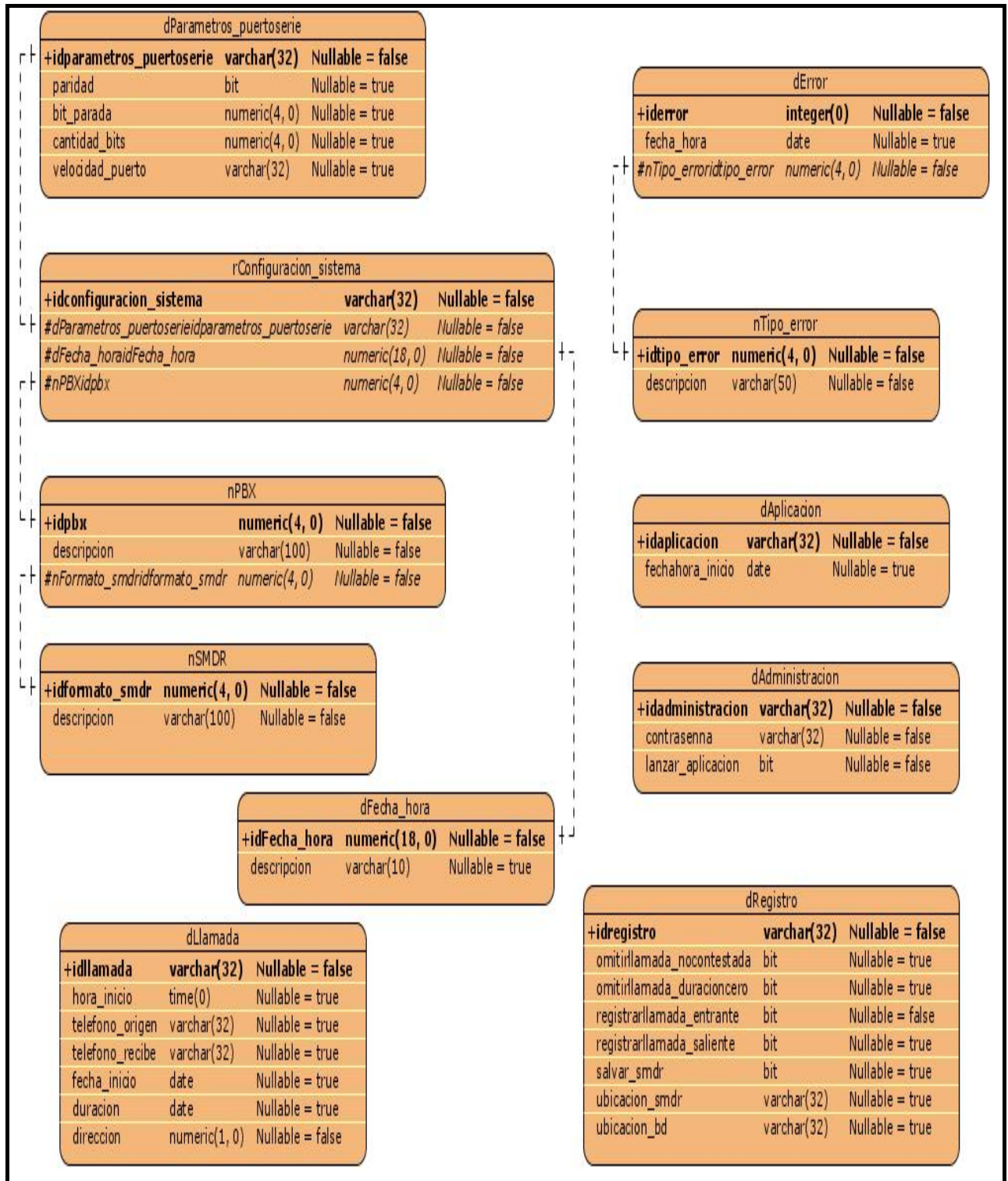


Fig.3.2022 Diagrama del modelo de datos del sistema.

3.5 Diagrama de despliegue.

El modelo de despliegue muestra la configuración de los nodos de procesamiento en tiempo de ejecución, los links de comunicación entre ellos. El propósito del modelo de despliegue es capturar la configuración de los elementos de procesamiento, y las conexiones entre estos elementos en el sistema. El modelo consiste en uno o más nodos (elementos de procesamiento con al menos un procesador, memoria, y posiblemente otros dispositivos), dispositivos (nodos estereotipados con una capacidad de procesamiento en el nivel modelado de abstracción), y conectores, entre nodos, y entre nodos y dispositivos. El modelo de despliegue también mapea procesos dentro de estos elementos de procesamiento, permitiendo la distribución del comportamiento a través de los nodos que son representados.

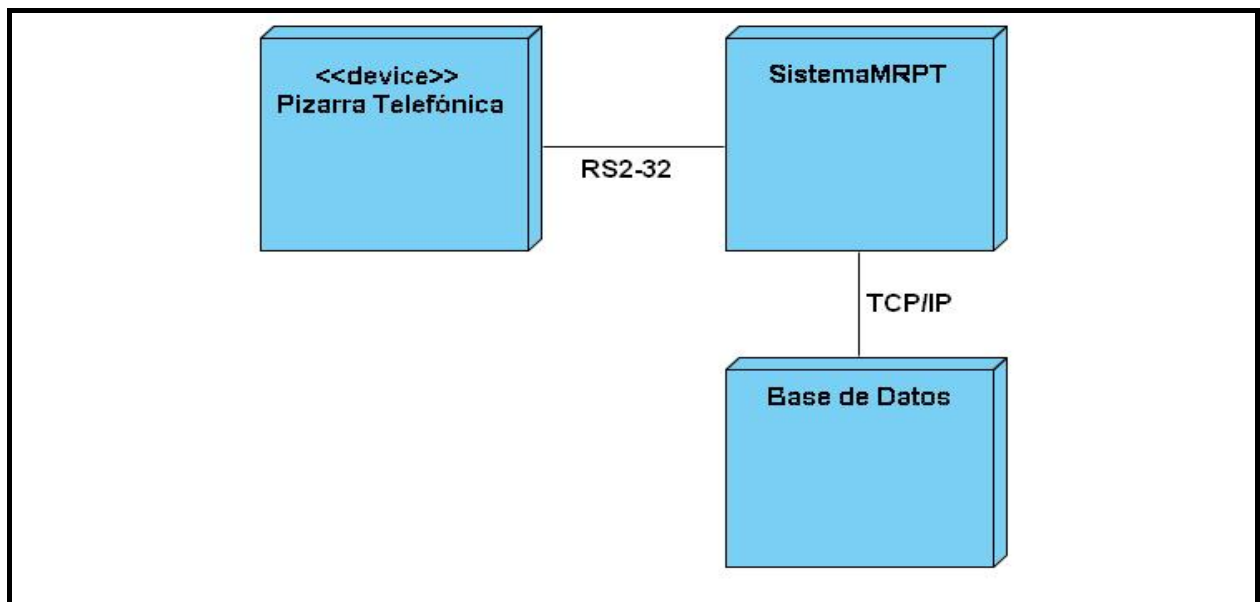


Fig.3.2123 Diagrama de Despliegue.

3.6 Diagrama de componentes.

Un componente es una parte modular de un sistema, desplegable y reemplazable que encapsula implementación y un conjunto de interfaces y proporciona la realización de los mismos. Además contiene clases y puede ser implementado por uno o más artefactos.

Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación. El uso más importante de estos diagramas es mostrar la estructura de alto nivel del modelo de implementación, especificando:

- Los subsistemas de implementación y sus dependencias a la hora de importar código.
- Organizar los subsistemas de implementación en capas.

