

**SISTEMA DE GESTION DE CONFIGURACION DE SERVICIOS
SUPLEMENTARIOS DE TELEFONIA FIJA POR INTERNET**

**NOLBERTO DEVIA ESPINOSA
GLENN WARD BOLÍVAR**

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERIA ELECTRONICA Y
TELECOMUNICACIONES
POPAYÁN, MAYO DE 2002**

**SISTEMA DE GESTION DE CONFIGURACION DE SERVICIOS
SUPLEMENTARIOS DE TELEFONIA FIJA POR INTERNET**

Autores:

**NOLBERTO DEVIA ESPINOSA
GLENN WARD BOLÍVAR**

**Trabajo de Grado para optar al título de
Ingeniero en Electrónica y Telecomunicaciones**

Director:

Dr. ÁLVARO RENDÓN GALLÓN

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERIA ELECTRONICA Y
TELECOMUNICACIONES
POPAYÁN, MAYO DE 2002**

TABLA DE CONTENIDO

1. INTRODUCCION.....	1
1.1 SISTEMA DE GESTIÓN DE CONFIGURACIÓN DE SERVICIOS.....	1
1.2. EL MODELO DE GESTION	4
1.3. LOS CASOS DE USO.....	5
1.4 ORGANIZACIÓN DE LA MONOGRAFIA.	6
2. DESCRIPCION DE LAS TECNOLOGÍAS	8
2.1 EL PROTOCOLO SNMP.....	8
2.1.1 Corta Reseña Histórica.....	8
2.1.2 Generalidades de SNMP.	9
2.1.3 Protocolo SNMP.	10
2.2. MODULO DE INFORMACIÓN DE GESTION MIB.....	12
2.2.1 Estructura de Información de Gestión SMI.....	13
2.2.2 Módulos de Información	14
2.2.3 Principales Estructuras de los Módulos de Información	14
2.2.4 Los Objetos en la MIB	16
2.3. LENGUAJE DE PROGRAMACION PHP.	19
2.3.1 Reseña Histórica.	21
2.3.2 Estructuras importantes en PHP.	21
2.3.3 Seguridad en PHP	23
2.4 SISTEMA DE ADMINISTRACION TELEFONICA SATplus.....	23
2.4.1 Características Generales del SATplus.....	24
2.4.2 Servicio de consulta de peticiones de Servicios Suplementarios.	25
2.4.3 Base de datos del SATplus	25
3. DISEÑO	30
3.1 DIAGRAMA DE IMPLANTACION.....	31
3.2 DIAGRAMAS DE SUBSISTEMAS E INTERFACES	32
3.3 DIAGRAMAS DE CLASES POR PAQUETES.....	34
3.4 BASE DE DATOS DE USUARIOS	40
3.4.1 Tabla Registro de Usuarios.	40
3.4.2 Tabla Gestión de Administradores.	40
3.5 DESCRIPCIÓN DE LA MIB	41

3.5.1	Árbol de Información de la MIB	41
3.5.2	Relaciones entre las tablas de la MIB	44
3.5.3	Estructura de datos de la MIB	45
3.5.4	Consulta por el Agente SNMP (Ejemplo)	46
3.6.	DIAGRAMAS DE SECUENCIA.....	47
3.6.1	Diagrama de Secuencia - Adicionar Usuario.....	47
3.6.2	Diagrama de Secuencia - Validar Usuario Web.....	49
3.6.3	Diagrama de Secuencia - Activar Servicio (Código secreto).....	51
3.6.4	Diagrama de Secuencia - Modificar servicio (Transferencia de llamada)	53
4.	MODULO DE GESTIÓN SNMP	55
4.1	DESCRIPCION DEL AGENTE SNMP.....	56
4.1.1	Características Importantes.....	56
4.1.2	Operaciones del Agente SNMP-SGCST.....	57
4.1.3	Generalidades de la Codificación.....	58
4.1.4	Descripción del código del Agente SNMP	59
4.1.5	Generación de la TGST.DLL.....	62
4.2	DESCRIPCION DE LA MIB.....	63
4.2.1	Características Importantes.....	63
4.2.2	Las Tablas de la MIB.....	64
4.2.3	El Árbol de la MIB.....	65
4.2.4	Descripción del Código.....	66
4.3	APLICACIÓN UPDOCI	68
4.4	TABLA DE PETICIÓN SERVICIOS SUPLEMENTARIOS.....	70
4.4.1	Parámetros que registra UPDOCI	71
5.	MODULO DE APLICACIÓN	72
5.1	LA ESTACIÓN DE GESTIÓN	72
5.1.1	La Estación de Gestión del Sistema SGCST	73
5.1.2	Conexión a la base de datos de Usuarios.....	75
5.2	MODELO DE APLICACIÓN PARTE USUARIO.....	76
5.2.1	Árbol de interfaces del Usuario.....	76
5.2.2	Descripción de Páginas	76
5.2.3	Casos Posibles de Usuario ser.php.....	79
5.3	MODELO DE APLICACIÓN PARTE USUARIO ADMINISTRADOR.....	81
5.3.1	Árbol de interfaces del Administrador	81

5.3.2 Descripción de Páginas	82
5.3.3 Casos Posibles	83
6. RECOMENDACIONES Y CONCLUSIONES	86
BIBLIOGRAFIA.....	93

TABLA DE FIGURAS

FIGURA 1.1: SISTEMA DE GESTIÓN SGCST.	2
FIGURA 1.2: DIAGRAMA DESCRIPTIVO DEL SISTEMA	3
FIGURA 1.3: MODELO DE GESTIÓN DEL SISTEMA	4
FIGURA 1.4: CASOS DE USO DEL SISTEMA SGCST	5
FIGURA 2.1: LA ARQUITECTURA DE COMUNICACIONES SNMP	10
FIGURA 2.2: MODELO DE GESTIÓN DEL AGENTE SNMP	11
FIGURA 2.3: MODELO DE GESTIÓN DE INFORMACIÓN	14
FIGURA 2.4: ESQUEMA DE FUNCIONAMIENTO EN PHP	20
FIGURA 2.5: EJEMPLO DE UN FORMULARIO DEL SATPLUS	25
FIGURA 3.1: ÁRBOL DE LA MIB - GST	43
FIGURA 3.2: ESTRUCTURA DE LOS DATOS EN LA MIB	45
FIGURA 3.3: RELACIÓN ENTRE LAS TABLAS DE LA MIB	46
FIGURA 4.1: MÓDULO DE GESTIÓN SNMP	56
FIGURA 4.2: INVOCACIÓN DE UPDOCI POR EL AGENTE SNMP	58
FIGURA 5.1: ESQUEMA DE UNA ESTACIÓN DE GESTIÓN	73
FIGURA 5.2: ESTACIÓN DE GESTIÓN SGCST	74
FIGURA 5.3: DIAGRAMA DE INTERFACES PARA EL USUARIO	76
FIGURA 5.4: PÁGINA DE GESTIÓN DE SERVICIOS	79
FIGURA 5.5: DIAGRAMA DE INTERFACES PARA EL ADMINISTRADOR	81
FIGURA 5.6: PÁGINA DE GESTIÓN DE USUARIOS	82

TABLAS ESPECIALES

TABLA 1: GESTIÓN DE USUARIOS	40
TABLA 2: GESTIÓN DE ADMINISTRADORES	41
TABLA 3: OID'S DE LA MIB DEL SISTEMA SGCST	44
TABLA 4: PARÁMETROS DE LOS OBJETOS DEL SISTEMA SGCST	45
TABLA 5: FUNCIONES API EXTENDIDAS DE SNMP	62
TABLA 6: PETICION DE SERVICIOS SUPLEMENTARIOS	70
TABLA 7: RFC'S RELACIONADOS CON SNMP V1	75

ANEXOS

ANEXO A: CÓDIGOS DEL AGENTE SISTEMA SGCST	
1. ARCHIVO TGSTMIB.C	
2. ARCHIVO TGSTMIB.H	
3. ARCHIVO TGSTDLL.C	
4. ARCHIVO TGSTDLL.DEF	
5. ARCHIVO MAKEFILE	
ANEXO B: ARCHIVO MIB	
1. ARCHIVO TGST.MIB	
ANEXO C: INTERFAZ SATplus	
1. APLICACIÓN UPDOCI	
ANEXO D. ARCHIVOS EN PHP	
1. ADICIONAR USUARIO.PHP	
2. ELIMINAR USUARIO.PHP	
3. AD SERVICIOS.PHP	
4. MD SERVICIO.PHP	
5. SERVICIO.PHP	
6. VALIDAR ADMINISTRADOR.PHP	
7. VALIDAR USUARIO.PHP	
8. ACCESAR ORACLE.PHP	
9. USUARIOSERV.HTML	

1. INTRODUCCION

En busca de una mayor competitividad las empresas de telecomunicaciones procuran con tecnología de punta brindarles a sus clientes nuevos y mejores servicios, muchas empresas aprovechan la amplia acogida obtenida por la red Internet para brindar un paquete de servicios mucho más amplio y satisfacer las nuevas necesidades de sus clientes. Es indispensable para lograr una satisfacción completa, velar por que los servicios se presten con la mejor calidad posible.

Para que un servicio sea de calidad no basta con satisfacer completamente al cliente, es necesario gestionar eficientemente toda la red empresarial administrando los recursos que la componen en términos de rendimiento, capacidad, trafico, configuración, mantenimiento, planificación etc. Pero la gestión no puede quedarse en términos solo de la red, debe extenderse a la gestión de los servicios y de los usuarios para alcanzar una gestión integral.

1.1 SISTEMA DE GESTIÓN DE CONFIGURACIÓN DE SERVICIOS

Este proyecto desarrolla una aplicación denominada SGCST soportada en la Web o red Internet. Este sistema les permite a los usuarios o clientes de las operadoras telefónicas configurar desde su PC conectado a Internet los aspectos más importantes de los servicios suplementarios que estas empresas les brindan, (Ver Figura 1.1)

Este sistema tiene la capacidad de mejorar la calidad de los servicios que ofrece la empresa telefónica a sus suscriptores, incrementar la eficiencia de la gestión en la información que maneja la empresa, mejorar la comunicación al interior de la empresa entre las diferentes dependencias, reducir costos de operación e incrementar calidad en la atención a los clientes.

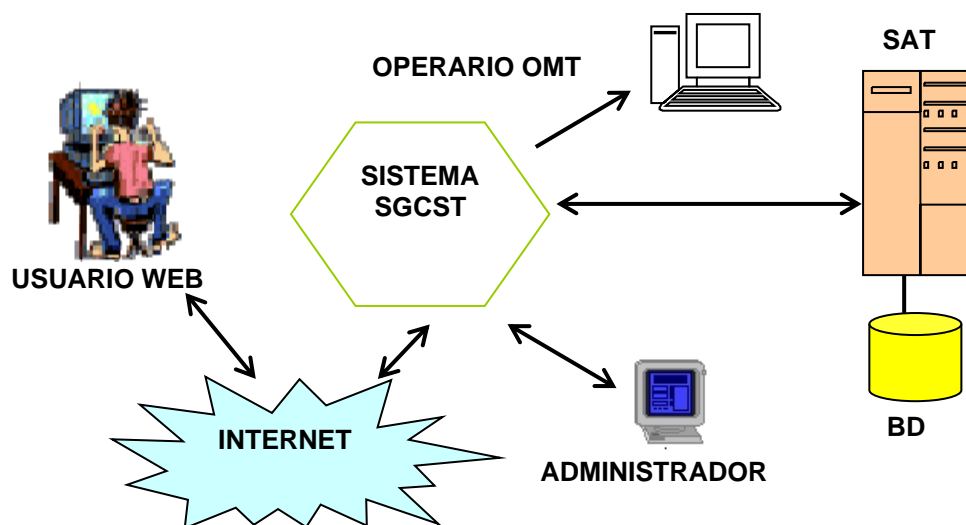


Figura 1.1: Sistema de Gestión SGCST.

En la Figura 1.1 se muestra el Sistema de Gestión de Servicios Suplementarios por Internet. En este dibujo se destacan los elementos con los que el sistema interactúa, son: el usuario Web que se conecta al sistema desde la red Internet, el Administrador del sistema, el operario de la terminal de operación y mantenimiento OMT, ambos conectados desde la LAN de la empresa, y finalmente el servidor donde se encuentran la aplicación SATplus o *Servicio de Administración Telefónica o de Telecomunicaciones*, también conectados a la LAN.

En este proyecto se desarrolla una aplicación que permite gestionar un sistema de servicios telefónicos suplementarios, el cual tiene como soporte la red Internet, se utiliza como protocolo de gestión de red a SNMP o protocolo simple de gestión de red. SNMP goza de gran aceptación en la gestión dentro de la red Internet por su gran compatibilidad con TCP/ IP o protocolo básico de Internet.

Este proyecto reviste de gran importancia para el Departamento de Conmutación y en particular al grupo de Ingeniería Telemática. Esta aplicación permite avanzar en la línea de investigación y desarrollo en sistemas de gestión basados en protocolos de gestión de red como SNMP, abre el camino para futuros proyectos que permitan aprovechar los logros que se obtengan con la realización de este

trabajo y sin duda el producto a obtener cuenta con el interés de las operadoras regionales de telefonía fija, como es el caso de TELEARMENIA, empresa que oporto su valiosa colaboración en las etapas finales del proyecto.

La aplicación que se desarrollo se enmarca dentro del proyecto GETWEB (Gestión integrada de Telecomunicaciones basada en la Web) y retoma diversos trabajos que ha realizado el Grupo de Ingeniería Telemática en temas como el acceso a bases de datos a través de Internet, la gestión basada en CORBA y la Web, y la operación y mantenimiento de centrales telefónicas. La puesta a punto de aplicaciones como la desarrollada en este trabajo, permiten hacer gestión de redes y servicios de telecomunicaciones que cumplan con las especificaciones de los protocolos SNMP, CMIP y DMI, objetivo importante que busca GETWEB.

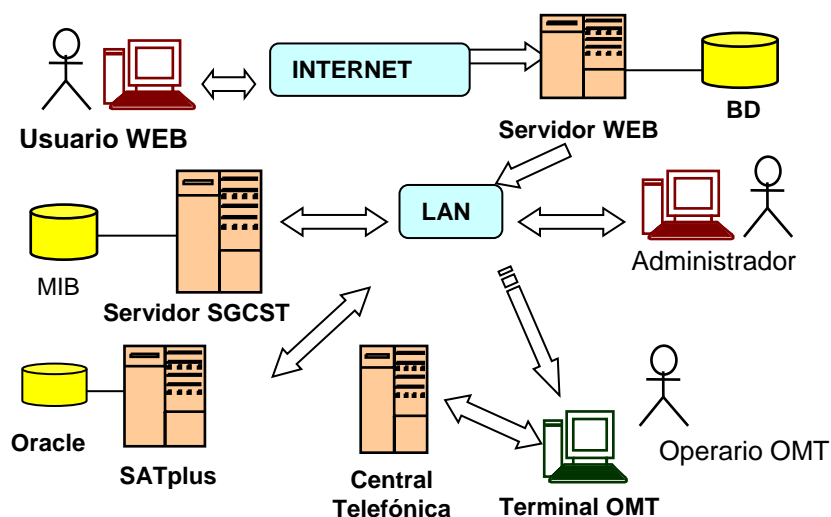


Figura 1.2: Diagrama descriptivo del sistema

El sistema esta soportado por interfaces amigables desarrolladas en páginas Web HTML alojadas en servidor WEB, por comodidad si se desarrolla una aplicación para alguna empresa se tomaría en cuenta el mismo sitio Web de la empresa, realizando un enlace al servidor SGCST, donde se realizaría la gestión de los servicios (ver Figura 1.2)

1.2. EL MODELO DE GESTION

En la Figura 1.3 se puede observar el modelo de gestión del sistema. En el se encuentran cada uno de los módulos del sistema, en este grafico se destacan la parte de gestión con el agente SNMP, la base de datos de gestión MIB donde se almacenan todos los objetos de gestión del sistema, por parte del protocolo SNMP y los actores o usuarios del sistema.