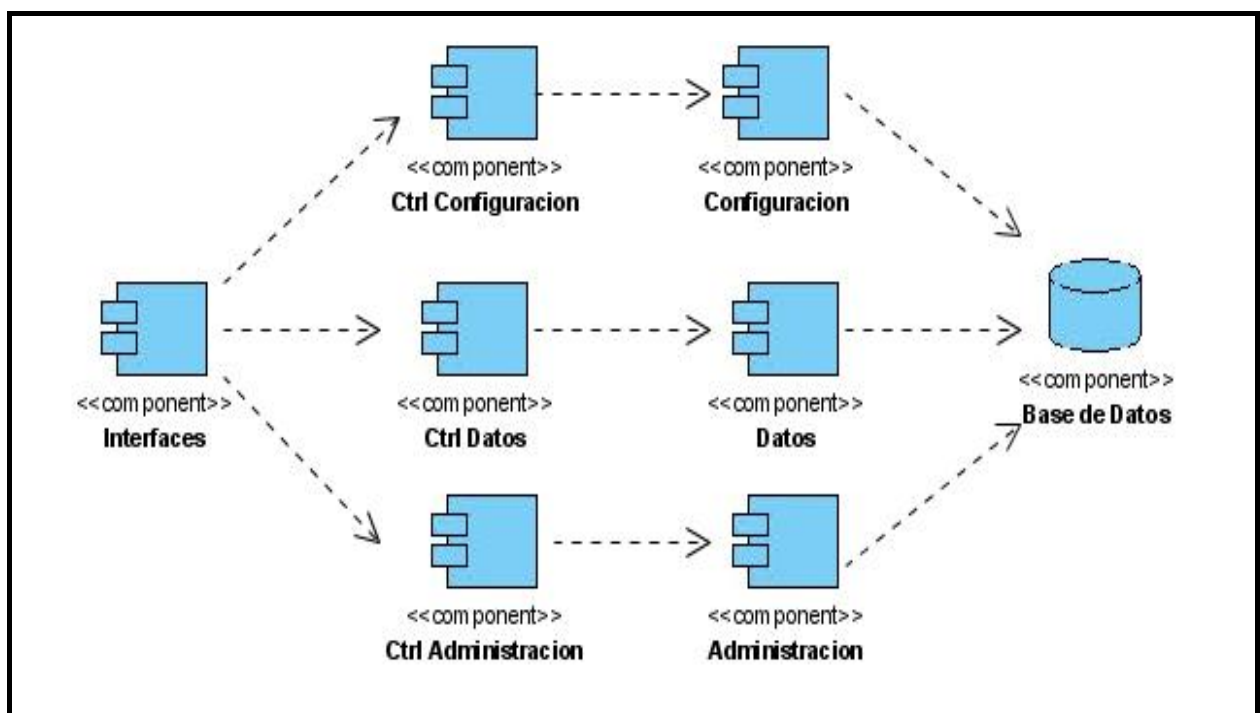


Fig.3.2224 Diagrama de Componentes.

3.7 Conclusiones

En este capítulo se desarrollaron los flujos de trabajo de Análisis y Diseño que propone la metodología RUP, modelando los diagramas fundamentales de ambos flujos. Se expusieron además los patrones utilizados y se hizo una propuesta de diagrama de despliegue y del diagrama de componentes, que servirán de apoyo a la implementación del sistema en próximas iteraciones.

CAPÍTULO 4: ESTUDIO DE FACTIBILIDAD.

4.1 Introducción

Como parte del desarrollo de un proyecto es muy importante realizar la estimación y planificación del mismo. De esta manera se determina lo que hace falta, en términos de costos, recursos humanos, materiales y tiempo; así como la factibilidad y viabilidad existente para desarrollar el sistema propuesto. En este capítulo se abordará el tema de la estimación a través de la técnica de Análisis de Puntos de Casos de Uso.

4.2 Método de estimación por Puntos de Casos de Uso.

Con esta técnica existe la posibilidad de predecir el tamaño de un sistema a partir de las características de sus requisitos, expresados en los casos de uso. La estimación mediante el análisis de Puntos de Casos de Uso es un método propuesto originalmente por Gustav Karner de Objectory AB, y posteriormente refinado. Se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores.

4.3 Planificación

4.3.1 Cálculo de Puntos de Casos de Uso sin ajustar.

Se tiene la siguiente ecuación:

$$UUCP = UAW + UUCW$$

Donde:

UUCP: Puntos de Casos de Uso sin ajustar.

UAW: Factor de Peso de los Actores sin ajustar.

UUCW: Factor de Peso de los Casos de Uso sin ajustar.

Tipo	Descripción	Peso	Cant* peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, Application Programming Interface).	1	0*1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2	1*2

Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3	1*3
Total UAW			5

Tabla 4.1 Para calcular el Factor de Peso de los Actores sin ajustar (**UAW**).

Tipo	Descripción	Peso	Cant* peso
Simple	El Caso de Uso contiene de 1 a 3 transacciones.	5	1*5
Medio	El Caso de Uso contiene de 4 a 7 transacciones.	10	2*10
Complejo	El Caso de Uso contiene más de 8 transacciones.	15	4*15
Total UUCW			85

Tabla 4.2 Para calcular el Factor de Peso de los Caso de Uso sin ajustar (**UUCW**).

Luego:

$$UUCP = 5 + 85$$

$$UUCP = 90 \text{ (Factor de Peso de los Casos de Uso sin ajustar)}$$

4.3.2 Cálculo de Puntos de Casos de Uso ajustados.

Se tiene la siguiente ecuación:

$$UCP = UUCP * TCF * EF$$

Donde:

UCP: Puntos de Casos de Uso ajustados.

UUCP: Puntos de Casos de Uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

Factor	Descripción	Peso	Valor	Σ (Pesoi * Valori)
T1	Sistema distribuido	2	0	0
T2	Objetivos de performance o tiempo de respuesta	1	5	5
T3	Eficiencia del usuario final	1	4	4

T4	Procesamiento interno complejo	1	5	5
T5	El código debe ser reutilizable	1	4	4
T6	Facilidad de instalación	0.5	3	1.5
T7	Facilidad de uso	0.5	3	1.5
T8	Portabilidad	2	3	6
T9	Facilidad de cambio	1	3	3
T10	Concurrencia	1	0	0
T11	Incluye objetivos especiales de seguridad	1	5	5
T12	Provee acceso directo a terceras partes	1	0	0
T13	Se requieren facilidades especiales de entrenamiento a los usuarios	1	3	3
Total TCF				38

Tabla 4.3 Para calcular Factor de complejidad técnica (TCF).

Para Calcular TCF

$$TCF = 0.6 + 0.01 * \sum (\text{Peso } i * \text{Valor } i)$$

$$TCF = 0.6 + 0.01 * 38$$

$$TCF = 0.98 \text{ (Factor de complejidad técnica)}$$

Factor	Descripción	Peso	Valor	$\sum (\text{Peso } i * \text{Valor } i)$
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	1	1.5
E2	Experiencia en la aplicación.	0.5	0	0
E3	Experiencia en orientación a objetos	1	4	4
E4	Capacidad del analista líder	0.5	4	2
E5	Motivación	1	5	5
E6	Estabilidad de los requerimientos	2	3	6
E7	Personal part-time	-1	5	-5
E8	Dificultad del lenguaje de programación	-1	3	-3
Total TCF				10.5

Tabla 4.4 Para calcular el Factor Ambiente (EF).

Para Calcular EF

$$EF = 1.4 - 0.03 * \Sigma (\text{Peso } i * \text{Valor } i)$$

$$EF = 1.4 - 0.03 * 10.5$$

$$EF = 1.08 \text{ (Factor de ambiente)}$$

Luego

$$UCP = 90 * 0.98 * 1.08$$

$$UCP = 95.256 \text{ (Puntos de Casos de Uso ajustados)}$$

4.3.3 Calcular esfuerzo del flujo de trabajo Implementación.

Se tiene la siguiente ecuación:

$$E = UCP * CF$$

Donde:

E: Esfuerzo estimado en horas-hombre.

UCP: Puntos de Casos de Uso ajustados.

CF: Factor de conversión.

Para calcular CF

CF = 20 horas-hombre (si Total EF \leq 2)

CF = 28 horas-hombre (si Total EF = 3 ó Total EF = 4)

CF = abandonar o cambiar proyecto (si Total EF \geq 5)

Total EF = Cant EF < 3 (entre E1 –E6) + Cant EF > 3 (entre E7, E8)

Total EF = 3+0

Total EF = 3.

CF = 28 horas-hombre (porque Total EF = 3)

$$E = UCP * CF$$

$$E = 95.256 * 28 \text{ horas-hombre}$$

$$E = 2667.168 \text{ horas-hombre (Esfuerzo estimado en horas-hombre)}$$

4.3.4 Calcular el esfuerzo total de todo el proyecto.

Actividad	% Esfuerzo	Valor esfuerzo
Análisis	10%	666.792
Diseño	20%	1333.584
Implementación	40%	2667.168
Prueba	15%	1000.188
Sobrecarga	15%	1000.188
Total	100%	6667.92

El esfuerzo total de todo el proyecto sería $E_T = 6667.92$ horas-hombre. Considerando nuestro trabajo sólo hasta el flujo de trabajo de Diseño, tendríamos un valor de $E_T=2000.37$ horas-hombre. Si se estima que cada mes tiene 176 horas laborables entonces: $E_T = 11$ mes-hombre.

4.4 Cálculo del costo

Se tiene la siguiente ecuación:

$$CHM = CH * S * H$$

Donde:

Salario mensual por hombre ($S * H$) = \$100

Cantidad de hombres (CH) = 2

Luego:

$$CHM = \$200/mes$$

$$Costo = CHM * ET / CH$$

$$Costo = 200 * 11 / 2$$

$$Costo = \$ 1100$$

Tiempo total del Proyecto:

$$Tiempo = ET / CH$$

Tiempo = 11 meses/ 2 hombres

Tiempo = 5meses y 15 días.

Atendiendo a los resultados obtenidos se puede concluir que con 2 hombres trabajando en el proyecto se puede obtener una propuesta en 5 meses y 15 días con un costo estimado de \$1100.

4.5 Beneficios tangibles e intangibles.

Se definen a los **beneficios tangibles** como aquellos que reportan ventajas económicas cuantificables.

- Se puede utilizar el software para cualquier tipo de pizarra telefónica.
- Reporta grandes ventajas a las empresas debido a que en cualquier momento en que el usuario lo solicite le pueden ofrecer una factura detallada con todas las operaciones telefónicas que este ha realizado, evitando así confusiones y errores.

Se definen los **beneficios intangibles** como aquellos que reportan beneficios organizativos, de funcionamiento o eficiencia. Facilidad de operación con una interfaz sencilla y agradable a la vista, con resultados inmediatos.

4.6 Análisis de costo y beneficios.

El sistema propuesto se desarrolla mayormente con herramientas de software libre, por lo que no se requieren gastos de licencias de productos, y aunque se utilizan recursos en su desarrollo, no resulta costosa su creación.

Teniendo en cuenta el análisis, realizado en el estudio de factibilidades, y los beneficios tangibles e intangibles mencionados anteriormente, se puede plantear que el sistema es factible y será de gran utilidad para la empresa de telecomunicaciones de Cuba, ETECSA y sus usuarios.

4.7 Conclusiones

En este capítulo se llevó a cabo el estudio de factibilidad correspondiente al sistema propuesto, tomando en consideración la estimación de costos y los beneficios que reportará. Planteando que la herramienta desarrollada será económica y de mucha utilidad.

CONCLUSIONES

En el presente trabajo se investigaron las principales características y funcionamiento de las pizarras telefónicas o PBX más comercializadas en Cuba. Se analizaron sistemas que realizan funciones similares a las del sistema de monitoreo de SMDR. Se seleccionó la metodología RUP para desarrollar el análisis y el diseño del sistema. Fueron propuestas herramientas óptimas para la implementación del sistema.

De forma general se realizó el análisis y el diseño del módulo a desarrollar tributando a la construcción de un sistema capaz de monitorear los registros generados por las pizarras telefónicas. Se puede concluir que se ha cumplido satisfactoriamente con el objetivo trazado inicialmente para el desarrollo del trabajo.

RECOMENDACIONES

Implementar la solución propuesta del Sistema de monitoreo de registros de pizarras telefónicas en este trabajo.

Adicionar una nueva funcionalidad a la aplicación que permita al administrador insertar, a través de una interfaz visual, las pizarras telefónicas con las que la aplicación trabajará posteriormente, así como los diferentes formatos SMDR que esta puede generar.

BIBLIOGRAFÍA

1. Olivera Medrano, Joan. *Sistema de gestión de servicios de la central telefónica digital de la UCI*. Ciudad de la Habana. Julio, 2004.
2. Maranje Agramonte, Adrian. *Terminal de comunicación para el conmutador digital Infinity TX2400L/XL, la actualización automática de datos y configuración de servicios telefónicos*. Ciudad de la Habana. 2004.
3. Alonso Maceda, José Carlos. Álvarez de la Cruz, Darién Jesús. *Implementación de protocolos para la comunicación con Centros de Mensajería Inalámbrica*. Ciudad de la Habana. Mayo, 2007.
4. ETECSA. *PBX Call Tarificator Pro*. Febrero, 2008.
5. ETECSA. *Recorded Information SMDR*. Febrero, 2008.
6. Martínez, Luis Felipe. Teran T, Wilson. *Manual de configuración del PBX Alcatel OmniPCX*. Junio, 2007.
7. Bermello López, Yiset. *Pizarras telefónicas: comunicaciones eficientes para su empresa*. 2004. [2008] Disponible en:
<http://www.opciones.cu/suplementos/fihav-2004/html/pizarras.htm>
8. Telsoft. *PBX Polling, Corporate Bill Back, Database synchronization*. 2005. [2008]. Disponible en:
<http://www.telsoft-solutions.com/callaccount.html>
9. Telsoft. *PBX Traffic Study Analysis*. 2005. [2008]. Disponible en:
<http://www.telsoft-solutions.com/traffic.html>
10. Wikipedia. *PBX*. 2007. [2008]. Disponible en: <http://es.wikipedia.org/wiki/PBX>
11. *Continuación del FT Análisis de Diseño. Modelo de Diseño*. 2008. [2008] Disponible en:
<http://teleformacion.uci.cu/mod/resource/view.php?id=21372>
12. RUMBAUGH, James, JACOBSON, Ivar; BOOCH, Grady, *El lenguaje unificado de modelado*.
13. LARMAN, G. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. 2000. 507 p.
14. NEUROTEC. *NeuroTel V1.0 Programa de Captura y gestión de datos SMDR*. 2001. [2008]. Disponible en: <http://members.fortunecity.com/neurotec/software/neurotel/neurotel2.htm>
15. CPanax. *Tarificador Telefónico*. 2000. [2008] Disponible en:

<http://www.centrum.com.ve/cpanax.htm>

16. Oktaba, Hanna. *Introducción a Patrones*. [2008] Disponible en:

<http://209.85.215.104/search?q=cache:WksTkUj-DxIJ:www.mcc.unam.mx/~cursos/Algoritmos/javaDC99-2/patrones.html+patrones+de+arquitectura&hl=es&ct=clnk&cd=1&gl=cu>