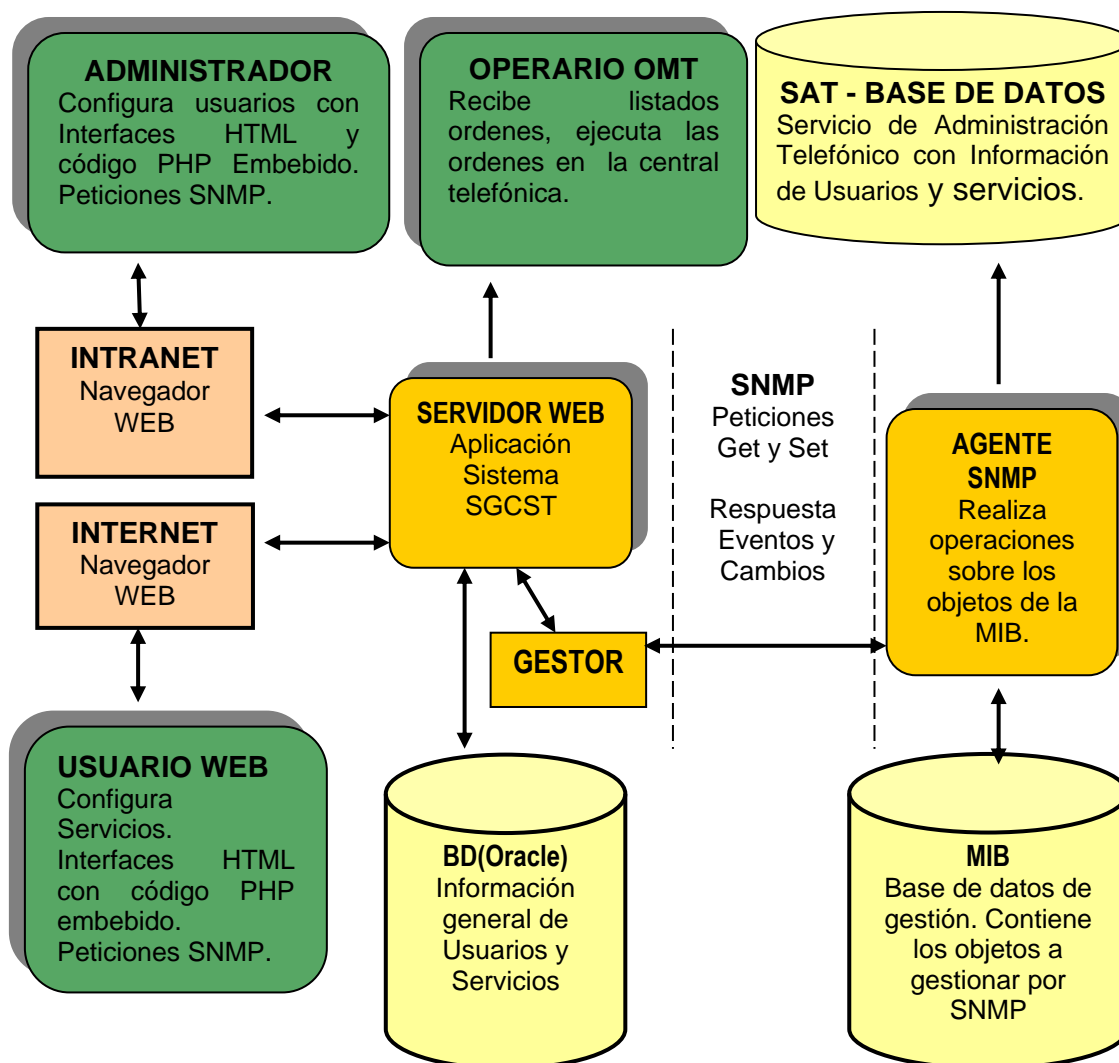


Figura 1.3: Modelo de Gestión del Sistema

1.3. LOS CASOS DE USO.

Los casos de uso se obtuvieron del estudio realizado en el documento de análisis del dominio, siguiendo las recomendaciones del R.U.P. o Proceso Unificado de Rational. En este análisis del dominio se busca definir el ambiente externo del sistema que se pretende desarrollar en el proyecto, es decir el entorno o contexto donde el sistema se desenvuelve.

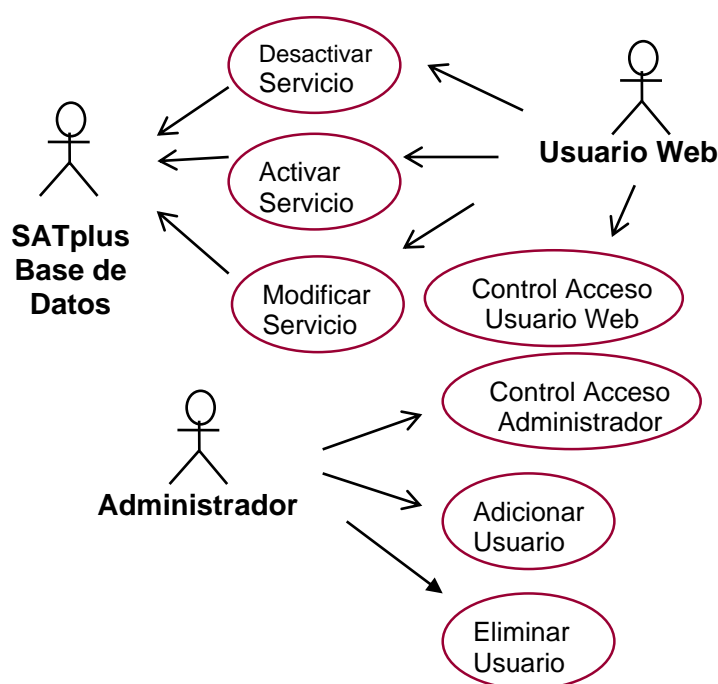


Figura 1.4: Casos de Uso del sistema SGCST

En la Figura 1.4 están los casos de uso, que son las posibles acciones que pueden realizar los usuarios o actores en el sistema SGCST. Se observan dos tipos de usuarios el primero es el **Usuario Web**, que son los clientes del sistema SGCST que se conectan desde Internet, luego se observa el **Usuario Administrador** que se conectan al sistema SGCST desde la LAN interna de la empresa telefónica.

Por ultimo se encuentra el SATplus o Sistema de Administración Telefónica o Telecomunicaciones. En la Base de datos del SATplus se almacenan los cambios realizados en la configuración de los servicios por parte de los Usuarios Web para efectos de tarificación y de control de tipo administrativo.

1.4 ORGANIZACIÓN DE LA MONOGRAFIA.

A continuación se describen de forma breve cada uno de los capítulos que componen este trabajo de grado:

Capítulo 1:

Titulo: “**INTRODUCCIÓN**”

Ilustra de manera breve el contenido del proyecto, haciendo énfasis en la importancia del mismo, muestra gráficamente el sistema y el modelo de gestión del mismo.

Capítulo 2:

Titulo: “**DESCRIPCION DE LAS TECNOLOGIAS**”

Abarca de forma amplia y con mucha claridad, todos los conceptos relevantes empleados en el sistema de gestión de servicios telefónicos suplementarios SGCST. Este capítulo contiene los siguientes temas:

- El Protocolo **SNMP**: Se hace una reseña histórica y un planteamiento de sus características más importantes.
- La base de información de gestión **MIB**: Se realiza un breve análisis del concepto dentro de SIM y una descripción de las principales estructuras SIM utilizadas en el proyecto.
- Lenguaje de programación **PHP**: Se tratan los aspectos más importantes de este lenguaje para el proyecto.
- El sistema de administración Telefónico SATplus: Se hace un estudio de este sistema, además se aborda el concepto de SQL.

Capítulo 3:**Título: “DISEÑO”**

Este Capítulo contiene el documento de diseño del sistema, que es muy útil para comprender mejor la aplicación del sistema SGCST. Contiene los diagramas de Implantación, Subsistemas, Clases y de Secuencias del sistema.

Capítulo 4:**Título: “MODULO DE GESTION SNMP”**

En este capítulo comprende la parte de gestión SNMP, que involucran al Agente SNMP, la base de información de gestión MIB y se complementa con un análisis de la aplicación UPDOCI que permite establecer una comunicación con la base de datos y es invocada por el agente SNMP:

- Análisis del Agente SNMP del sistema SGCST.
- Análisis de la MIB del sistema SGCST.
- Aplicación UPDOCI.

Capítulo 5:**Título: “MODULO DE APLICACIÓN”**

Este capítulo lo componen los siguientes temas:

- Estación de gestión sistema SGCST.
- Modelo de la aplicación para el Usuario Web.
- Modelo de la aplicación para el Usuario Administrador.

Capítulo 6:**Título: “RECOMENDACIONES Y CONCLUSIONES”**

Contiene las conclusiones y recomendaciones producto de la experiencia de trabajar en el desarrollo del proyecto, que servirá de fundamento y consulta para futuras aplicaciones en los temas tratados.

2. DESCRIPCION DE LAS TECNOLOGÍAS

Es importante hacer una referencia clara y sencilla de cada una de las tecnologías que hacen parte del sistema de gestión de servicios telefónicos, se busca con este capítulo ampliar los conceptos de interés a esta monografía, tomando en cuenta los avances recientes en el mundo de las telecomunicaciones.

2.1 EL PROTOCOLO SNMP.

2.1.1 Corta Reseña Histórica.

El organismo que administra y regula la red Internet perteneciente a la UIT encargó en 1987 a un grupo técnico (El encargado de encontrar soluciones a los problemas técnicos que plantea el funcionamiento de la red), encontrar una solución de gestión integrada para Internet.

Este grupo técnico propuso una solución en dos fases:

- Utilizar un único protocolo capaz de ser entendido por todos los dispositivos de la red Internet.
- Posteriormente cuando estuviesen desarrolladas las normas OSI, utilizar los protocolos de gestión OSI soportados sobre la plataforma de comunicaciones de Internet. Esta solución se conoce como CMOT (CMIP sobre TCP/IP)

En 1988 se implementó y comenzó a utilizarse un protocolo de gestión denominado SNMP (Simple Network Management Protocol), un protocolo sencillo para la gestión de red. Sin embargo, la segunda estrategia CMOT (CMIP sobre TCP/IP) no ha avanzado suficiente, debido quizá a la falta de estabilidad de las normas OSI de gestión por lo que muy pocos fabricantes ofrecen esta solución para Gestión. La última implementación del protocolo SNMP es la norma SNMP v3, actualmente disponible en el mercado.

El protocolo SNMP procede del protocolo SGMP *Simple Gateway Management Protocol*, un Protocolo Sencillo para Gestión de Ordenadores que efectúan el encaminamiento de los datagramas IP en Internet. El protocolo SNMP fue desarrollado por los mismos autores que el protocolo SGMP. Este grupo de personas tuvieron una visión práctica de las redes y desarrollaron el protocolo en pocos meses.

2.1.2 Generalidades de SNMP.

El protocolo SNMP es similar al protocolo CMIP de OSI, proporciona una forma de comunicar información de gestión entre procesos gestores y procesos agentes. La Base de Información de Gestión MIB contiene los recursos de comunicaciones según el modelo de estructura de información de gestión.

La gestión de red SNMP está formada por un conjunto de estaciones o servidores de gestión de red y un conjunto de elementos de red. Las estaciones de gestión de red ejecutan las aplicaciones de gestión que monitorizan y controlan los elementos de red. Los elementos de red son dispositivos tales como Ordenadores Centrales, Pasarelas (Sistemas Intermedios o Gateways en terminología de Internet), Concentradores de Terminales, etc. Estos elementos de red disponen de procesos agentes responsables de la realización de las actividades de gestión.

Al igual que en la Gestión OSI, se utiliza la representación de los recursos como objetos. El acceso a los Objetos Gestionados se realiza a través de la MIB. Los objetos en la MIB se definen mediante ASN.1. Cada tipo de objeto tiene un nombre, unos atributos y una sintaxis. El nombre se asigna de forma universal mediante un identificador de objeto.

La sintaxis de un tipo de objeto define la estructura abstracta de datos correspondiente al tipo de objeto en particular. Se utiliza ASN.1 pero más bien un subconjunto tanto de los tipos como de las reglas de codificación para reducir la complejidad. Los nombres se utilizan para identificar los objetos gestionados y tienen una naturaleza jerárquica. El identificador de objeto permite distinguir los objetos y consiste en una secuencia de enteros que recorre un árbol global.

2.1.3 Protocolo SNMP.

SNMP es un protocolo desarrollado para la gestión de redes basadas en protocolos TCP/IP. El modelo de comunicaciones es similar al de Gestión OSI, existen un proceso Gestor y un proceso Agente que intercambian información de gestión mediante el protocolo SNMP. En la **Figura 2.1** se muestran la arquitectura de comunicaciones de gestión en entornos SNMP.

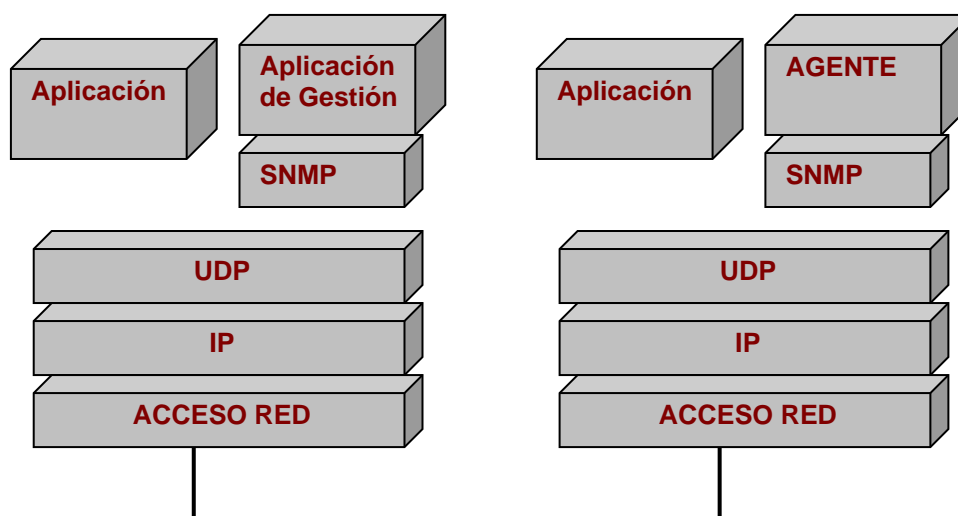


Figura 2.1: La arquitectura de comunicaciones SNMP

Las unidades de datos del protocolo SNMP se transportan utilizando los servicios proporcionados por el protocolo UDP (User Datagram Protocol). Como se sabe, este protocolo proporciona un servicio no orientado a conexión. UDP requiere menos recursos que TCP (Transmisión Control Protocol), puesto que las unidades de datos no se envían en el contexto de una conexión, ni ha que garantizar la entrega fiable, por lo que la implementación es más sencilla.

De esta forma el desarrollo de los procesos agentes en dispositivos simples de comunicaciones, es más simple aún a costa de mayor dificultad en el desarrollo del proceso gestor que tiene que implementar una funcionalidad que en OSI es proporcionada por los niveles inferiores.

Los protocolos de gestión de red especifican las normas de comunicación entre un Gestor que envía comando a un Agente, el cual las ejecuta y devuelve las respuestas apropiadas (ver Figura 2.2). Cabe esperar que los potentes protocolos de gestión tengan gran cantidad de comandos diferentes para poder realizar un gran número de operaciones diferentes como: reinicializar el equipo, cambiar el encaminamiento, activar una línea, etc.

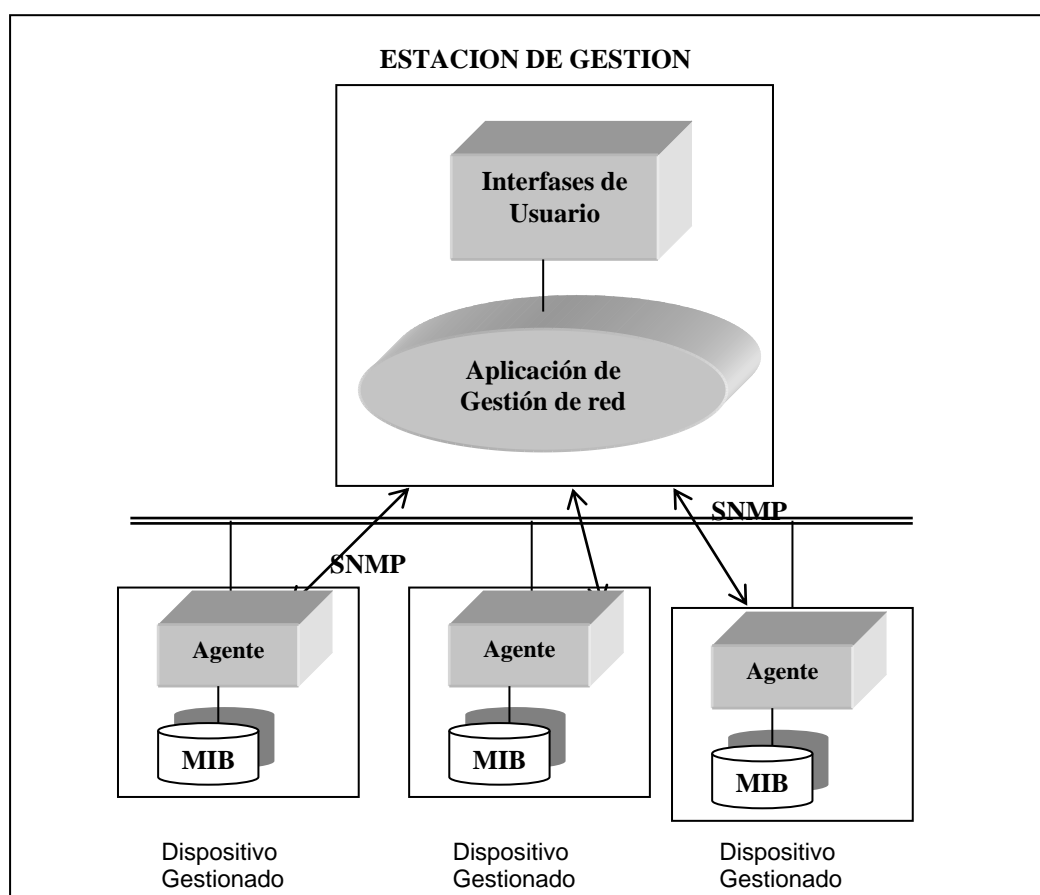


Figura 2.2: Modelo de gestión del Agente SNMP

El protocolo SNMP se ha diseñado de una forma radicalmente diferente. En vez de definir un gran número de comandos diferentes, se emplea la fórmula de tomar y almacenar (fetch-store) sobre el conjunto de valores definidos en la MIB. Por tanto, SNMP emplea únicamente dos comandos, uno para obtener información de variables, y otro para almacenar información en variables. Todas las demás operaciones se producen como efecto de estas dos.