17. Wikipedia. *Patrón de diseño*. [2008] Disponible en:

http://es.wikipedia.org/wiki/Patrones_de_dise%C3%B1o#Categor%C3.ADas_de_patrones

18. Data & Object Factory. *Desing Patterns*.2001 [2008] Disponible en:

<http://www.dofactory.com/Patterns/Patterns.aspx>

19. Ciberaula. *Patrones de Diseño en aplicaciones Web con Java J2EE*.2006.[2008] Disponible en:

http://209.85.215.104/search?q=cache:vLKwwU1ol1QJ:java.ciberaula.com/articulo/disen%C3%B3_patrones_j2ee/+patrones+GOF&hl=es&ct=clnk&cd=4&gl=cu

20. Larman Craig. *Patrones GRASP*.2006. [2008]. Disponible en:

<http://jorgesaavedra.wordpress.com/category/patrones-grasp/>

21. Wikipedia. *PostgreSQL*.2008. [2008]. Disponible en:

http://209.85.215.104/search?q=cache:JS9_Jph9oCkJ:en.wikipedia.org/wiki/PostgreSQL+PostgreSQL&hl=es&ct=clnk&cd=6&gl=cu

22. García Joaquín. *Patrones de diseño, diseño de software orientado a objetos*.2003. [2008].

Disponible en: <http://www.ingenierosoftware.com/analisisydise%C3%B1o/patrones-dise%C3%B1o.php>

23. Worsley ,John. Drake, Joshua. *¿Qué es PostgreSQL?* 2000.[2008] Disponible en:

<http://www.sobl.org/traduccion/practical-postgres/node12.html>

24. Universidad de Jaén. *Ingeniería del Software de Gestión*.2008 [2008].Disponible en:

http://209.85.165.104/search?q=cache:w55UOuM2A5UJ:wwdi.ujaen.es/asignaturas/isg/Software.html+visual+paradigm+for+UML+6.1&hl=es&ct=clnk&cd=3&gl=cu&lr=lang_es

ANEXOS

Anexo1. Diagrama de clases del diseño.

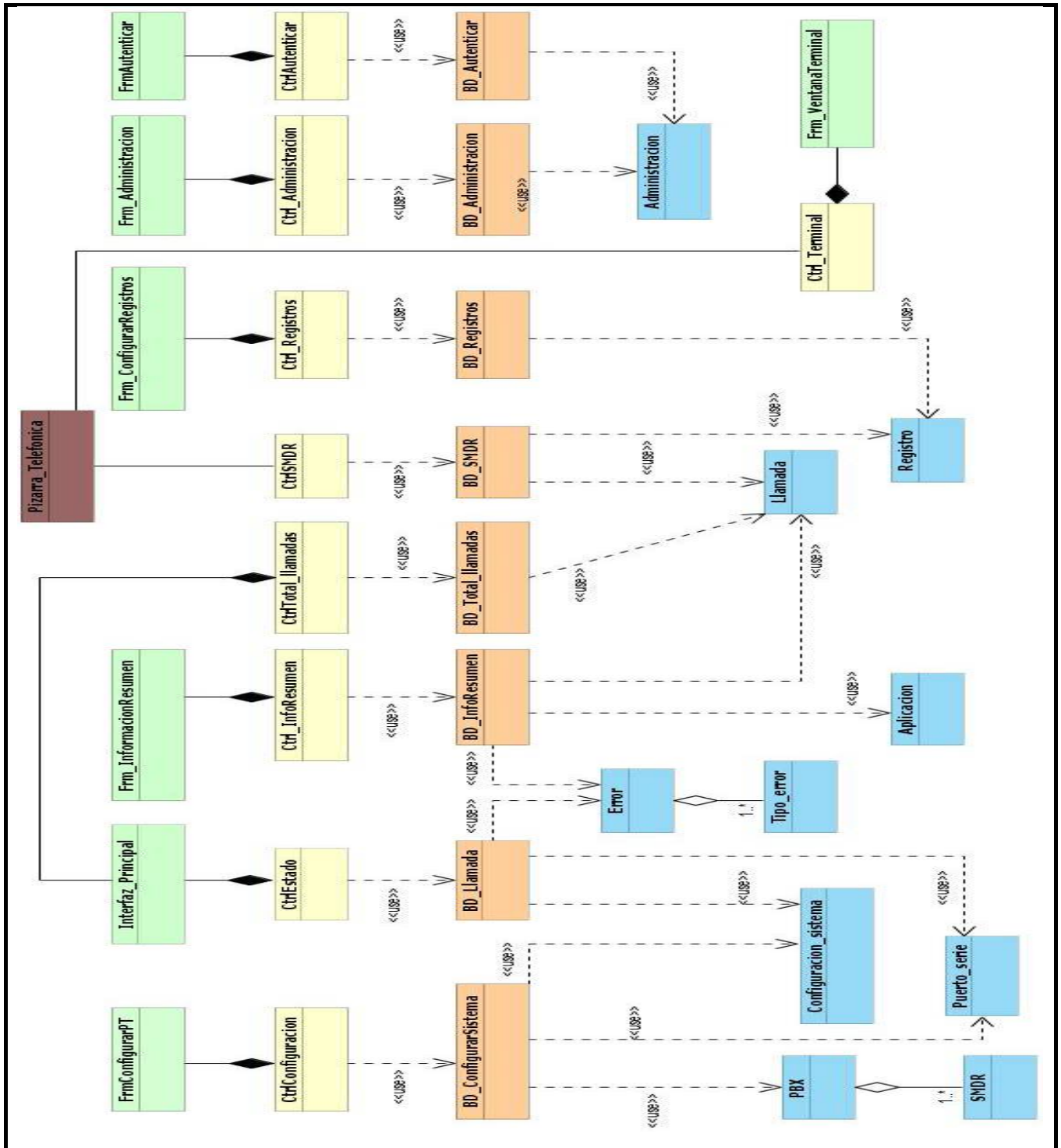


Fig. 4.1 Diagrama de clases del diseño.

Anexo 2. Diagramas de Secuencia del diseño.

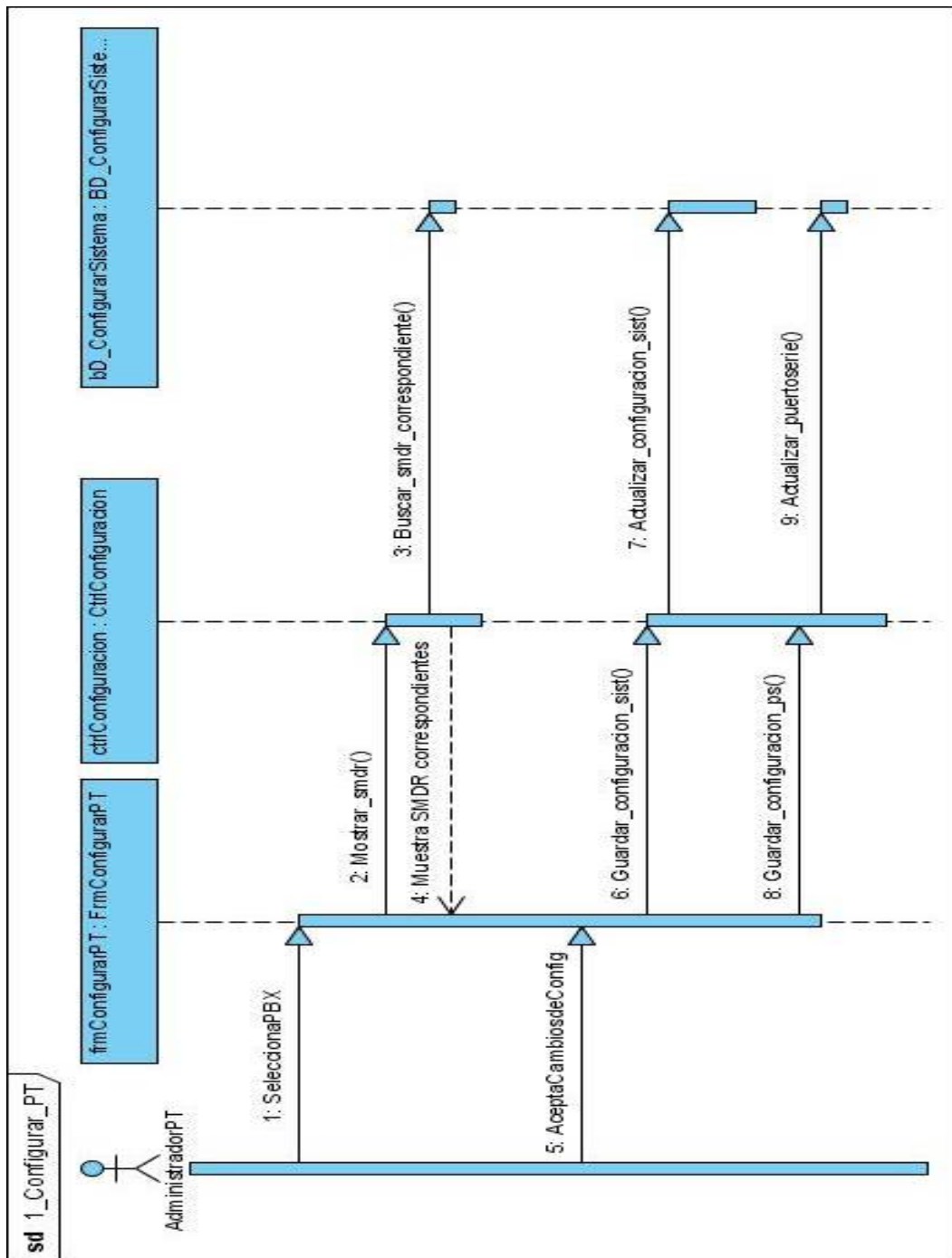


Fig 4.2 Diagrama de secuencia del diseño, Configurar sistema.

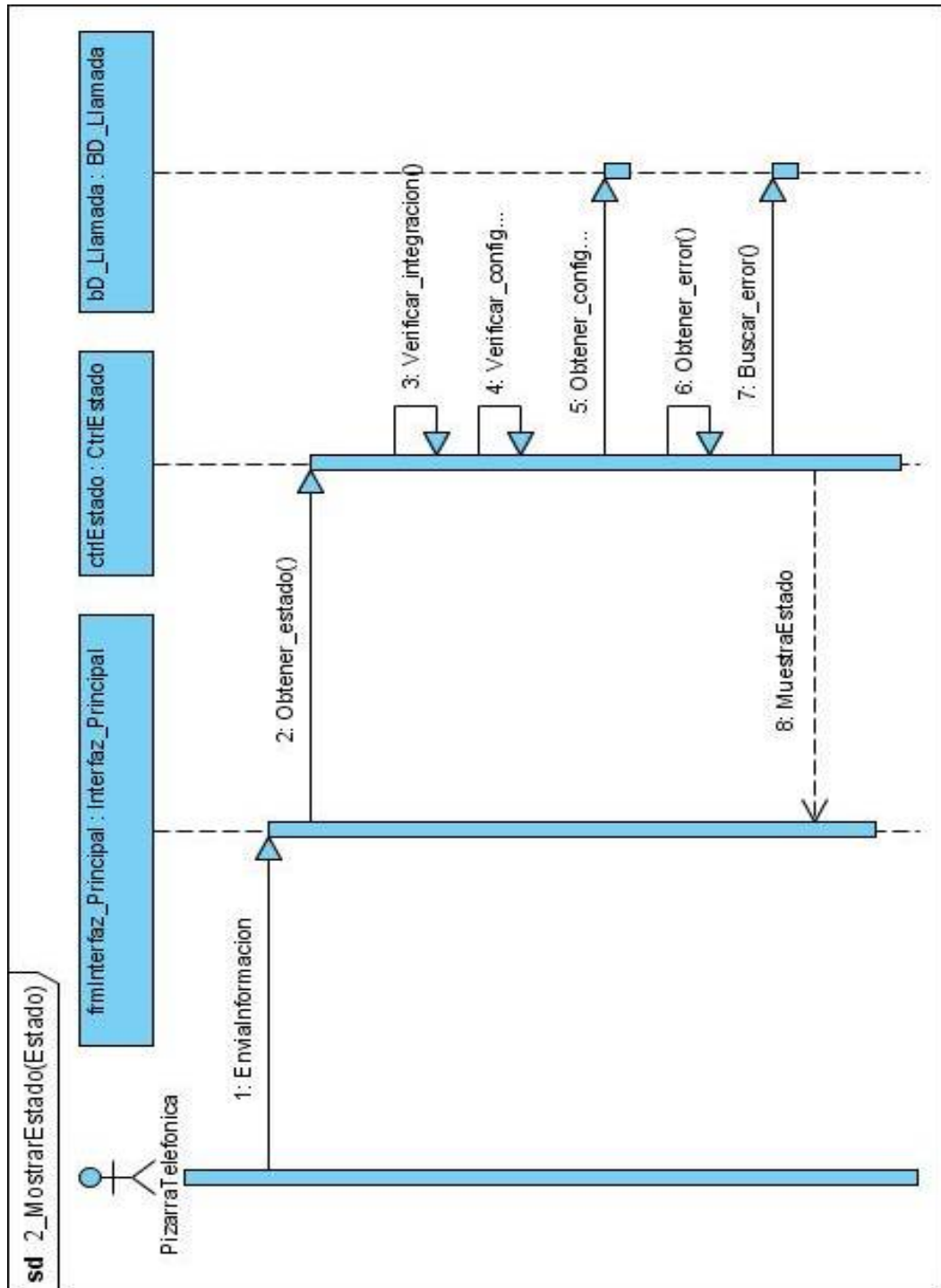


Fig 4.3 Diagrama de secuencia del diseño, Mostrar estado.