Por ejemplo, para reinicializar un dispositivo, puede definirse una variable que contenga el tiempo que ha de transcurrir hasta el próximo reinicio. Si se permite al agente almacenar un cero en dicha variable, el efecto será el mismo que el de enviar un comando específico para reinicializar el dispositivo. La principal ventaja de esta técnica reside en la simplicidad y estabilidad del protocolo. Aunque se añadan nuevas variables a la MIB el protocolo SNMP permanecerá estable.

La estrategia que utiliza el SNMP es la interrogación periódica de las variables de la MIB por el Gestor y la notificación asíncrona de algunos sucesos por el Agente sin que hayan sido solicitados por el gestor. El protocolo SNMP dispone de alguna operación más de las dos básicas que hemos mencionado. Son las siguientes:

- GET: Obtiene el valor de objetos (variables MIB)
- GET-NEXT: Obtiene el valor de la variable adyacente (por el léxico)
- SET: Modifica el valor de una variable.
- GET-RESPONSE: Proporciona el resultado de las operaciones GET, GET-NEXT y SET.
- TRAP: Informe asíncrono de un evento.

Las operaciones de get, get-response y set permiten la lectura y escritura de variables. Las operaciones deben ser atómicas, es decir, si un único mensaje especifica operaciones en varias variables, el agente debe realizar todas las operaciones o ninguna de ellas. La operación trap permite a los agentes informar de los eventos que suceden en los dispositivos gestionados (fallos, alarmas, etc.)

2.2. MODULO DE INFORMACIÓN DE GESTION MIB

SNMP es un protocolo de gestión de red, esto es, un conjunto de estructuras y primitivas que permiten obtener y modificar datos o estados de elementos concretos tanto hardware como software en la red. El objetivo para esta parte del capítulo será realizar una descripción del sistema de información de gestión con énfasis en el módulo MIB, que es quien contiene los objetos que serán gestionados por SNMP.

De la MIB se analizan ciertas estructuras características que son del interés del proyecto y por considerarse útiles para una mejor comprensión de la aplicación SGCST.

2.2.1 Estructura de Información de Gestión SMI

SMI (*Structure of Management Information*) es un concepto innovador que plantea una metodología de coordinación de información para ayudar en la colaboración de los sistemas distribuidos y permitir el compartir información o recursos por clientes heterogéneos. El aporte más sobresaliente de SIM es el desarrollo del modelo "pull" o modelo de gestión de información distribuida. Los eventos pueden ser semánticamente descritos usando estados básicos mediante las técnicas de interacción para comunicar y reproducir los mensajes en tiempo real.

SMI define las reglas para definir la información de gestión independiente de los detalles de implementación. Si se piensa en una colección de objetos gestionables que están almacenados, por ejemplo, en una base de datos, la SIM define el esquema de esa base de datos. En realidad, esa base de datos se denomina **Base de Información de Gestión (MIB)**. (Ver modelo en la Figura 2.3)

La Base de Información de Gestión (*Management Information Base MIB*). Representa la información que se está utilizando, modificando o transfiriendo en la arquitectura de los protocolos de gestión OSI como es el caso de SNMP (ver Figura 2.3)

La MIB conoce todos los objetos gestionables y sus atributos. No es necesario que este centralizada físicamente en un lugar concreto, puede estar distribuida a través del sistema y en cada uno de sus niveles. Para lograr claridad hay que entender los conceptos y relaciones entre variables, objetos y tipos de objetos. Un **objeto gestionable** tiene asociado una sintaxis y una semántica de tipo abstracto, mientras que una **variable** es una instancia de un objeto particular; en este caso también se denomina instancia de un objeto.

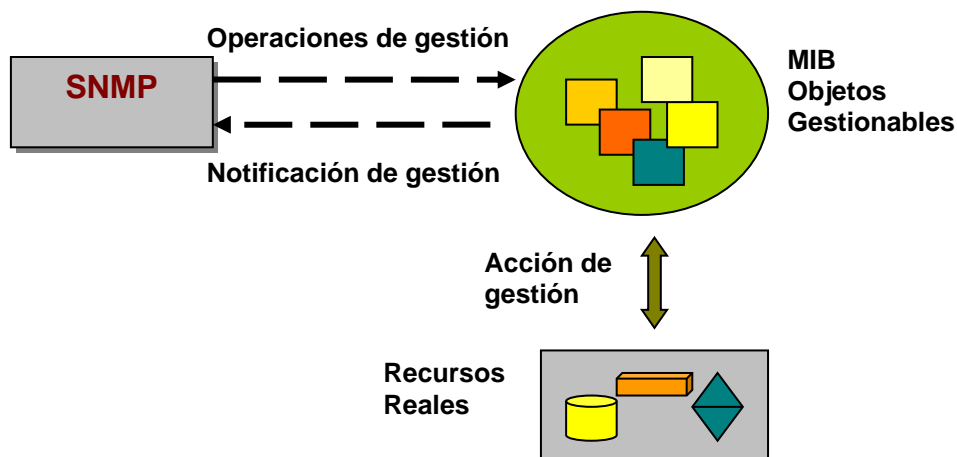


Figura 2.3: Modelo de Gestión de Información

2.2.2 Módulos de Información

Existen tres clases de módulos o tipos de estructuras de información dentro de las MIB que son definidas por el SMI:

- **Módulos MIB:** Define una colección de objetos de gestión con características afines con características muy similares y definidos por un nombre o identificador.
- **Sentencias de Conformidad:** Definen un conjunto de requisitos de los nodos con respecto a uno o más módulos MIB.
- **Sentencias de Capacidad:** Describen la capacidad de un nodo para implementar los objetos definidos en uno o más módulos MIB.

2.2.3 Principales Estructuras de los Módulos de Información

Identificadores del módulo de información

Cada módulo de información debe comenzar con la indicación de su identidad y la historia de sus revisiones. Para ello, la SIM define una macro especial, La macro MODULE-IDENTITY se usa para mantener los históricos de las revisiones de cada módulo de información. Debe existir una y sólo una en cada módulo de información.

Cláusula LAST-UPDATED

La cláusula LAST-UPDATED, que debe existir, contiene la fecha y hora de la última revisión realizada sobre este módulo de información.

Cláusula ORGANIZATION

La cláusula ORGANIZATION, que debe existir, contiene una descripción textual de la organización bajo cuyo patrocinio se desarrollo este módulo de información.

Cláusula CONTACT-INFO

La cláusula CONTACT-INFO, que debe estar presente, contiene el nombre, dirección postal, número del teléfono, y la dirección de correo electrónico de la persona a quien deben ser enviadas las preguntas de carácter técnico relacionadas con este módulo de información.

Cláusula DESCRIPTION

La cláusula DESCRIPTION, indispensable, contiene una descripción textual de alto nivel de los contenidos de este módulo de información.

Cláusula REVISION

La cláusula REVISION, que no es necesaria, es usada normalmente para describir las revisiones hechas al módulo de información, en orden inverso a su sucesión cronológica. Cada instancia de esta cláusula contiene la fecha y hora de la revisión.

Cláusula DESCRIPTION

La cláusula DESCRIPTION, que debe estar presente por cada cláusula REVISION, contiene una descripción textual de alto nivel de la revisión identificada en esa cláusula REVISION.

Valor de MODULE-IDENTITY

El valor devuelto de una llamada a la macro MODULE-IDENTITY es un identificador de objetos (OBJECT IDENTIFIER). Como tal, este valor puede usarse al referirse al módulo de información que contiene la llamada.

Ejemplo de uso

Ejemplo de cómo se construiría el esqueleto de un módulo MIB:

```
FIZBIN-MIB DEFINITIONS ::= BEGIN
IMPORTS
MODULE-IDENTITY, OBJECT-TYPE,
experimental
FROM SNMP v1-SMI;

    fizbin MODULE-IDENTITY
LAST-UPDATED "098-5124689"
ORGANIZATION "GETWEB "
CONTACT-INFO " Glenn Ward"
        A.A.: 13-46 Popayán.
        Colombia: Tel. 092-8237076
        E-mail: gward@atenea.ucauca.edu.co "
DESCRIPTION "Módulo MIB para entidades
SNMPV1."
REVISION "000001"
DESCRIPTION "Versión inicial de
este módulo MIB."
::= { snmpModules 1 }
END
```

2.2.4 Los Objetos en la MIB

Se definen objetos con características afines dentro de grupos de objetos (*object group*). Desde el punto de vista de la implementación, un grupo de objetos se ve como una unidad de implementación. La asignación de identificadores de objeto a los tipos en un módulo MIB, sigue la siguiente pauta:

- Los tipos de objeto se ponen dentro de grupos de objetos.
- Un identificador de objeto es asignado a cada grupo.
- Los objetos poseen un identificador secuencial subordinado al del grupo.

Identificadores de los Objetos.

Es necesario conocer el nombre de un objeto para poder gestionarlo, pero los objetos en sí solo son plantillas donde se especifican mediante indentificadores los parámetros y características de los mismos, y es sobre las instancias de los objetos donde el agente SNMP realiza las operaciones de gestión.

El agente SNMP utiliza para identificar las instancias, un identificador de objeto, formado por la concatenación del nombre, del tipo de objeto y un sufijo. Si el objeto no es parte de una tabla, entonces, representa una instancia de un tipo de objeto en un dispositivo particular. Como ilustración se incluyen a continuación los identificadores de los objetos que se utilizaron en la MIB de la Aplicación SGCST.

Cláusula MAX-ACCESS

La cláusula MAX-ACCESS, que debe estar presente, define si tiene privilegios para leer, escribir y / o crear una instancia del objeto. Éste es el máximo nivel de acceso para el objeto que es independiente de cualquier política administrativa de accesos. El valor "read-write" (lectura-escritura) indica que está permitida la lectura y la escritura, pero no la creación. Un valor "read-create" (lectura-creación) indica posibilidad de lectura, escritura y creación.

Cláusula DESCRIPTION

La cláusula DESCRIPTION, de carácter obligatorio, contiene una definición textual de ese objeto que mantiene todas las definiciones semánticas necesarias para la implementación, y debe incluir cualquier información que pudiera ser comunicada de otro modo en cualquier comentario asociado con el objeto.

Cláusula STATUS

La cláusula STATUS, que debe estar presente, indica si esta definición es actual o histórica.

Cláusula SINTAX

La cláusula SINTAX, que debe estar presente, define las estructuras de datos abstractas correspondientes a ese objeto.

Cláusula INDEX

La cláusula INDEX, que debe estar presente si ese objeto corresponde a una fila conceptual (a menos que una cláusula AUGMENTS se encuentre explícitamente presente), y debe estar ausente en cualquier otro caso, define información de identificación de instancia para los objetos subordinados a este objeto.

OBJECT IDENTIFIER

Un identificador de objeto es un nombre asignado arbitrariamente. Los identificadores de objeto se usan frecuentemente en el funcionamiento del protocolo, y existe una macro para asociar información textual con asignaciones de identificadores, que es la macro *OBJECT-IDENTITY*:

OBJECT-TYPE

El valor devuelto por una llamada a la macro OBJECT-TYPE es el nombre del objeto, que es un identificador de objeto (OBJECT IDENTIFIER), un nombre asignado administrativamente.

OBJECT-IDENTITY

El valor devuelto en una llamada a la macro OBJECT-IDENTITY es un identificador de objeto (OBJECT IDENTIFIER). La macro OBJECT-IDENTITY se usa para definir información sobre la asignación de un identificador de objeto. La expansión de la macro OBJECT-IDENTITY conceptualmente sucede durante la implementación y no en tiempo de ejecución.

Definición de Tablas

Actualmente la SMI soporta únicamente una forma de estructuración de datos, estas son precisamente tablas bidimensionales simples con entradas de valores escalares. La definición de tablas involucra el uso de los tipos ASN.1: SEQUENCE y SEQUENCE OF y el IndexPart de la macro OBJECT-TYPE).

Como ejemplo se toma el fragmento donde se declara la tabla Gestionar Usuarios de la MIB del sistema SGCST.

```
gstGestionarUsuariosTable OBJECT-TYPE
    SYNTAX SEQUENCE OF GstGestionarUsuariosEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "Tabla para adicionar o eliminar usuarios del Servicio
                 Gestión de Servicios Telefónicos"
    ::= { tgst 1 }
```

2.3. LENGUAJE DE PROGRAMACION PHP.

El lenguaje PHP es un lenguaje de programación de estilo clásico, lo que quiere decir que es un lenguaje de programación con variables, sentencias condicionales, bucles, funciones etc. No es un lenguaje de marcas como podría ser HTML, XML o WML. Está más cercano a lenguajes de programación como JavaScript o a C++. Pero a diferencia de Java o JavaScript, que se ejecutan en el navegador, PHP se ejecuta en el servidor.

Como ejemplo un código Sencillo de PHP que imprime en pantalla varias veces la palabra "Color" seguido de una numeración.

```
<!-- Manual de PHP -->
//Parte de HTML normal.
<html>
<head>
<title>Ejemplo de PHP</title>
</head>
<body>
<BR><BR>
// Parte de PHP
<?php
echo "Ejemplo de PHP<br>";
    for($i=0;$i<10;$i++)
    {
        echo "Color".$i."<br>";
    }
?>
</body>
</html>
```

Como PHP se ejecuta en el servidor, permite acceder a los recursos que tenga el servidor como una base de datos. En PHP los resultados son enviados al navegador, estos resultados pueden ser una página HTML (ver Figura 2.4).

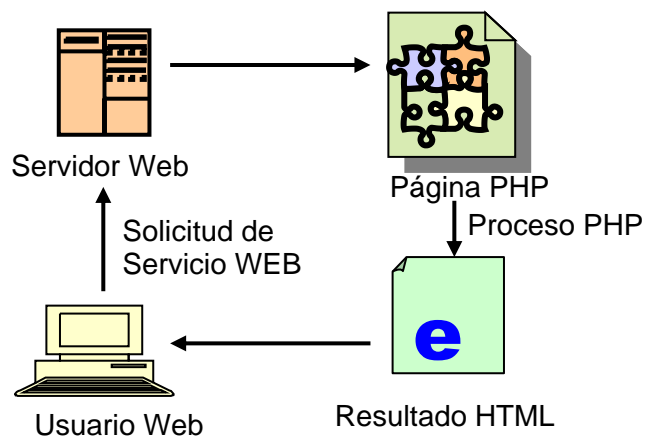


Figura 2.4: Esquema de funcionamiento en PHP

PHP es independiente del navegador, pero sin embargo para que las páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP. Este lenguaje permite embeber pequeños fragmentos de código dentro de páginas HTML y realizar determinadas acciones de una forma fácil y eficaz sin tener que generar programas programados íntegramente en un lenguaje distinto al HTML.

Gestión de bases de datos

Una de las grandes virtudes de PHP, es poder gestionar bases de datos, lo cual permite desarrollar aplicaciones completas con volúmenes elevados de información, siendo muy útil en particular al crear sitios Web. El lenguaje PHP ofrece interfaces para el acceso a la mayoría de las bases de datos comerciales ODBC, PHP resulta sencillo de manejar a la hora de gestionar bases de datos desde aplicaciones Web.

PHP permite las conexiones persistentes, que son enlaces SQL que no se cierran cuando termina la ejecución del archivo de comandos. Otra forma de conexión es la conocida como la 'idéntica', es una conexión que se abre hacia el mismo "host", con el mismo usuario y la misma contraseña.

2.3.1 Reseña Histórica.

El lenguaje PHP fue desarrollado originalmente en el año 1994 por Rasmus Lerdorf, aunque no hubo una versión utilizable por otros usuarios hasta principios de 1995. Esta primera versión se llamó, *Personal Home Page Tools* o "Herramientas para páginas Web personales", consistía en un CGI escrito en C que permitía la interpretación de un número limitado de comandos.