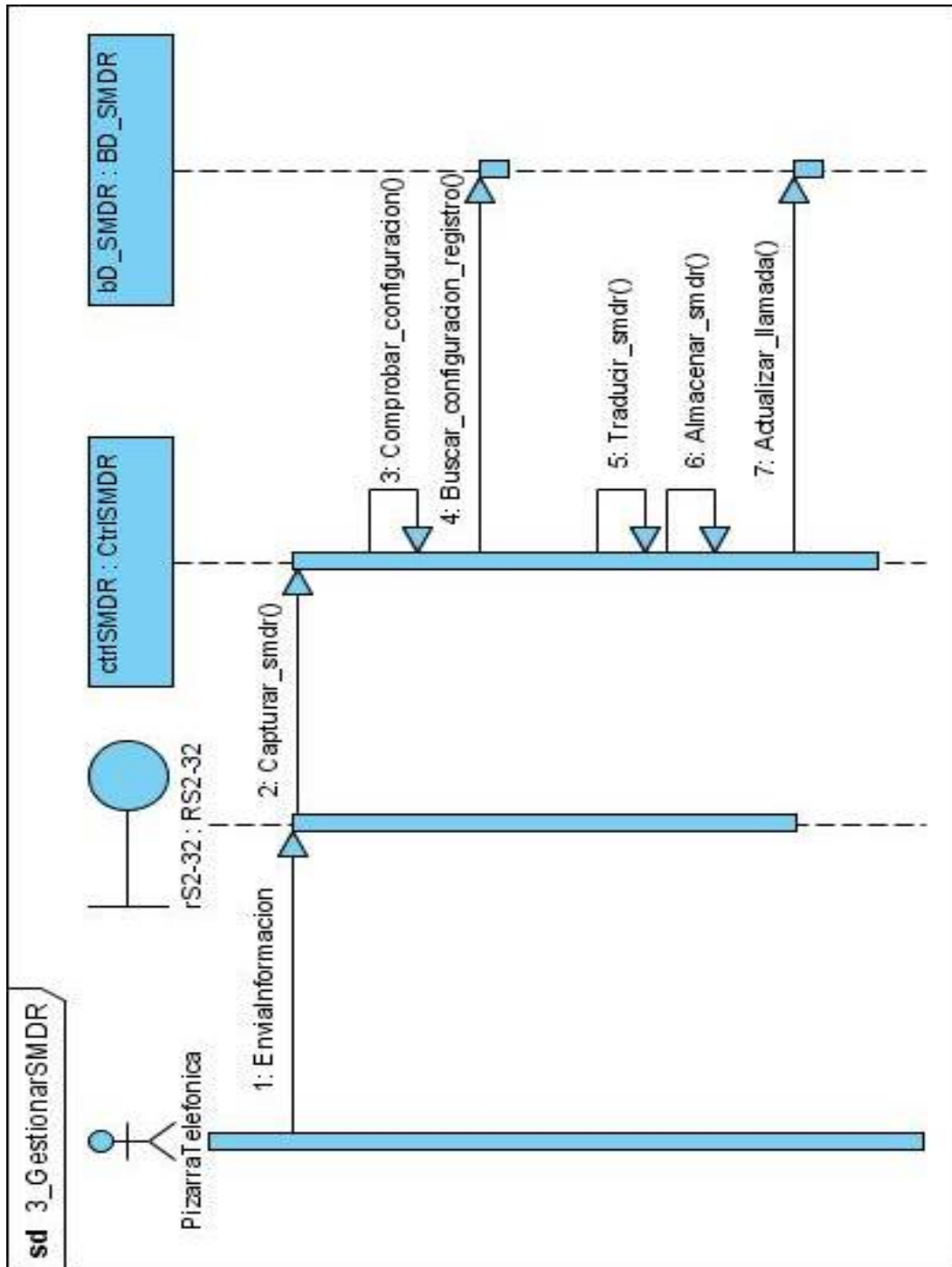


Fig 4.4 Diagrama de secuencia del diseño, Gestionar SMDR.

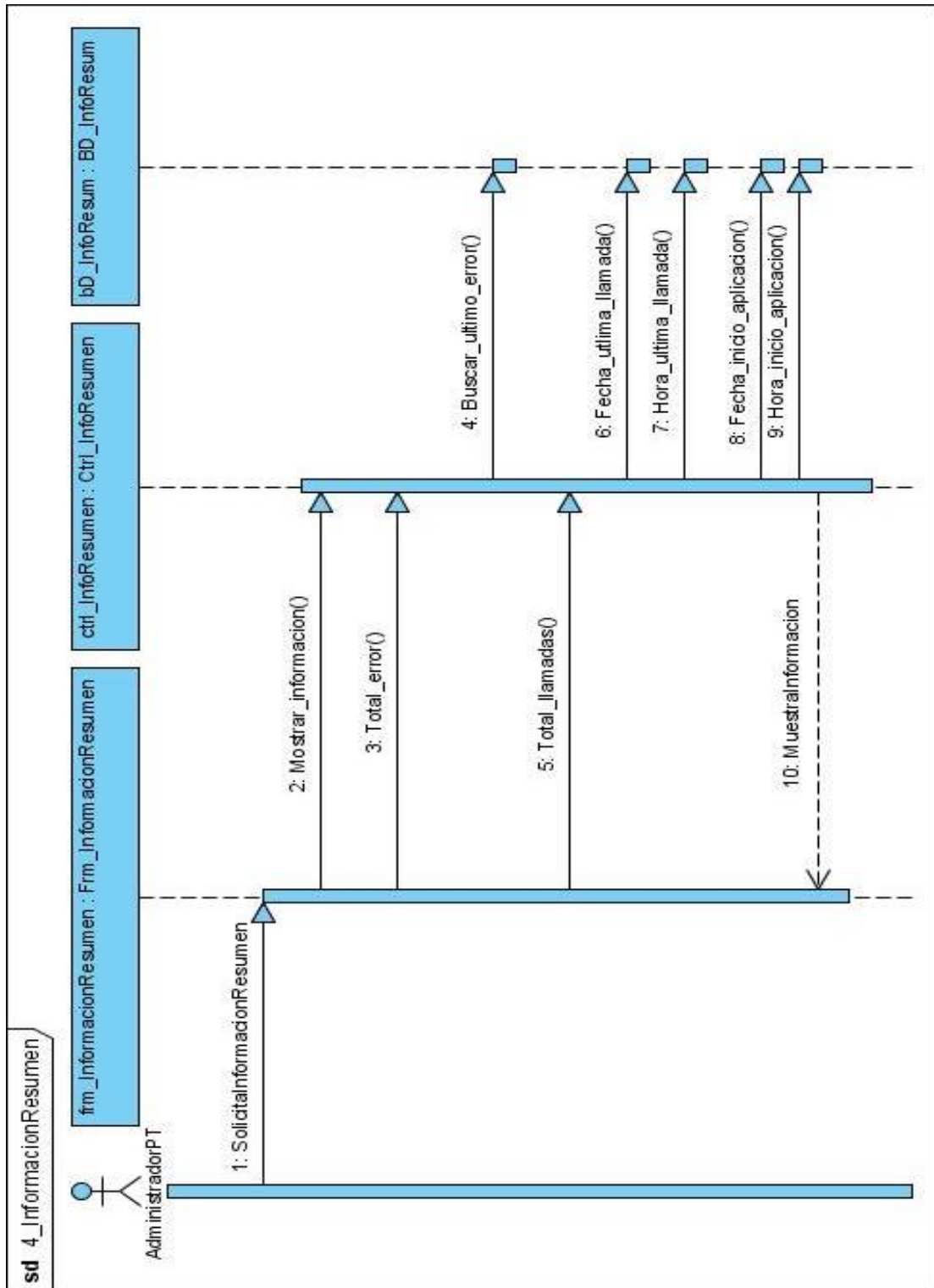


Fig 4.5 Diagrama de secuencia del diseño, Mostrar información resumen.

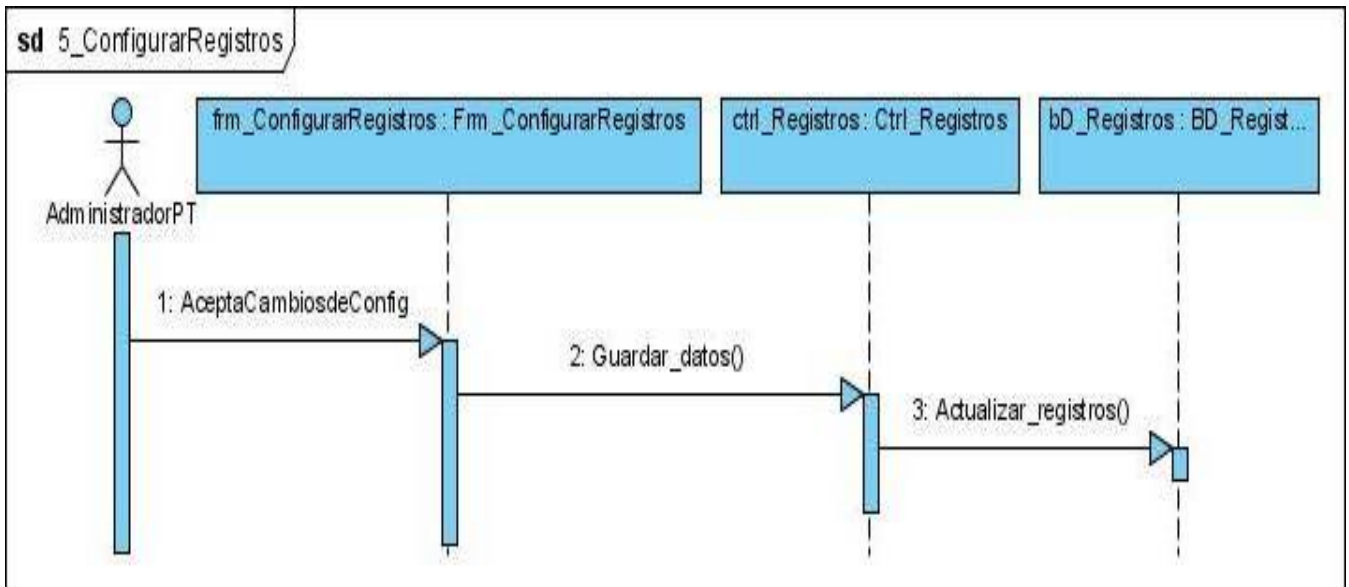


Fig 4.6 Diagrama de secuencia del diseño, Configurar registro.

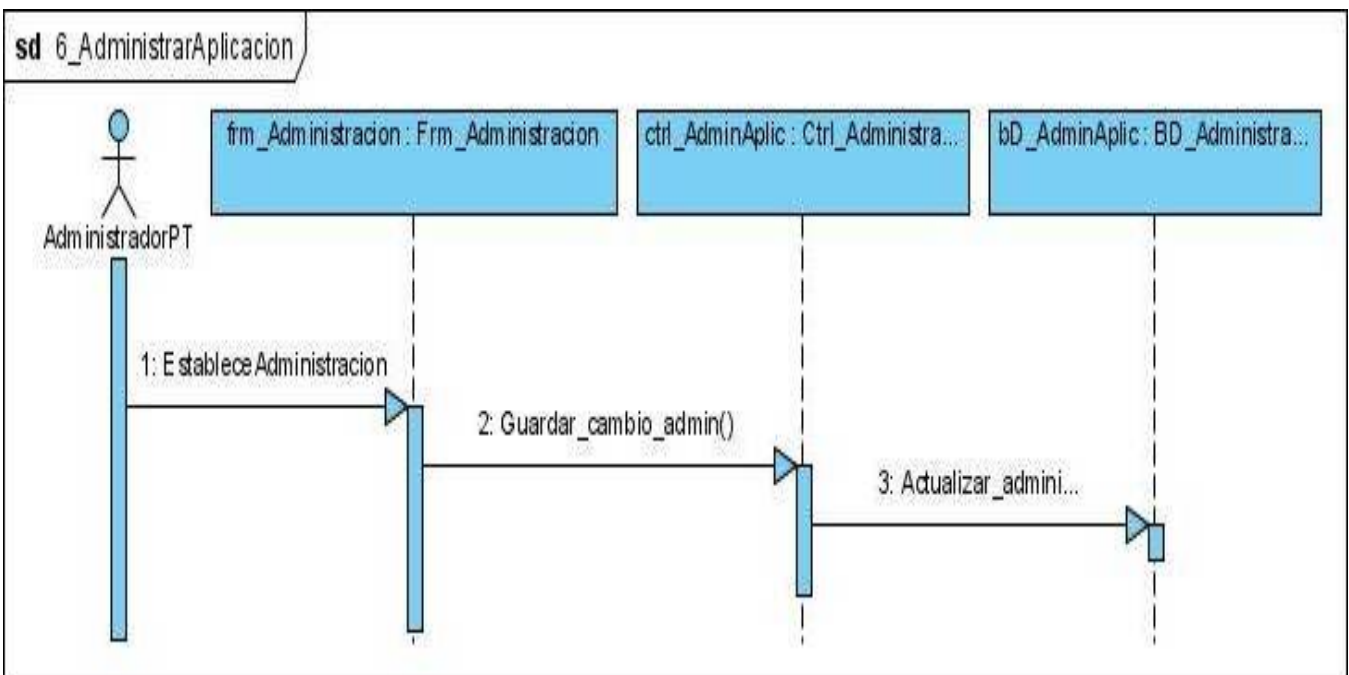


Fig 4.7 Diagrama de secuencia del diseño, Administrar aplicación.

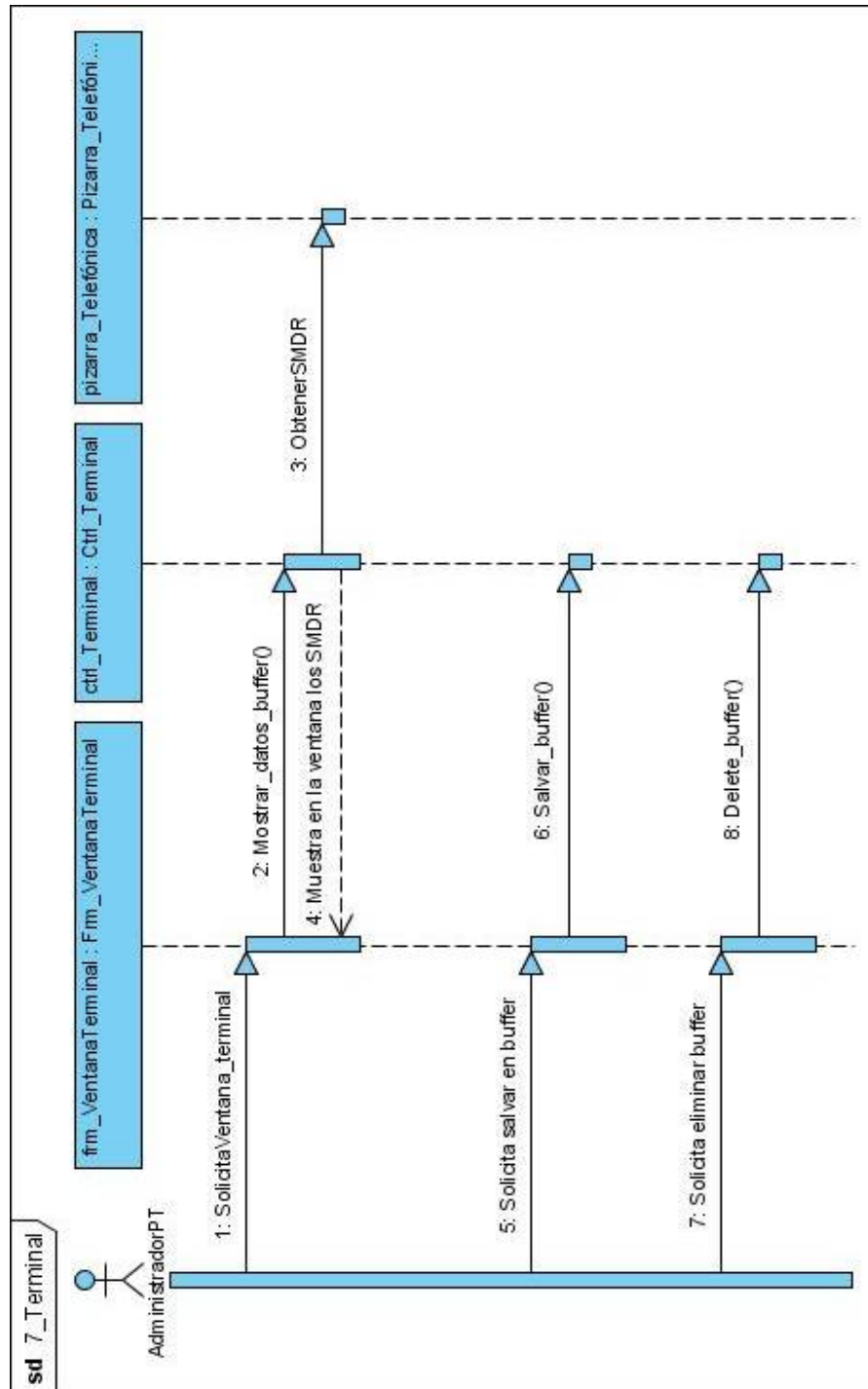


Fig 4.8 Diagrama de secuencia del diseño, Mostrar ventana Terminal.