El siguiente paso en la evolución de PHP se realizó a mediados del 97 cuando se creó una nueva versión, la cual incluía nuevas funcionalidades como el soporte a nuevos protocolos de Internet y funciones para la gran mayoría de las bases de datos comerciales. Todas estas mejoras sentaron las bases de PHP versión 3. Actualmente PHP se encuentra en su versión 4, que utiliza un motor desarrollado con mayor meditación para cubrir las necesidades actuales y solucionar algunos inconvenientes de versiones anteriores.

En los últimos años los servidores que utilizan PHP van en aumento, convirtiéndolo en una tecnología popular. Gracias a la aceptación que ha logrado, y los grandes esfuerzos realizados por una creciente comunidad de colaboradores para implementarlo de la manera más óptima, podemos asegurar que el lenguaje PHP compartirá los éxitos logrados por los anteriores sistemas desarrollados en código abierto.

2.3.2 Estructuras importantes en PHP.

Variables

Una variable PHP permite almacenar datos como se realizaría en la mayoría de lenguajes de programación. En PHP todas las variables comienzan con el símbolo peso \$, y no es necesario definir una variable antes de usarla. Las variables en PHP no presentan tipos, es decir que una misma variable puede contener un número y luego puede contener caracteres (ver ejemplo), esta es una gran diferencia a otros tipos de lenguajes de programación.

Funciones:

El uso de funciones da la capacidad de agrupar varias instrucciones bajo un solo nombre y poder llamarlas a estas varias veces desde diferentes sitios, ahorra la necesidad de escribirlas de nuevo. Opcionalmente se pasan parámetros a las funciones que se tratarán dentro de ellas como variables locales y así mismo poder devolver un resultado. Para declarar una función se utiliza la instrucción **function** seguido del nombre, y después entre paréntesis la lista de argumentos separados por comas.

Ejemplo general:

```
<?php
function mayor ($x, $y)
{
//Instrucciones de la función
    $msg = "";
    if ($x > $y) {
        $msg = $x." es mayor que".$y;
    }
    else {
        $msg = $y." es mayor que".$x;
    }
    return $msg;
}
?>
```

Cualquier instrucción válida de PHP puede aparecer en el cuerpo de una función, incluso otras funciones y definiciones de clases.

Funciones SNMP:

En PHP es muy fácil invocar funciones u órdenes SNMP, lo cual permite realizar consultas a la MIB del sistema de gestión telefónica:

Ejemplo con **SNMPGET** para extraer los estados de los servicios:

```
//Busca los valores de las instancias en cada tabla de la MIB según la posición "ins"
/"ins" define la posición del usuario
$EstadoCodigoSecreto= snmpget("127.0.0.1", "public", "$oidecs.$ins");
$EstadoCasilleroVoz= snmpget("127.0.0.1", "public", "$oidecv.$ins");
$EstadoTransferenciaLLamada= snmpget("127.0.0.1", "public", "$oidet11.$ins");
```

2.3.3 Seguridad en PHP

PHP tiene mucha versatilidad cuando es instalado en un servidor Web como modulo o ejecutado como CGI. Puede acceder archivos, ejecutar comandos y abrir comunicaciones de red en el servidor. Todas estas características hacen que lo que se ejecute en el servidor Web sea inseguro por defecto. PHP ha sido diseñado específicamente, para ser un lenguaje más seguro en el desarrollo de programas CGI, que lenguajes como Perl o C.

Para PHP, existen multitud de opciones de configuración que permiten controlar su funcionamiento. Las combinaciones de estas opciones y configuraciones del servidor pueden realizar instalaciones inseguras. Con la correcta selección de las opciones de configuración en tiempo de compilación y ejecución se puede lograr una muy buena combinación de libertad y seguridad.

2.4 SISTEMA DE ADMINISTRACION TELEFONICA SATplus.

El sistema SATplus es una eficiente herramienta de gestión para empresas telefónicas, es ampliamente aceptada y es muy utilizada por las operadoras nacionales de telefonía fija. Esta aplicación fue desarrollada por la empresa Open Systems como solución a múltiples problemas de administración de la información que manejan las operadoras telefónicas, en general SATplus es una aplicación que permite realizar gestión sobre una base de datos relacional mediante interfaces en forma de tablas planas.

El SATplus es una aplicación soportada sobre un sistema operativo como Unix o Windows. Esta aplicación permite administrar o gestionar la información de los usuarios, de los servicios y de diversos aspectos técnicos y administrativos de las empresas telefónicas. El sistema SATplus opera sobre una base de datos relacional, pero con una interfaz amigable fácil de usar. Las áreas en las que se emplea el SATplus son:

- Atención Al cliente
- Mercadeo y Comercial
- Redes
- Facturación
- Mantenimiento
- Servicios

Este sistema posee una gran capacidad de procesamiento de información, cuenta además con un nivel de seguridad extremadamente alto, algo muy necesario ya que en la base de datos se almacena información vital para la empresa, como la utilizada para realizar la tarificación que se incluye en los recibos de los usuarios.

2.4.1 Características Generales del SATplus

El SATplus presenta las siguientes características:

- Funciona en ambientes Unix y Windows, con una amigable interfaz gráfica.
- Es compatible con prácticamente todas las marcas y modelos de computadoras.
- Posee poderosos generadores de reportes en forma de tablas, que es una de las grandes diferencias con una aplicación básica de bases de datos. Permite una visualización más clara de usuarios y servicios de las líneas telefónicas.
- Su operación es multiusuario, lo que permite que dos o más personas en diferentes equipos conectados en red puedan obtener resultados del mismo sistema SATplus en forma simultánea.
- Incluye soporte multiproveedor que le garantizan funcionalidad absoluta con los principales proveedores de larga distancia.

2.4.2 Servicio de consulta de peticiones de Servicios Suplementarios.

Dentro de las múltiples opciones que permite el SATplus para realizar consultas, la que más interesa a la aplicación SGCST, son las que involucran a los servicios suplementarios. Para realizar una consulta se escribe en la página inicial la orden correspondiente según la consulta que se desee, como en este ejemplo de Consulta a Servicios Suplementarios:

[Petición / Servicio Suplementario / Consulta Peticiones Serv. Supl.]

También es posible realizar la misma consulta solo ingresando el mnemónico **SCPS**. Existe un mnemónico para cada tipo de consulta y son muy utilizados por los operadores del SATplus por su facilidad de uso.

Para ilustrar se muestra el formulario que se genera cuando se solicita la consulta de Peticiones a Servicios Suplementarios, tomado de la Guía del Usuario, Sección 9, correspondiente al Módulo de atención, página 9-135, facilitada por la empresa TELEARMENIA (ver Figura 2.5).

The screenshot shows a web browser window titled 'SATplus [Petición Servicios Suplementarios]'. The main content area displays a form titled 'Petición de Servicios Suplementarios'. The form consists of two columns of input fields:

Departamento		Localidad	
No. Petición		Asignación	
Tipo de Petición		Valor Petición	
No. Servicio		Fecha de petición	
Suscripción		Nombre	
Serv. Suplementario		Estado Serv. Supl.	

Figura 2.5: Ejemplo de un Formulario del SATplus

2.4.3 Base de datos del SATplus

La aplicación SATplus fue desarrollada por Open Systems para gestionar bases de datos relacionales. Dado que las nuevas versiones de SATplus que son del interés del proyecto, están basadas en la base de datos ORACLE y el lenguaje SQL, se hace una breve descripción de las mismas.

- **ORACLE**

Es un Manejador de Bases de Datos Relacional (RDMS), unido a una gran cantidad de herramientas que permiten crear tablas, insertar, borrar, modificar y seleccionar datos, autorizar accesos, realizar cálculos y obtener informes. ORACLE soporta sentencias SQL, que es el lenguaje estándar para el manejo de bases de datos relacionales.

Características propias de ORACLE

El manejador de base de datos ORACLE tiene características importantes que facilitan diversos tipos de procesos de datos, como: simplicidad y generalidad, son de fácil manejo para el usuario final y las consultas de información se realizan de forma sencilla, además permiten el manejo de un gran volumen de tablas.

El manejo de las tablas en las bases de datos ORACLE, permiten representar la información de una forma más compacta, es posible acceder a la información contenida en dos o más tablas de forma simultanea.

Las tablas almacenan información de una forma organizada y cumplen las siguientes leyes básicas:

- Un número definido de campos por tabla.
- Cada nombre de los campos de una tabla es distinto.
- Cada registro de la tabla es único.
- El orden de los registros y de los campos no está determinados.
- Para cada campo existe un conjunto de valores posible.

Las bases de datos de ORACLE definen los **Objetos** Relacionales, estos objetos se relacionan entre sí mediante referencias a objeto e integridad referencial, para el futuro se espera que estos Objetos relacionales adquieran características de la orientación a objetos, se podrán crear tablas de forma independiente que heredaran atributos y métodos de alguna estructura padre que “podría” ser una clase.

Otras características de ORACLE 8, versión utilizada en la aplicación SGCST son:

- Implementación del estándar SQL3.
- Llamadas a procedimientos externos (Ejemplo llamar a una función "C")
- Muchas mejoras en el rendimiento de SQL/PLSQL/OCI con un uso más eficiente de los recursos Hardware del sistema.
- "Connection Pooling": Se usa una conexión física para usuarios sin carga y de forma transparente se reestablece la conexión cuando se necesita.
- Security Server: Para administración centralizada de usuarios y aumento de la seguridad de los accesos.

Componentes de su entorno de Desarrollo.

En el mercado del software se encuentran muchos productos de desarrollo para soporte a las bases de datos ORACLE, se destacan las siguientes:

SQLMenu:

Es un producto de ORACLE destinado a producir árboles de menús que permiten al usuario desplazarse fácilmente a través de su aplicación, facilitando a su vez, la integración con los diferentes módulos de ORACLE.

SQLForms:

Es la herramienta de Oracle que permite, de un modo sencillo y eficiente, diseñar pantallas para el ingreso, modificaciones, bajas y consultas de registros. El usuario podrá, una vez definida la forma, trabajar con ella sin necesidad de generar códigos, dado que Oracle trae incorporado un conjunto de procedimientos y funciones asociados a las teclas de funciones, como por ejemplo la tecla [F7], que se usa para iniciar una consulta.

SQLReport:

Herramienta de Oracle que realiza de forma flexible, sencilla y eficiente la creación de reportes, informes o listados permitiendo, entre otras facilidades, la visualización previa por pantalla con una gran variedad en estilos de presentación.

Oracle Designer:

Oracle Designer es un conjunto de herramientas de generación determinada por modelos de Oracle que permiten el diseño y generación de lógica de aplicaciones y esquemas de base de datos, esta aplicación soporta el diseño de complejos sistemas de información con reingeniería de procesos de negocios (BPR).

Características de Oracle Designer:

- Desarrollo de Escala Empresarial.
- Reingeniería de Procesos de Negocios.
- Sistemas Visuales y Modelajes de Diseño.
- Desarrollo de Sistemas Dirigido por Modelos.
- Repositorio Abierto / Diagramador de Relaciones.
- Creación de Aplicaciones Cliente.
- Entrega de diseño de Cliente / Servidor de segunda generación.
- Soporte Flexible para Técnicas BPR.
- Modelador de Jerarquía de Funciones.
- Modelador de Flujo de Datos.
- Funciones de Generación de Aplicación Developer/2000.
- Generador de código de Visual Basic.

• SQL

Es un lenguaje para organizar, gestionar y recuperar datos almacenados en una base de datos informática. El nombre SQL es una abreviatura de Structured Query Language (Lenguaje Estructurado de Consultas). SQL trabaja con un tipo específico de base de datos, llamado base de datos relacional.

SQL se utiliza para controlar todas las funciones de un DBMS. Son propiedades de este lenguaje:

- ◆ Definición de datos: SQL permite a un usuario definir la estructura y organización de los datos almacenados y las relaciones entre ellos.

- ◆ Recuperación de datos: SQL permite a un usuario o un programa de aplicación recuperar los datos almacenados de la base de datos y utilizarlos.
- ◆ Manipulación de datos: SQL permite a un usuario o a un programa de aplicación actualizar la base de datos añadiendo nuevos datos, suprimiendo datos antiguos y modificando datos previamente almacenados.
- ◆ Control de acceso: SQL puede ser utilizado para restringir la capacidad de un usuario para recuperar, añadir y modificar datos, protegiendo así los datos almacenados frente a accesos no autorizados.
- ◆ Comparación de datos: SQL se utiliza para coordinar la comparación de datos por parte de usuarios concurrentes, asegurando que no interfieren unos con otros.
- ◆ Integridad de datos: SQL define restricciones de integridad en la base de datos, protegiéndola contra corrupciones debidas a actualizaciones inconsistentes o a fallos del sistema.
- ◆ El motor de la base de datos es el corazón del DBMS, responsable de estructurar, almacenar y recuperar realmente los datos en el disco. Acepta peticiones SQL procedentes de otros componentes RDBMS, tales como formularios, un escritor de reportes o una facilidad de consultas interactivas, desde los programas de aplicación escritos por el usuario e incluso desde otros sistemas de información.

SQL es a la vez un lenguaje fácil de entender y una herramienta completa para gestionar datos. Otras características de SQL son:

- ◆ Su portabilidad a través de sistemas de información.
- ◆ Su fundamento relacional.
- ◆ Su estructura de alto nivel semejante al inglés.
- ◆ Las consultas interactivas.
- ◆ Su acceso a la base de datos mediante programas.
- ◆ Su definición dinámica de datos.
- ◆ La arquitectura cliente / servidor.

3. DISEÑO

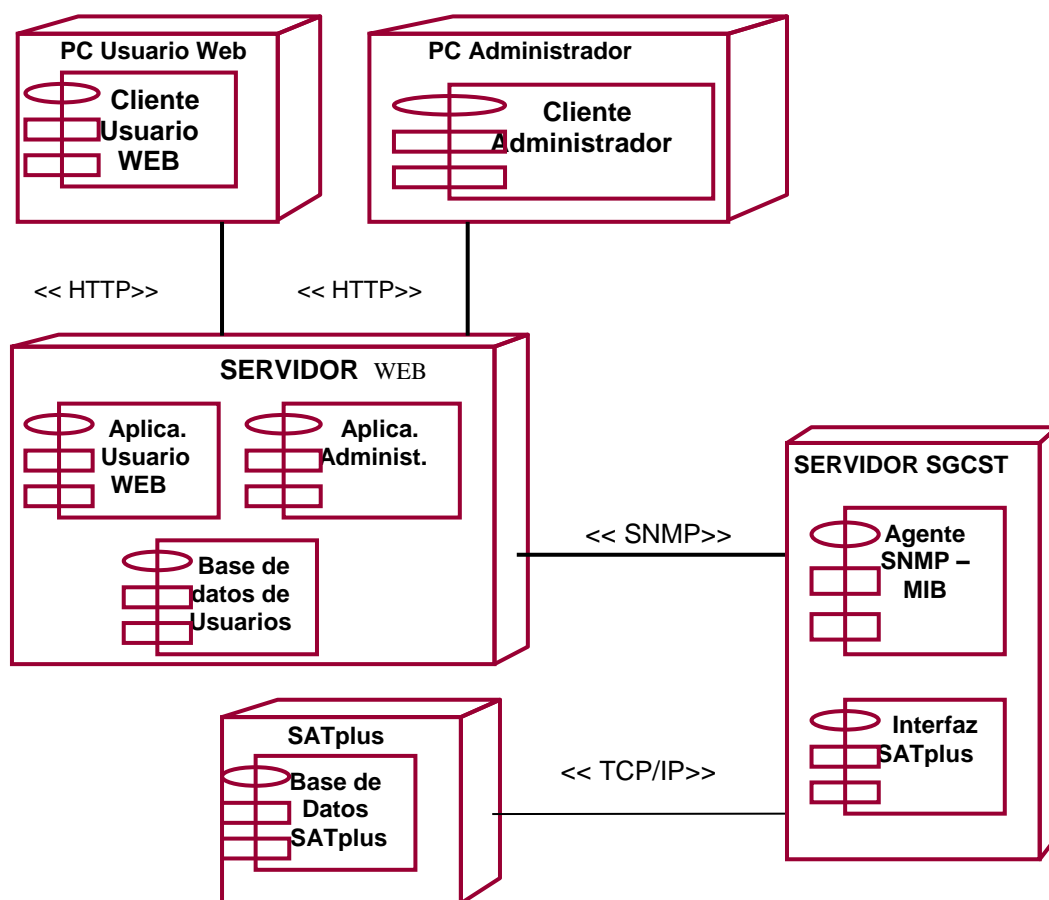
Siguiendo las pautas anotadas en el Anteproyecto, se propone en este capítulo el diseño del Sistema de Gestión de Configuración de Servicios Telefónicos Suplementarios por Internet o SGCST. En el se describen los diagramas de Implantación, Subsistemas, Clases y de Secuencias del sistema SGCST.