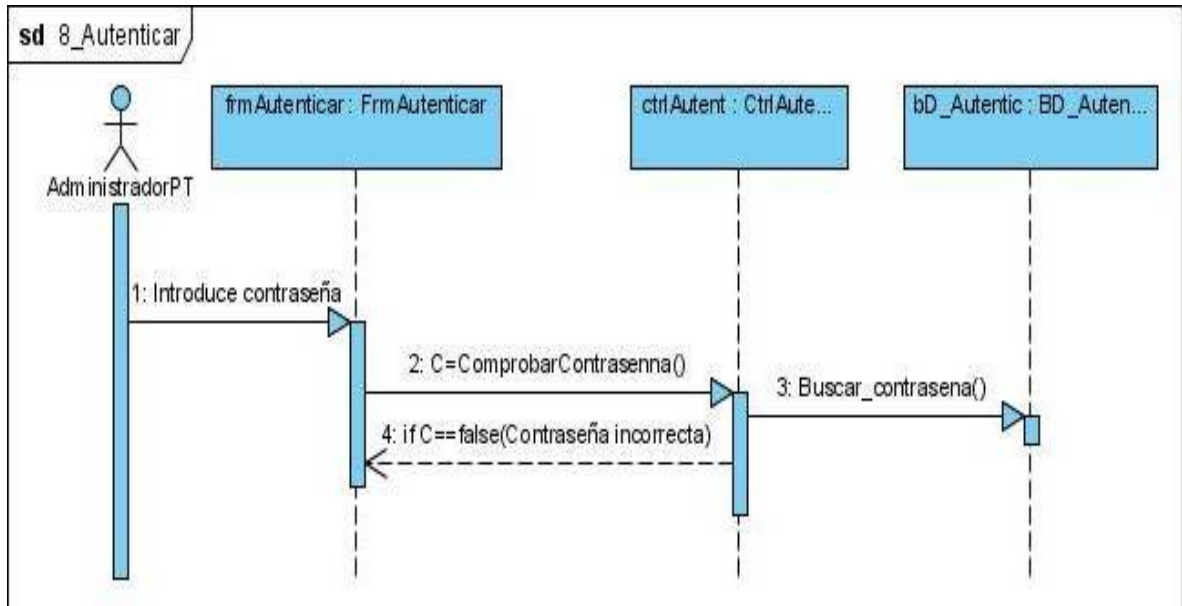


Fig 4.9 Diagrama de secuencia del diseño, Autenticar.

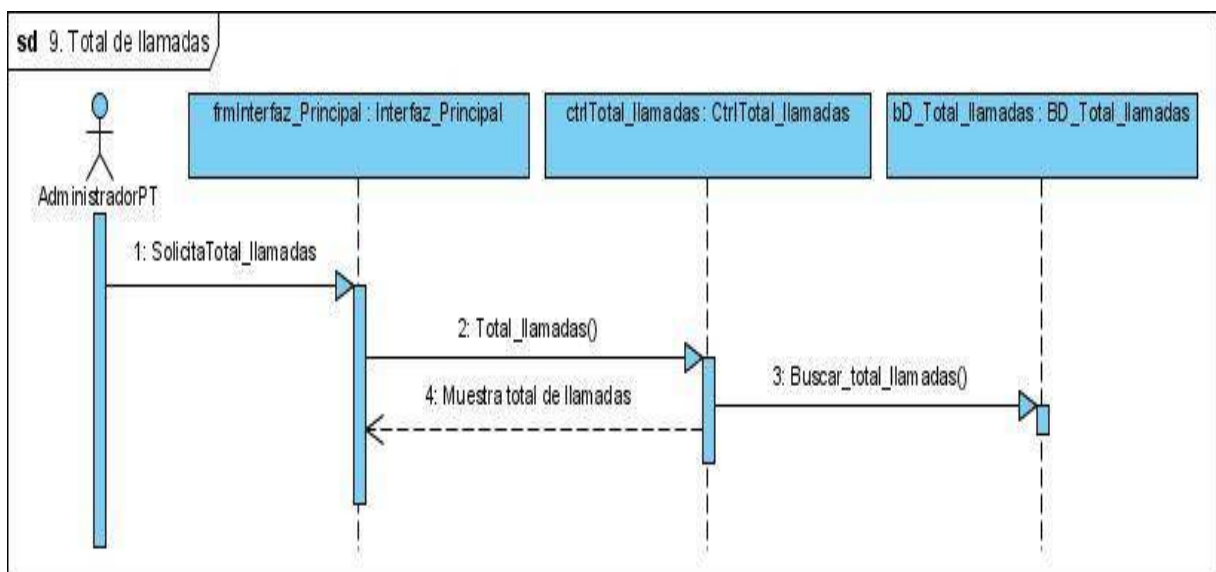


Fig 4.10 Diagrama de secuencia del diseño, Autenticar.

1 GLOSARIO

- 2 **PBX:** (Private Branch Exchange) - Intercambiador de redes privadas.
- 3 **SMDR:** (Station Message Detail Recording) -Registro generado por la pizarra telefónica.
- 4 **PABX:** (Private Automatic Branch Exchange) -Intercambiador automático de redes privadas.
- 5 **PMBX:** (Private Manual Branch Exchange) -Intercambiador manual de redes privadas.
- 6 **DND:** (Do Not Disturb) -No molestar.
- 7 **VoIP:** (Voice Over Internet Protocol) -permiten la conmutación de voz vía Internet
- 8 **RTC:** Red de Telefonía Local Pública.
- 9 **ISDN:** (Integrated Services Digital Net) -Red Digital de Servicios Integrados.
- 10 **UML:** (Unified Modeling Language) -Lenguaje unificado de modelado.
- 11 **SGBD:** Sistemas de gestión de base de datos.
- 12 **ORDBMS:** (Object-Relational Database Management System) –Sistema gestor de Base de datos
- 13 objeto-relacional.
- 14 **ODBMS:** (Object Database Management System) –Sistema gestor de Base de datos orientado a
- 15 objeto.
- 16 **RDBMS:** (Relational Database Management System) -Sistema gestor de Base de datos relacional.
- 17 **Linux:** Sistema operativo.
- 18 **Módem:** Dispositivo que permite poner en contacto dos ordenadores u otros dispositivos.
- 19 **J2EE:**(Java 2 Enterprise Edition) -La edición empresarial del paquete Java, comprenden un conjunto
- 20 de especificaciones y funcionalidades orientadas al desarrollo de aplicaciones empresariales.
- 21 **J2SE:**(Java 2 Standard Edition) -Plataforma de desarrollo, contiene colecciones de APIs utilizada por
- 22 cualquier programa que utilice Java.
- 23 **API:** (Application Programming Interface) -Interfaz de Programación de Aplicaciones.
- 24 **Links:** Vínculos.
- 25 **MVCC:** (Multi-Version Concurrency Control) -Control de Concurrencia Multi-versión.