Adicionalmente en este capítulo se estudian de forma breve las Bases de Datos de Usuarios y de Información de Gestión MIB del sistema. En ellas se describen las tablas que las conforman y sus atributos. Por ser simples, al contener muy pocas tablas y al no tener relaciones directas, no se hace necesario incluir los diagramas de entidad relación de estas bases de datos.

El documento de diseño resulta de mucha importancia para entender de forma clara y precisa el trabajo realizado en la aplicación del sistema SGCST. Por esta razón se decidió incluirlo como un capítulo previo al estudio de Gestión y de Aplicación del sistema.

El diagrama descriptivo del sistema y los casos de usos reales que son parte del Diseño, se incluyen previamente en el capítulo primero de este documento. De ellos se parte para el análisis de los elementos que se estudian en este capítulo. Se sugiere examinarlos antes de continuar.

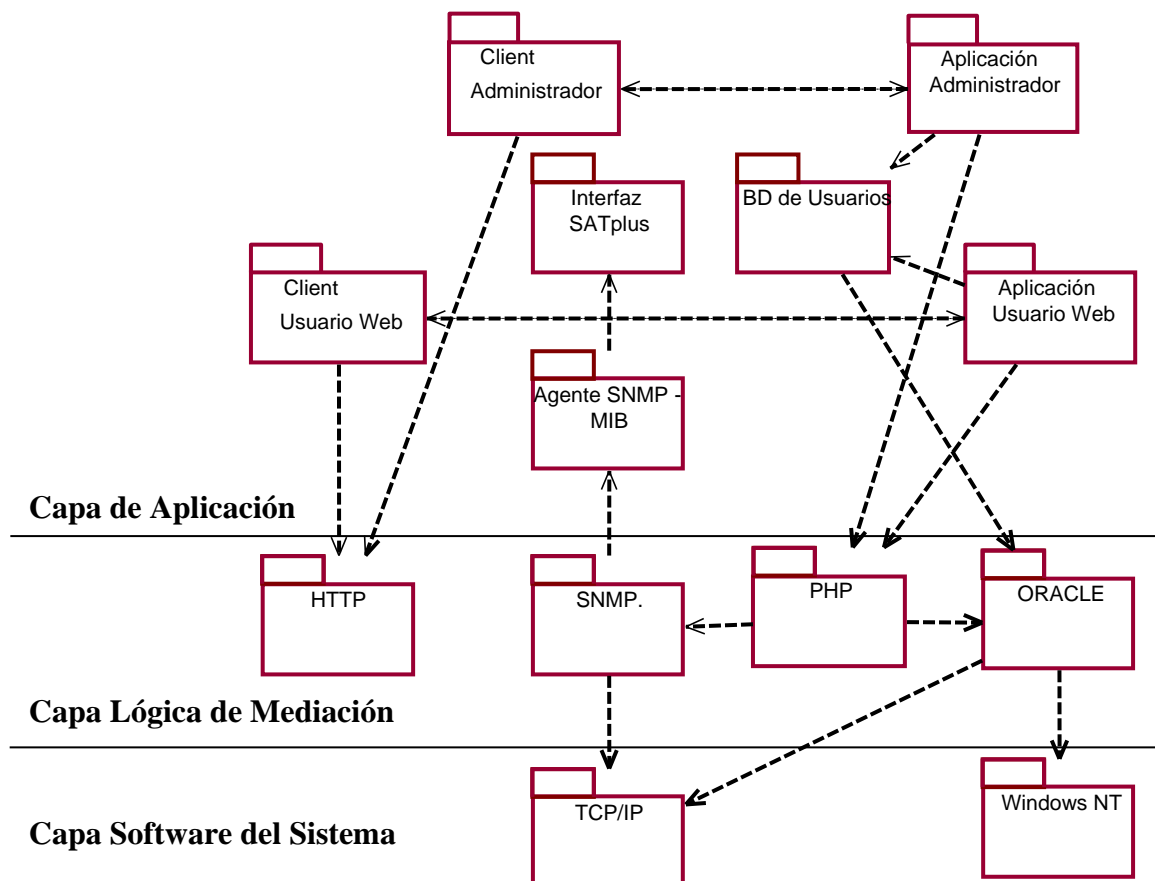
3.1 DIAGRAMA DE IMPLANTACION



En este diagrama de implantación se tiene una aplicación cliente para cada usuario del sistema: el Usuario Web y el Administrador del sistema, cada uno representado como un nodo independiente. Adicionalmente se tiene el servidor de gestión SGCST que atiende las solicitudes enviadas desde el Servidor Web mediante ordenes SNMP.

En el servidor SGCST se gestiona la información en la MIB a partir de ordenes SNMP. Estas son enviadas por el Servidor Web vía protocolo SNMP. Una vez ejecutadas las órdenes hace una llamada a una aplicación que sirve de interfaz, la cual realiza una conexión al servidor con la Base de Datos donde corre el sistema SATplus. En la Base de datos se realizan modificaciones a la tabla que registra las peticiones de servicios suplementarios de los clientes.

3.2 DIAGRAMAS DE SUBSISTEMAS E INTERFACES



A continuación se describen de forma general uno a uno los paquetes de diseño del sistema de gestión telefónica:

- 1. Client Administrador:** Contiene las interfaces que son páginas en HTML, para el administrador del sistema.
- 2. Client Usuario Web:** Contiene las interfaces que son páginas en HTML, para el Usuario del sistema.
- 3. Aplicación Administrador:** Contiene las aplicaciones de gestión que son páginas en PHP, para el administrador del sistema.
- 4. Aplicación Usuario Web:** Contiene las aplicaciones de gestión que son páginas en PHP, para el Usuario del sistema.

5. Interfaz SATplus: Es un programa que permite comunicar el agente con la base de datos del SATplus, denominada UPDPOCI.

6. PHP: Lenguaje de programación en que se desarrollaron las aplicaciones, soporta órdenes SNMP Get y Set, y manejo de órdenes a bases de datos.

7. ORACLE: Motor de bases de datos sobre la cual esta implantada la base de datos de usuarios.

8. SNMP: El protocolo SNMP contiene primitivas básicas que le permiten gestionar los objetos o las instancias de los objetos en la MIB, estas son GET y SET. Comunica la gestión del sistema con el Agente SNMP.

9. Agente SNMP - MIB: Es una aplicación donde se desarrollan todas las operaciones sobre las instancias en el árbol de la MIB, además permite comunicarse con una aplicación que sirve de interfaz con la base de datos del SATplus, y en la cual ejecuta operaciones, esta desarrollado y compilado en lenguaje de programación Visual C++. El Agente SNMP contiene la **MIB**, esta compuesta por tablas de información (Tipo lista) orientadas a los servicios y los usuarios, las cuales son gestionadas por el Agente SNMP.

10. Base de datos de Usuarios: Es la base de datos que almacena la tabla de registros de los usuarios del sistema, también contiene la tabla de los Administradores o usuarios de la Intranet.

11. Windows NT: Es el sistema operativo, permite realizar tareas de gestión y brinda un gran soporte para aplicaciones basadas en SNMP gracias al servicio SNMP.

12. HTTP y TCP/IP: Son los protocolos que permiten la comunicación hardware de los equipos tanto clientes como servidores, son protocolos estándar para aplicaciones en Internet, en este proyecto funcionan bajo la plataforma Windows NT.

3.3 DIAGRAMAS DE CLASES POR PAQUETES.

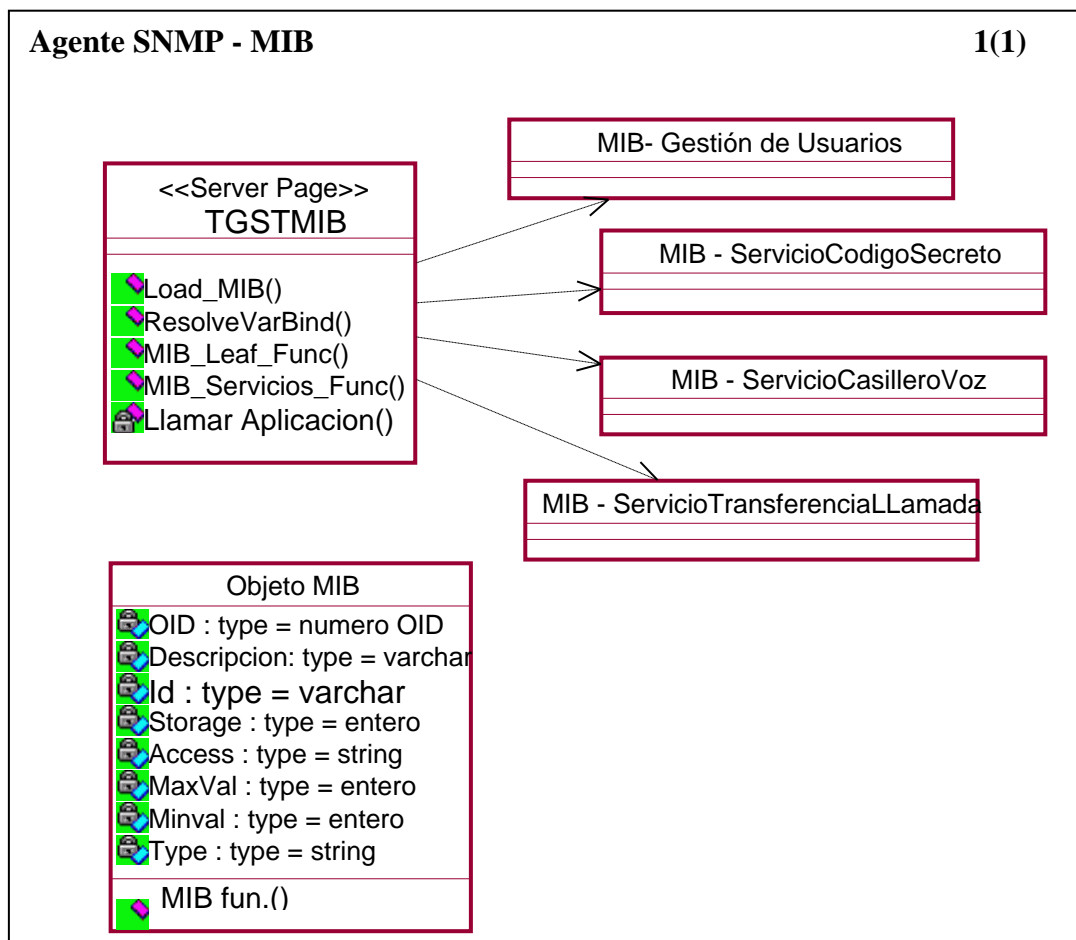


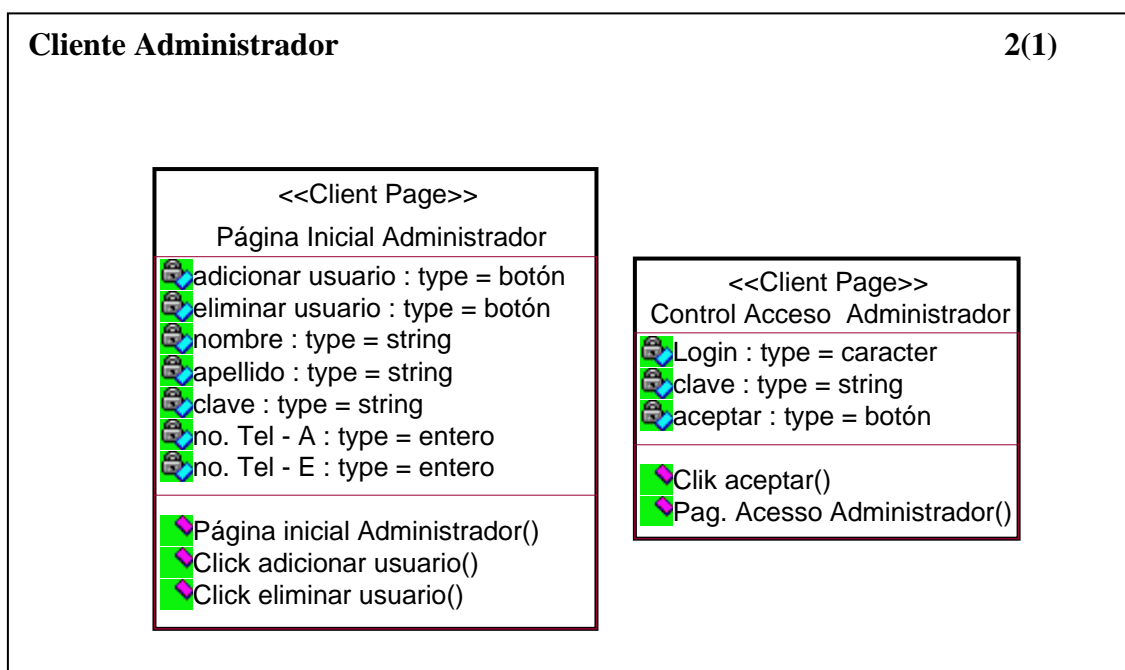
Tabla	Descripción
MIB- Gestión de Usuarios	Esta tabla almacena la información general del usuario del sistema y permite definir el index del usuario.
MIB - Servicio Código secreto	Esta tabla almacena la información del servicio de código secreto en la MIB
MIB - Servicios Casillero de Voz	Esta tabla almacena la información del servicio de casillero de voz en la MIB
MIB - Servicio Transferencia Llamada	Esta tabla almacena la información del servicio de transferencia de llamada en la MIB

Ojo dibujo





Para mayor información de las tablas de la MIB, dirijase al capítulo 6.

Agente SGT	Descripción:	Es el agente SNMP del sistema de gestión SGCST
	Operaciones:	
	Get	Captura datos en las instancias de la MIB del sistema
	Set	Modifica Datos en las instancias de la MIB del sistema
	Call Interfaz SATplus	Llama la aplicación SQL que se conecta a la base de datos del SATplus, a la tabla Petición de Servicios.

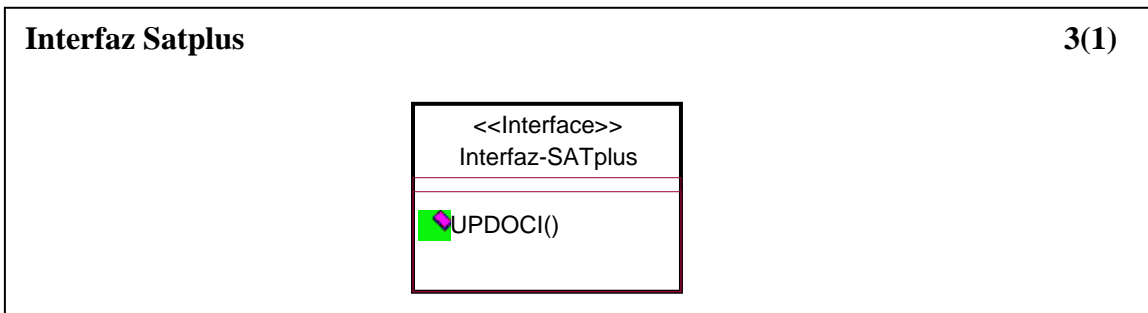
Objeto MIB	Descripción:	Es la clase de los objetos de las tablas de la MIB
	Operaciones:	
	MIB func	Función para el objeto



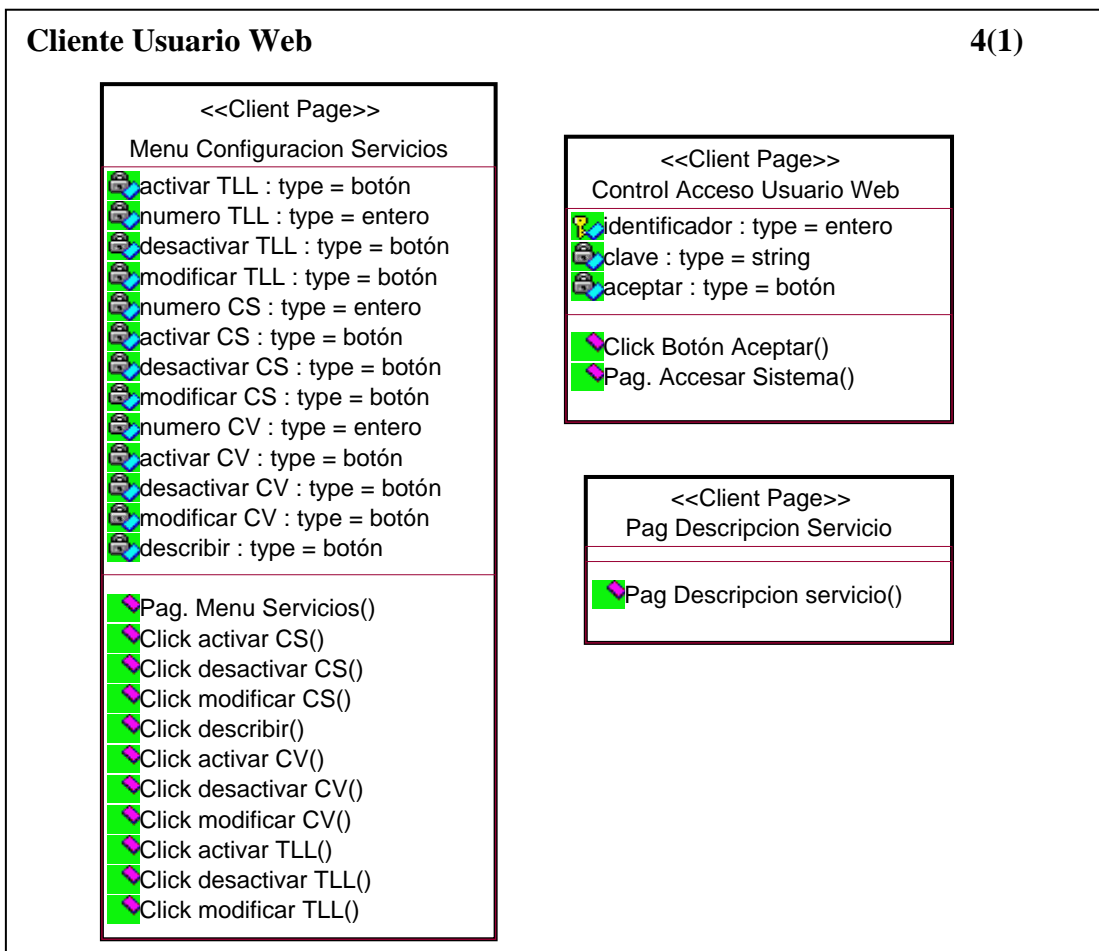
Página Inicial Administrador	Descripción:	Es la página principal del Administrador de desde donde puede gestionar a los usuarios del sistema.
	Operaciones:	
	Pág. Inicial Administrador	Carga la página Inicial del administrador.
	Click Adicionar usuario,	Invoca a la función que permite dar de alta a un usuario en el sistema.
Control Acceso Administrador	Descripción:	Es la página de inicio del administrador, donde digita los datos para ser validado en el sistema.
	Operaciones:	
	Pág. Acceso	Carga la página que permite el ingreso al sistema.
	Click Botón Aceptar	Invoca la operación de validación de los datos del Usuario Administrador.

Aplicación Administrador	2(2)
<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> <p style="text-align: center;"><<Server Page>> Gestion Usuarios</p> <hr/> <ul style="list-style-type: none">  Adicionar Usuario()  Eliminar Usuario()  Validar Administrador()  Enviar Pag. Inicial() </div>	

Gestión Usuarios	Descripción:	Esta página es la que invoca las operaciones de configuración y de gestión de los usuarios, además ejecuta la operación de validar al administrador.
	Operaciones:	
	Adicionar usuario	Ejecuta la operación de Adicionar o dar de alta un usuario en el sistema.
	Eliminar usuario	Ejecuta la operación de eliminar o dar de baja un usuario en el sistema.
	Validar Administrador	Ejecuta la operación de validación de los datos del Usuario Intranet.



Interfaz SATplus	Descripción:	Esta clase permite conectar la base de datos sobre la que corre el SATplus, accede a la tabla Petición de servicios.
	Operaciones:	
	UPDOCI()	Realiza una conexión a la base de datos del SATplus desde la cual puede realizar consultas y modificar la tabla Petición de servicios.



Menú Configuración Servicio.	Descripción	Es la página principal del usuario donde puede gestionar los servicios suplementarios adscritos.
	Operaciones	
	Activar	Invoca la operación Activar servicio (Según servicio).
	Modificar	Invoca la operación Modificar servicio (Según servicio).
	Desactivar	Invoca la operación Desactivar servicio (Según servicio).
	Pág. Menú Servicios	Carga la pantalla de Menú de configuración de servicios.
Pág. Descripción Servicio.	Descripción:	Página donde se describe el funcionamiento del sistema de gestión telefónica SGCST para los usuarios.
	Operaciones	
	Pág. Descripción Servicio	Carga la pantalla donde se describe el funcionamiento de la aplicación SGCST para los usuarios.
Control Acceso Usuario Web	Descripción	Página de inicio del usuario web, donde digita los datos para ser validado en el sistema.
	Operaciones:	
	Pág. Acceso	Carga la página que permite el ingreso al sistema.
	Click Botón Aceptar	Invoca la operación Validar usuario Web.

Aplicación Usuario Web	4(2)
<div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: 80%;"> <p style="text-align: center;"><<Server Page>> Gestion Servicios</p> <hr style="border: 0.5px solid red;"/> <ul style="list-style-type: none"> ▶ Validar Usuario Web() ▶ Activar Servicio() ▶ Desactivar Servicio() ▶ Modifcar Servicio() ▶ Enviar Pag. Menu Servicios() ▶ Enviar Pag. Descripcion Servicio() </div>	

Gestión Servicios	Descripción:	Clase que permite atender las solicitudes de gestión de servicios por parte de los Usuarios Web, también valida los accesos de los usuarios al sistema.
	Operaciones:	
	Enviar Pág. Menú Servicios.	Carga la página de menú de servicios del usuario.
	Validar usuario web	Ejecuta la operación que valida los datos del usuario.
	Enviar Pág. Descripción	Carga la página de descripción del servicio SGCST
	Activar Servicio	Ejecuta la operación que activa el servicio solicitado.
	Desactivar Servicio	Ejecuta la operación que desactiva el servicio deseado.
	Modificar Servicio	Ejecuta la operación que modifica el servicio solicitado

3.4 BASE DE DATOS DE USUARIOS

La base de datos de usuarios del sistema de gestión telefónica por Internet únicamente contiene dos tablas y no están relacionadas por ser completamente independientes. Una tabla almacena la información de los usuarios del sistema y la otra de los administradores.

3.4.1 Tabla Registro de Usuarios.

En la tabla de Registro de Usuarios (Nombre de la tabla en la base de datos: **Gesteruser**) se almacena la información básica de los usuarios o clientes del servicio de gestión telefónica. La tabla contiene un identificador (**Index**) con características de llave primaria, que nos permite definir la posición del usuario dentro de la tabla, además este parámetro es de mucha importancia, por que con el se genera la consulta en la MIB del sistema. La tabla Gesteruser es la siguiente:

GESTERUSER			
Parámetro	Descripción	Longitud	Tipo
Index	Numero que indica la posición del usuario	38	(Number)
Nombre	Nombre del usuario	50	(Varchar2),
Apellido	Apellido del usuario	100	(Varchar2),
Teléfono	Numero telefónico del usuario	10	(Number)
Password	Clave de ingreso para el usuario	100	(Varchar2),

TABLA 1: GESTIÓN DE USUARIOS

3.4.2 Tabla Gestión de Administradores.

En la tabla de Registro de administradores se almacena la información básica de los administradores del sistema de gestión telefónica, el nombre de la tabla en la base de datos es **Gesteradmi**, tiene un identificador (**Index**) el cual permite definir el número de usuarios administradores que existen en la tabla y en el sistema. La tabla Gesteradmi es la siguiente:

GESTEADMI			
Parámetro	Descripción	Longitud	Tipo
Index	Indica el número del Administrador.	38	(Number)
Nombre	Nombre del Administrador	50	(Varchar2),
Apellido	Apellido del Administrador	100	(Varchar2),
Login	Nombre identificador de entrada	10	(Number)
Password	Clave de ingreso para el Administrador	100	(Varchar2),

TABLA 2: GESTIÓN DE ADMINISTRADORES

3.5 DESCRIPCIÓN DE LA MIB

Este capítulo describe el contenido del árbol de la MIB de la aplicación SGCST, realizando una reseña de cada una de las tablas y sus atributos, también se estudia las relaciones de las tablas en los procesos de consulta y modificación.

3.5.1 Árbol de Información de la MIB

El árbol de la MIB - SGCST lo componen cuatro Tablas así:

- TABLA GestionarUsuarios
- TABLA ServicioCodigoSecreto
- TABLA ServicioCasilleroVoz
- TABLA TransferenciaLLamada

La tabla Gestionar usuarios:

TABLA GestionarUsuarios

- **NumeroTelefonico:** Dato numérico de 6.
- **UsuarioIndex:** Dato numérico de 2 o más dígitos según se requiera.
- **CodigoUsuario:** Dato numérico de 2 o mas dígitos.

El **Numero Telefónico** y el **Usuario Index** son datos generales del usuario o suscriptor del servicio, la instancia Numero telefónico se utiliza para identificar al usuario y como "Llave primaria" en la MIB. El Usuario Index es la instancia utilizada como referencia para acceder a la información del usuario en las demás tablas, cuando un usuario se elimina se borra el Numero Telefónico pero el Usuario Index se conserva y se puede utilizar para otro número telefónico (ver Figura 3.1).

Las Tablas de servicios:

TABLA ServicioCodigoSecreto

- **CodigoSecretoIndex:** Indica la posición dentro de la tabla
- **EstadoCodigoSecreto:** Estado del servicio Activo / Desactivo, tipo Bit 1 o 0.
- **NumeroCodigoSecreto:** Indica el número del código secreto, 4 dígitos.

TABLA ServicioCasilleroVoz

- **CasilleroVozIndex:** Indica la posición dentro de la tabla
- **EstadoCasilleroVoz:** Estado del servicio Activo / Desactivo, tipo Bit 1 o 0.
- **NumeroMensajesCasilleroVoz:** Indica el número de mensajes, 2 dígitos.

TABLA TransferenciaLLamada

- **TransferenciaLLamadaIndex:** Indica la posición dentro de la tabla
 - **EstadoTransferenciaLLamada:** Estado del servicio Activo / Desactivo.
 - **NumeroTransferenciaLLamada:** Numero a transferir llamada, 6 dígitos.

Los **Index** de cada una de las tablas de servicios nos indican la posición en la tabla para cada usuario del valor del servicio, gracias a esta posición se efectúa la consulta y la modificación de los datos del usuario en la MIB.

Cada servicio tiene asociada una tabla y cada tabla se compone de tres parámetros además del **Index** de cada tabla se tiene: Para el código secreto el **Número código secreto** que es el número clave de cuatro cifras para bloquear el teléfono, y el **estado código secreto** que indica el estado del servicio entre activo y desactivo. Para cada servicio el estado tiene el mismo significado. Para Transferencia de Llamadas se tiene además el **numero de transferencia** que indica a que numero serán transferidas las llamadas entrantes, y por ultimo para casillero de voz se tiene el **numero de mensajes** que permite recibir el casillero, se escoge entre los posibles valores entre **1** y **15** mensajes.

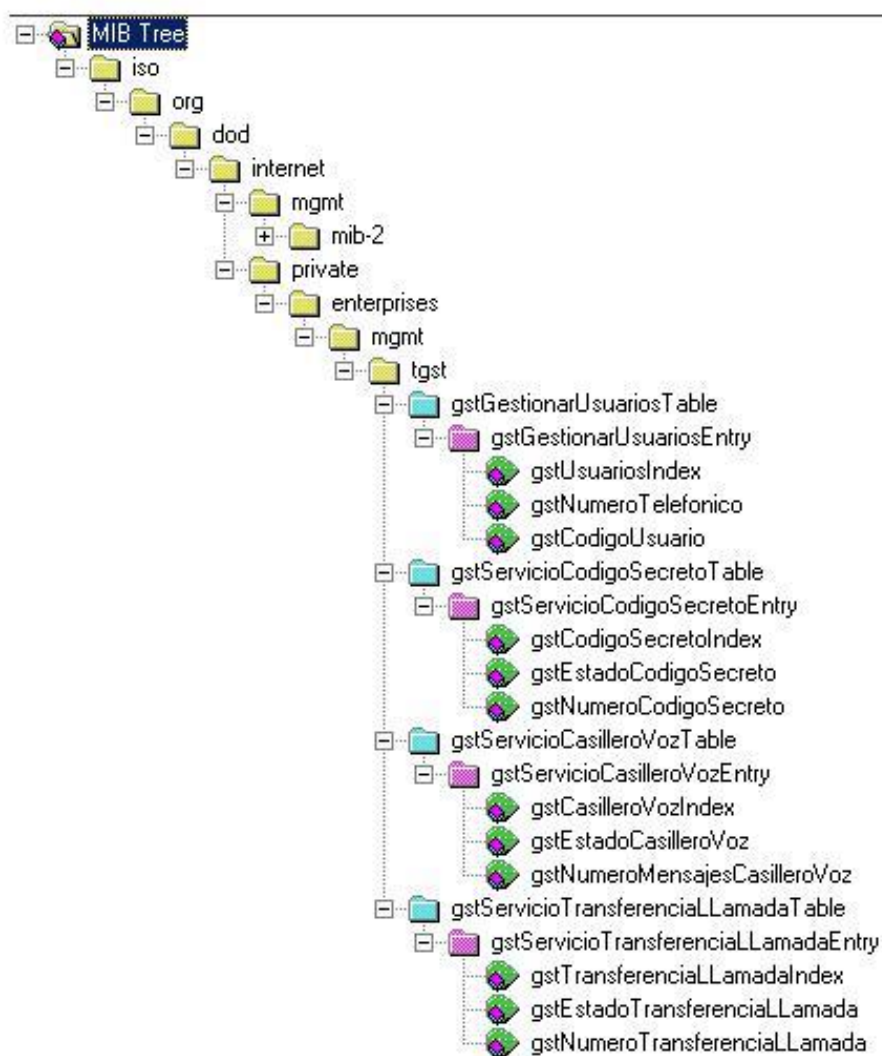


Figura 3.1: Árbol de la MIB - GST

Para acceder a los distintos elementos del árbol SNMP se utiliza el OID que define una posición dentro del mismo, cada OID de nuestro árbol presenta la siguiente estructura:

ISO	ORG	DOD	Internet	Private	Enterprises	MGMT	TGST	Tabla	Entry	Objeto	Instancia
1.	3.	6.	1.	4.	1.	11.	2.	1.	1.	1.	1.

3.5.2 Relaciones entre las tablas de la MIB

En cada tabla de la MIB se tiene un atributo adicional a los que se estudiaron previamente denominado **Index** que indica la posición dentro de la tabla de una instancia, estos Index nos permiten relacionar las tablas de la MIB ya que para un usuario el **Index** es igual en todas las tablas.

Como se menciona la tabla de Gestión de usuarios sirve de referencia para las demás tablas al realizarse una consulta a un usuario, el sistema previamente a capturado el número telefónico del usuario en el proceso de validación. Con el número telefónico el sistema ejecuta la operación **SNMPWALK**, que es muy similar al GETNEXT, con la diferencia que esta operación extrae de la MIB todos los valores de las instancias del objeto sobre el cual se ejecuta, en este caso sobre Número telefónico utilizando su **OID** ver Tabla.

TABLA	OBJETO	OID
Gestionar Usuarios	Usuario Index	.1.3.6.1.4.1.11.2.1.1.1
	Número telefónico	.1.3.6.1.4.1.11.2.1.1.2
Código Secreto	Código secreto Index	1.3.6.1.4.1.11.2.2.1.1
	Estado código secreto	.1.3.6.1.4.1.11.2.2.1.2
	Número código secreto	.1.3.6.1.4.1.11.2.2.1.3
Casillero de Voz	Casillero de Voz Index	.1.3.6.1.4.1.11.2.3.1.1
	Estado casillero de voz	.1.3.6.1.4.1.11.2.3.1.2
	Número casillero de voz	.1.3.6.1.4.1.11.2.3.1.3
Transferencia de Llamada	Transferencia llamada Index	.1.3.6.1.4.1.11.2.4.1.1
	Estado Transferencia	.1.3.6.1.4.1.11.2.4.1.2
	Número Transferencia	.1.3.6.1.4.1.11.2.4.1.3

TABLA 3: OID'S DE LA MIB DEL SISTEMA SGCST

3.5.3 Estructura de datos de la MIB

Antes de realizar una descripción de cómo el agente realiza una consulta a la MIB, es necesario dar claridad de los tipos de datos o la estructura de los datos que maneja la base de información de gestión MIB. En la MIB se manejan estructuras tipo lista para cada uno de los Objetos, como se muestra en el Grafico.

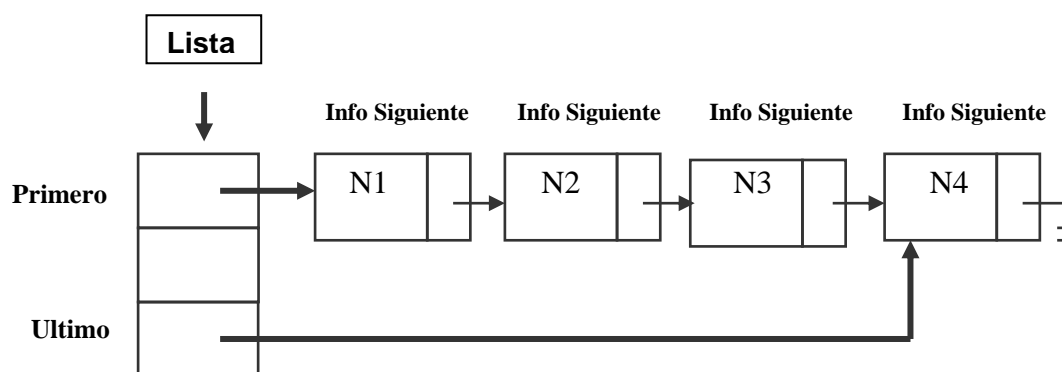


Figura 3.2: Estructura de los datos en la MIB

Todos los bloques o instancias de información que conforman las listas, son almacenados siguiendo los parámetros establecidos en el agente SNMP para todos los objetos en la MIB, los parámetros son:

OID	identificador de cada instancia, que permite Ubicarla dentro del Árbol de la MIB
Valor	El valor que almacena
Tipo	El tipo de información que almacena, (En la MIB SGCST solo se utilizan tipo entero)
Acceso	Tipo de acceso (Lectura o Lectura y Escritura)
Valor Máx.	Valor máximo que puede tomar
Valor Min.	Valor mínimo que puede tomar
Operación	La operación que permite leer o modificar el valor de esta instancia (GET , SET).

TABLA 4: PARÁMETROS DE LOS OBJETOS DEL SISTEMA SGCST

3.5.4 Consulta por el Agente SNMP (Ejemplo)

Como Ejemplo describimos el proceso para armar la página inicial del Usuario WEB:

La aplicación arma una lista con todos los números telefónicos de los usuarios del sistema, datos que extrae de la tabla de la MIB Gestión de usuarios mediante la orden SNMPWALK, luego mediante un ciclo For, ubica el numero del usuario mediante su **Index**, esta ubicación permite guardar la posición en una variable. Una vez se tiene la posición del usuario, la aplicación arma una consulta **SNMPGET** múltiple llamada SNMPWALK a los valores de las tablas de los servicios, así se obtiene los valores de los servicios del usuario y es posible “armar” la página inicial del usuario (ver Figura 3.3).

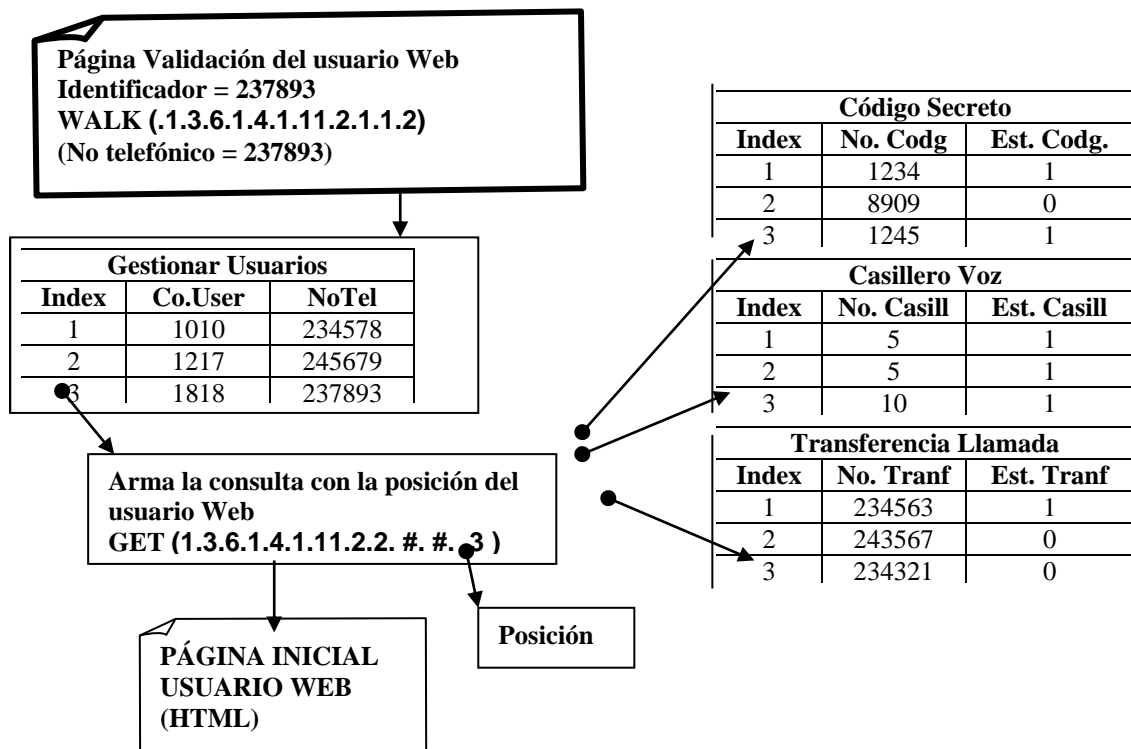


Figura 3.3: Relación entre las tablas de la MIB

3.6. DIAGRAMAS DE SECUENCIA

3.6.1 Diagrama de Secuencia - Adicionar Usuario.

Para este caso del uso el sistema permite Habilitar un usuario por parte del administrador.



Diagrama de Clases - Adicionar Usuario.

3.6.2 Diagrama de Secuencia - Validar Usuario Web.

El caso de uso Control de acceso Usuario Web, permite validar los clientes o usuarios del sistema.



Diagrama de Clases - Validar Usuario Web.

3.6.3 Diagrama de Secuencia - Activar Servicio (Código secreto)

Este diagrama analiza la activación de servicios por parte del usuario, se toma el ejemplo para código secreto .



Diagrama de Clases - Activar Servicio.

3.6.4 Diagrama de Secuencia - Modificar servicio (Transferencia de llamada)

Este diagrama analiza como se modifican los de servicios por parte del usuario



Diagrama de Clases - Modificar Servicio.

4. MODULO DE GESTIÓN SNMP

Sin duda el mayor aporte que brinda el proyecto de gestión de configuración de servicios de telefonía fija por Internet SGCST, es su modulo de gestión SNMP cuyo pilar fundamental es el agente SNMP desarrollado para el sistema SGCST, razón suficiente para llevar a cabo en este capítulo un análisis completo y detallado del funcionamiento de cada uno de los elementos que componen este modulo, el agente SNMP y la MIB del sistema. Para cada uno de estos dos elementos se describen sus características más relevantes y su codificación.

Básicamente él modulo de gestión está constituido por tres componentes: El protocolo de gestión simple SNMP, el Agente SNMP y finalmente la base información de gestión MIB del sistema (ver Figura 4.1). La MIB almacena la información de gestión de cada uno de los elementos gestionables del sistema, y sobre ellos el agente SNMP realiza operaciones de gestión.

Para implementar el sistema SGCST se emplea SNMP v1 que es la primera versión del protocolo SNMP. SNMP v1 no es una versión reciente y tiene asociadas limitaciones en seguridad, capacidad en el manejo de información y en el número de eventos que puede llegar a atender simultáneamente. Pero estas limitantes no impiden al protocolo SNMP ni al Agente SNMP cumplir con los objetivos propuestos en este proyecto.

Para complementar este capítulo se describe la aplicación que sirve de interfaz entre él modulo de gestión SNMP y la base de datos del sistema SATplus o *sistema de administración telefónica o de telecomunicaciones*. A está aplicación la denominamos UPDOCI (nombre que tomado de la sentencia UPDATE de SQL) y es invocada por el agente SNMP cuando se necesita registrar en el SATplus las solicitudes a servicios suplementarios que han sido requeridas por los usuarios del sistema SGCST.

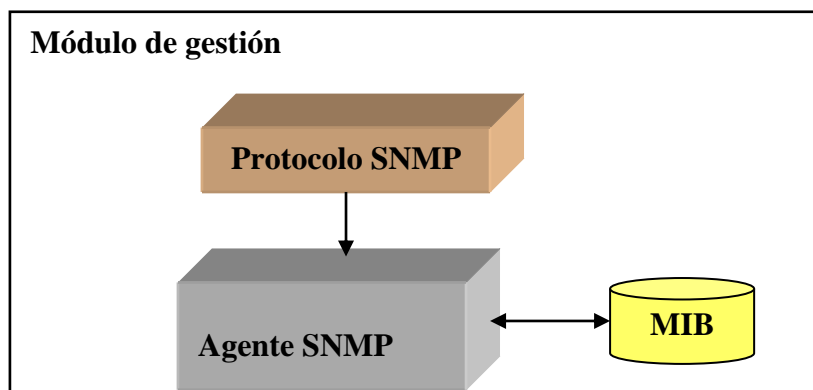


Figura 4.1: Módulo de gestión SNMP

4.1 DESCRIPCION DEL AGENTE SNMP.

4.1.1 Características Importantes.

Las siguientes son las características destacables del agente SNMP:

- El agente SNMP se encarga de la gestión de los elementos gestionables del sistema, como servicios y usuarios. Esta gestión la realiza gracias a las instancias de los objetos contenidas en la MIB, sobre las cuáles puede ejecutar operaciones de consulta y de modificación.
- Es de código flexible, es decir que es posible escalar el código del agente y agregarle funcionalidad que permita gestionar nuevos servicios y nuevos tipos de usuarios. Para realizar con éxito estos cambios en el agente es necesario además realizar modificaciones en el código de la MIB según los nuevos elementos.
- El agente SNMP hereda todas las facilidades que le brinda el protocolo SNMP para comunicar las aplicaciones de gestión con la MIB del sistema.
- El protocolo SNMP permite la gestión remota del agente con el soporte de una red como Internet.
- El agente SNMP permite integrar nuevas aplicaciones que posibilitan la comunicación con recursos externos del sistema. En este proyecto se desarrolla una aplicación que sirve de interfaz con la base de datos de la aplicación SATplus.

4.1.2 Operaciones del Agente SNMP-SGCST.

La programación y desarrollo del agente SNMP está directamente relacionada con la información que se almacena en la base de información de gestión MIB. En el agente se crean operaciones para el manejo de las instancias de los objetos que se encuentran en la MIB. Las operaciones sobre estas instancias deben tener en cuenta los atributos de cada instancia para preservar su integridad y determinar la ubicación dentro del árbol de la MIB.

Intermediación SNMP.

Por ser SNMP un protocolo de gestión simple no maneja operaciones complejas. El protocolo en sí sólo permite extraer información de una instancia mediante la orden Get o modificar la información que almacena una instancia por medio de la orden Set. Esta cualidad de “simple” hace de SNMP muy popular a la hora de desarrollar una aplicación de gestión. La función prioritaria de SNMP en el sistema SGCST es la de comunicar las aplicaciones de gestión del sistema con el agente SNMP mediante el envío de órdenes, que son ejecutadas por el agente con operaciones de gestión en la MIB, derivadas de las primitivas Get y Set del protocolo SNMP.

Manejo de la Información de gestión.

El procesamiento que realiza el agente SNMP de la información de gestión es simple. Existen operaciones de lectura y escritura basadas en las órdenes que son enviadas por el protocolo SNMP. Estas operaciones permiten gestionar las instancias de los servicios y usuarios en el árbol de la MIB. Todo esto lo realiza el agente gracias a que conoce la ubicación de cada elemento en la MIB mediante el identificador OID (Object Identification).

El agente SNMP básicamente se limita a ejecutar operaciones sobre las instancias existentes de los objetos en el árbol de la MIB. Es importante aclarar que al agente SNMP no le es posible crear nuevas instancias o eliminar las existentes en la MIB, ya que SNMP no soporta estas operaciones. Para adicionar o crear nuevas instancias es necesario volver a compilar la MIB.

Comunicación con la Base de datos.

El agente SNMP del sistema SGCST tiene la propiedad de invocar una aplicación que sirven de interfaz con la base de datos que gestiona el sistema SATplus denominada UPDOCI; dentro de ella se incluyen sentencias SQL que permiten ejecutar operaciones en la base de datos (Ver Figura 4.2).

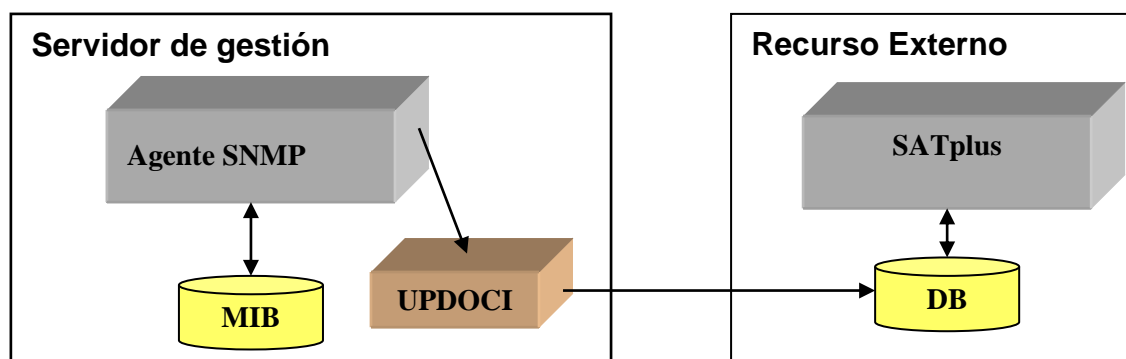


Figura 4.2: Invocación de UPDOCI por el agente SNMP

4.1.3 Generalidades de la Codificación.

La codificación del agente SNMP del sistema SGCST se realizó en lenguaje Visual C++. El código resultante es sencillo y muy eficiente en el uso de funciones y librerías que brinda la plataforma de programación Visual C++.

Básicamente en el agente SNMP se desarrollan operaciones que permiten gestionar las tablas y objetos contenidos en la MIB. Adicionalmente se crean todas las instancias de los objetos sobre un número definido de usuarios para la MIB; esto último lo hace el sistema en tiempo de compilación. Como se mencionó, el agente SNMP puede gestionar las instancias existentes pero no le es posible eliminar ni crear instancias dinámicamente; cuando se requiera eliminar o adicionar instancias es necesario compilar nuevamente la MIB.

Dentro del agente SNMP se crea una función que permite invocar una aplicación que se conecta a una base de datos remota llamada UPDOCI; es una aplicación creada para servir de interfaz con la base de datos del SATplus. El agente le envía a UPDOCI una cadena de caracteres con la información a modificar.

4.1.4 Descripción del código del Agente SNMP

A continuación se destacan algunos apartes del código del agente SNMP del sistema SGCST, con énfasis en fragmentos donde se declaran las operaciones que realizan la gestión sobre objetos de la MIB. Por último se describe la operación que invoca la aplicación externa UPDOCI. El código completo del agente está contenido en el archivo TGSTMIB.C que se encuentra en los anexos de este trabajo de grado. Como complemento se recomienda revisar el archivo TGSTMIB.H, también incluido en los anexos

Archivos de Encabezamiento

Los archivos de encabezamiento empleados en este programa son WINDOWS.H, SNMP.H y TGSTMIB.H. Los dos primeros permiten el soporte para Windows y SNMP respectivamente, en TGSTMIB.H se declaran las funciones y variables del agente SNMP.

```
#include <windows.h>
#include <snmp.h>
#include "tgstmib.h"
```

Las Tablas

En la siguiente parte del código del agente se define el número OID de cada tabla y la rama a la que están colgadas todas las tablas de la MIB en la red. Con el OID se puede localizar cada tabla y cada instancia en el árbol de la MIB. Las tablas creadas en el árbol de la MIB son cuatro; una por cada servicio y una para la gestión de los usuarios.

```
UINT MIB_gst[] = { 2 };
UINT MIB_table_GestionarUsuarios_oid[] = { 1, 0 };
UINT MIB_entry_GestionarUsuarios_oid[] = { 1, 1, 0 };
UINT MIB_table_CodigoSecreto_oid[] = { 2, 0 };
UINT MIB_entry_CodigoSecreto_oid[] = { 2, 1, 0 };
UINT MIB_table_CasilleroVoz_oid[] = { 3, 0 };
UINT MIB_entry_CasilleroVoz_oid[] = { 3, 1, 0 };
UINT MIB_table_TransfLLamada_oid[] = { 4, 0 };
UINT MIB_entry_TransfLLamada_oid[] = { 4, 1, 0 };
```

Los Objetos

Dentro del código se define cada grupo de objetos que integran las cuatro tablas del árbol de la MIB:

```
// Define el número telefónico y el estado de los servicios
UINT MIB_NumeroTelefonico_oid[NUM_MAX_USUARIOS][4];
UINT MIB_Estado_CodigoSecreto_oid[NUM_MAX_USUARIOS][4];
UINT MIB_Estado_CasilleroVoz_oid[NUM_MAX_USUARIOS][4];
UINT MIB_Estado_TransfLLamada_oid[NUM_MAX_USUARIOS][4];
```

```
// Define el código de cada usuario y el valor de cada servicio
UINT MIB_CodigoUsuario_oid[NUM_MAX_USUARIOS][4];
UINT MIB_Numero_CodigoSecreto_oid[NUM_MAX_USUARIOS][4];
UINT MIB_Numero_CasilleroVoz_oid[NUM_MAX_USUARIOS][4];
UINT MIB_Numero_TransfLLamada_oid[NUM_MAX_USUARIOS][4];
```

Se crean las instancias

Mediante ciclos se crean las instancias de cada objeto dentro de la MIB y se inicializan en "0". El código tiene 12 inicializaciones uno por cada objeto, y se le asigna un identificador OID a cada instancia.

```
for(u=0;u<NUM_MAX_USUARIOS;u++){
  for(i=0;i<4;i++){
    if((i!=2)&&(i!=3)){
      MIB_indexGU_oid[u][i]=MIB_entry_GestionarUsuarios_oid[i];
      MIB_NumeroTelefonico_oid[u][i]=MIB_entry_GestionarUsuarios_oid[i];
      MIB_CodigoUsuario_oid[u][i]=MIB_entry_GestionarUsuarios_oid[i];
      MIB_indexCS_oid[u][i]=MIB_entry_CodigoSecreto_oid[i];
      MIB_Estado_CodigoSecreto_oid[u][i]=MIB_entry_CodigoSecreto_oid[i];
      MIB_Numero_CodigoSecreto_oid[u][i]=MIB_entry_CodigoSecreto_oid[i];
      MIB_indexCV_oid[u][i]=MIB_entry_CasilleroVoz_oid[i];
      MIB_Estado_CasilleroVoz_oid[u][i]=MIB_entry_CasilleroVoz_oid[i];
      MIB_Numero_CasilleroVoz_oid[u][i]=MIB_entry_CasilleroVoz_oid[i];
      MIB_indexTLL_oid[u][i]=MIB_entry_TransfLLamada_oid[i];
      MIB_Estado_TransfLLamada_oid[u][i]=MIB_entry_TransfLLamada_oid[i];
      MIB_Numero_TransfLLamada_oid[u][i]=MIB_entry_TransfLLamada_oid[i];
    }
  }
}
```

Acciones del agente

En el código del agente se desarrollan las operaciones Get y Set que van a ejecutarse en la MIB y que son enviadas por el protocolo SNMP. Estas operaciones están contenidas dentro de una instrucción "SWITCH" de la función MIB_leaf_func (ver archivo TGSTMIB.C), el siguiente es el fragmento del código correspondiente:

```

switch ( Action ){
    case MIB_ACTION_GET:
        • // Determina si el objeto tiene acceso de lectura
        if ( MibPtr->Access != MIB_ACCESS_READ && MibPtr->Access != MIB_ACCESS_READWRITE ){
            ErrStat = SNMP_ERRORSTATUS_NOSUCHNAME; // No encuentra el objeto
            goto Exit;
        }
        • //Determina el tipo SINTAX (integer, couter o gauge) del objeto de la MIB
        VarBind->value.asnType = MibPtr->Type;
        switch ( VarBind->value.asnType ){
            case ASN_RFC1155_COUNTER:
            case ASN_RFC1155_GAUGE:
                • //Si es INTEGER almacena el valor del objeto en la variable VarBind
                case ASN_INTEGER: VarBind->value.asnValue.number = *(AsnInteger*)(MibPtr->Storage);
            }
        break;
    case MIB_ACTION_SET:
        • //Determina si el objeto tiene permiso de escritura
        if ( MibPtr->Access != MIB_ACCESS_READWRITE && MibPtr->Access != MIB_ACCESS_WRITE ){
            ErrStat = SNMP_ERRORSTATUS_NOSUCHNAME; // No encuentra el objeto
            goto Exit;
        }
        • //Pregunta el tipo de SINTAX del objeto, si es INTEGER o no
        if ( MibPtr->Type != VarBind->value.asnType ){
            ErrStat = SNMP_ERRORSTATUS_BADVALUE; // No es un valor valido
            goto Exit;
        }
        switch ( VarBind->value.asnType ){
            case ASN_RFC1155_COUNTER:
            case ASN_RFC1155_GAUGE:
                • // Para el caso de INTEGER asigna el nuevo valor al objeto dentro de la MIB
                case ASN_INTEGER: *(AsnInteger*)(MibPtr->Storage) = VarBind->value.asnValue.number;
            }
        break;
    }
}

```

Invocar función UPDOCI

En el código del agente SNMP se desarrolla una función que invoca a la aplicación UPDOCI. El agente arma una cadena concatenando los valores de número telefónico, tipo de servicio y valor del servicio, y los envía a la aplicación UPDOCI.

// Arma cadena a enviar a la aplicación UPDOCI

```

itoa(tel, mensaje, 10);
itoa(Servicio, service, 10);
strcat(mensaje,service);
itoa(ValObject, valor, 10);
strcat(mensaje,valor);

```

// Llama función UPDOCI

```

if((Servicio == ESCS)||((Servicio == ESCV)||((Servicio == ESTLL)||((Servicio == NSCS)||((Servicio == NSCV)||((Servicio == NSTLL)) {
    ShellExecute(NULL, "open", "c:\\temp\\UpdOCI.exe", mensaje, "", SW_SHOWMINNOACTIVE);
}
}

```

4.1.5 Generación de la TGST.DLL.

La TGSTDLL.DLL permite que el servicio del protocolo SNMP reconozca al agente SNMP en el sistema operativo Windows 2000. Para generar la TGSTDLL.DLL es necesario compilar los archivos TGSTDLL.C, TGSTDLL.DEF, TGSTMIB.C, y TGSTMIB.H en el compilador de Visual C++.

Una vez se obtiene TGSTDLL.DLL se incluye en la carpeta SYSTEM del REGEDIT de Windows y así lo reconoce el servicio SNMP de Windows, que es quien le permite funcionar al agente SNMP en el sistema. Los códigos de los archivos utilizados en la compilación del agente SNMP se incluyen como anexos de esta monografía. A continuación se hace una breve reseña de cada uno de los archivos que se utilizan en la compilación de la TGST.DLL.

- TGSTDLL.DEF: Este archivo declara las funciones API extendidas de SNMP que son utilizadas en el archivo TGSTDLL.C.
- TGSTDLL.C: Este archivo contiene las funciones API extendidas de SNMP, estas funciones definen la interfaz entre el protocolo SNMP de Windows (servicio SNMP) y el agente SNMP (ver Tabla 1).
- TGSTMIB.H: En este archivo se declaran las operaciones que serán utilizadas por el agente SNMP, y Define la estructura de los Objetos de la MIB.
- TGSTMIB.C: Este contiene el código del agente en el se desarrollan las operaciones que fueron declaradas en el archivo TGSTMIB.H.

Función	Descripción
SnmpExtensionClose	Libera los recursos y los terminales de operación.
SnmpExtensionInit	Inicializa las Extensiones y servicios del agente SNMP
SnmpExtensionQuery	Ejecuta solicitudes SNMP que involucran variables registradas en el árbol de la MIB del agente SNMP.
SnmpExtensionTrap	Recupera información necesaria para que los servicios en el agente SNMP generen los traps.
SnmpExtensionMonitor	Esta es una función opcional que provee al agente con un panorama de los servicios internos de gestión de objetos definidos en la RFC 1156

TABLA 5: FUNCIONES API EXTENDIDAS DE SNMP

4.2 DESCRIPCION DE LA MIB.

El complemento al análisis del agente SNMP es la base de datos que almacena la información de gestión de los elementos gestionados por el sistema, o sea la MIB. La MIB ha sido diseñada utilizando un esquema de representación común que soporta la interoperabilidad, basado en sentencias definidas por la SIM (*Structure of Management Information*). El código completo de la MIB se encuentra en los anexos de esta monografía en el Archivo TGS.MIB.

Cabe destacar que el Archivo TGS.MIB solo sirve como referencia y no es utilizado para generar el Agente SNMP, es en el archivo TGSTDLL.H donde se define la estructura de los objetos de la MIB que se compilarán con el Agente SNMP (Ver fragmento). La compilación y visualización de TGS.MIB se realiza en el MG-SOFT Browser. TGS.MIB se utiliza para verificar la sintaxis y para comprobar que no existan errores en la descripción de los objetos.

//(Archivo TGSTMIB.H) Define la estructura de los objetos en la MIB y sus parámetros
typedef struct mib_entry

```
{
    AsnObjectIdentifier Oid;
    UINT      Descripción;
    UINT      Id;
    void *    Storage;
    BYTE      Type;
    DWORD     MinVal;
    DWORD     MaxVal;
    UINT      Access;
    UINT      (*MibFunc)( UINT, struct mib_entry *, RFC1157VarBind * );
    struct mib_entry * MibNext;
} MIB_ENTRY;
```

A continuación se estudian las principales características de la MIB del sistema de gestión de servicios telefónicos.

4.2.1 Características Importantes.

La MIB del sistema SGCST está compuesta por tablas dentro de las cuales se definen los objetos que serán gestionados. Actualmente la SMI soporta únicamente una forma de estructuración de datos, y estas son precisamente tablas bidimensionales simples que soportan solamente entradas de valores escalares.

Dentro de las tablas que fueron definidas en la MIB SGCST se encuentran los objetos de gestión. Por conveniencia estos objetos son programados usando las reglas de codificación básicas (BER) asociadas con ASN.1. Aunque no es la forma más compacta o eficiente de codificación, BER es un esquema de codificación estandarizado ampliamente utilizado.

La MIB SGCST se basa en las normas propuestas en la recomendación RFC 1156 (ver anexos). Esta recomendación define los aspectos más importantes en el diseño de las bases de información de gestión MIB y su funcionamiento en plataformas TCP/IP o protocolo de Internet.

Dentro de la recomendación RFC 1156 se definen objetos de “control simple” que son los tipos de objetos utilizados en la MIB SGCST. Estos objetos permiten la abstracción y modificación de sus contenidos, pero no es directamente sobre los objetos que se realizan estas operaciones si no por medio de las instancias de estos objetos. La característica de “control simple” asegura que la manipulación de estas instancias sólo pueden causar daños limitados en el caso de ser violada la integridad de estos elementos de gestión.

El criterio anterior tiene su razón de ser en el hecho de que los protocolos de gestión simple como SNMP no son suficientemente seguros para hacer operaciones de control más poderosas. Pero sin duda la madurez en el diseño de los nuevos protocolos de gestión como SNMP v3 permitirán en el futuro que se puedan definir una nueva clase de objetos mucho más versátiles ejerciendo un control riguroso sobre ellos y así garantizar la integridad y la seguridad de la información y del sistema en general.

4.2.2 Las Tablas de la MIB.

La información en la MIB SGCST está organizada por servicios, lo que significa que existirá una tabla para cada servicio, la cual contiene la información de gestión para cada usuario según el servicio.

Adicionalmente existe una tabla de gestión de usuarios que hace las veces de directorio, pues a partir del número telefónico se establece la posición (index) de la información del usuario en las tablas de servicios. En ellas cada usuario será identificado con su (index) que es tomado como identificador; este parámetro sirve para realizar la búsqueda de toda la información del usuario por medio del identificador OID.

Para la definición de las tablas se involucraron tipos ASN.1 SEQUENCE y SEQUENCE OF de la macro OBJECT-TYPE. El siguiente fragmento tomado de la MIB de la aplicación SGCST muestra cómo fue definida la tabla de gestión de servicios:

```
gstGestionarUsuariosTable OBJECT-TYPE
    SYNTAX SEQUENCE OF GstGestionarUsuariosEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Tabla para adicionar o eliminar usuarios del Servicio de gestión de
        Servicios Telefónicos."
    ::= { tgst 1 }
```

4.2.3 El Árbol de la MIB.

Como se menciona la MIB SGCST está estructurada de tal forma que la información de los usuarios se gestiona definiendo una tabla por cada servicio suplementario del sistema y una tabla adicional de registro de usuarios que almacena el identificador y el (index) de cada usuario. Para esta aplicación se escogieron tres servicios suplementarios: código secreto, casillero de voz y transferencia de llamada. En el capítulo tercero (Diseño) se describen cada una de las tablas y los objetos que conforman la MIB, y se ilustra el árbol de la MIB.

4.2.4 Descripción del Código.

El código de la MIB SGCST está basado en estructuras SIM. La mayor parte del código tiene como fin la declaración de las tablas y los objetos que éstas contienen. A continuación se realiza una descripción parcial del código de la MIB SGCST, el código completo se encuentra en los anexos de esta monografía en el Archivo TGS.MIB.

Etiqueta de Inicio - BEGIN

Esta etiqueta permite declarar el nombre de la MIB indicando el inicio de la misma, es necesario siempre colocar el nombre de la MIB para efectos de compilación. Existen otras etiquetas que permiten declarar datos de los grupos de desarrollo y datos personales de los programadores, pero que no son de carácter obligatorio. El siguiente fragmento indica el inicio de la MIB SGCST:

```
TGST-MIB DEFINITIONS ::= BEGIN
```

Etiqueta IMPORTS

Es una etiqueta muy importante ya que es aquí donde se definen los tipos de estructuras que serán utilizadas dentro de la MIB según las recomendaciones RFC 1155 y 1213 de la SMI. **Enterprises** es la rama de Internet a la que se adiciona la MIB.

```
IMPORTS
    Enterprises          //se pega el árbol de la MIB SGCST
                        FROM RFC1155-SMI
    OBJECT-TYPE          //forma como se definen los objetos de la MIB
                        FROM RFC-1213-SMI
```

Estas cabeceras o etiquetas funcionan como librerías, que permiten utilizar sus estructuras en la definición de los objetos, son de tipo IMPORTS por que se importan de archivos definidos para el agente SNMP y la MIB dentro de ASN 1. La etiqueta IMPORTS define las ramas correspondientes al grupo y al proyecto de al que pertenece la MIB.

```
mgmt OBJECT IDENTIFIER ::= { enterprises 11 } //mgmt y tgst se pegan a enterprises
tgst  OBJECT IDENTIFIER ::= { mgmt 2 }
```

Declaración de las tablas

La siguiente parte del código muestra como se declara cada una de las tablas que componen la MIB. En cada una de estas tablas se definen los objetos que serán gestionados por el agente SNMP. Cada tabla se declara por medio de la estructura OBJECT TYPE, al nombre de cada tabla se le antepone las iniciales GST que corresponden al nombre de la MIB y termina con la palabra TABLE.

```
gstGestionarUsuariosTable OBJECT-TYPE
    SYNTAX SEQUENCE OF GstGestionarUsuariosEntry
    MAX-ACCESS not-accessible // Tipo de acceso de este objeto Tabla
    STATUS current
    DESCRIPTION "Tabla para adicionar o eliminar usuarios del Servicio
        de gestión de Servicios Telefónicos."
    ::= { tgst 1 }
```

Luego de ser declarada la tabla se tiene que definir los tipos de objetos y los parámetros de estos dentro de esta tabla, se hace por medio de un objeto tabla tipo "Entry", y finalmente se define otra tabla tipo "Entry", donde se define la secuencia y orden de los objetos en la tabla. Las estructuras SYNTAX, MAX ACCES, STATUS, DESCRITION e INDEX son descritas en el Capítulo 2, pero en general sólo definen las características del objeto Tabla.

```
gstGestionarUsuariosEntry OBJECT-TYPE // Se definen los parámetros de los objetos
    SYNTAX GstGestionarUsuariosEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION "Parámetros para Gestionar Usuarios entry."
    INDEX { gstUsuariosIndex }
    ::= { gstGestionarUsuariosTable 1 }
```

La anterior forma de declaración se realiza para cada una de las tablas que componen la MIB siguiendo un orden preestablecido. La tabla que se describe en el párrafo anterior es la tabla Gestinarusuarios, en la que se definen dos objetos UsuarioIndex y NumeroTelefonico.

Declaración de los Objetos dentro de las tablas

Una vez son definidos los objetos dentro de las tablas, se declaran y se definen sus propiedades y parámetros para poder ser gestionados, cada objeto en su etiqueta final es relacionado con la tabla y se le asigna un orden correspondiente dentro de la tabla.

```
gstUsuariosIndex OBJECT-TYPE
    SYNTAX  INTEGER (1..255) // Rango de valores posibles de 1 a 255
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION  "Es un valor único para este objeto"
    ::= { gstGestionarUsuariosEntry 1 }

gstNumeroTelefonico OBJECT-TYPE
    SYNTAX  INTEGER (0..255) // Rango de valores posibles de 0 a 255
    ACCESS  read-write
    STATUS  mandatory
    DESCRIPTION  "El estado del servicio.
        Si el valor es 0 el Usuario ha sido eliminado
        Si el valor es diferente de cero existe un usuario"
    ::= { gstGestionarUsuariosEntry 2 }
```

Una vez son declaradas todas las tablas y sus respectivos objetos, se coloca la etiqueta END que indica el final de la MIB.

4.3 APLICACIÓN UPDOCI

En el código del agente SNMP existe una operación que invoca una aplicación externa que sirve de interfaz con la base de datos del SATplus. Esta aplicación se ha denominado UPDOCI, el código completo de este programa está disponible en los anexos de esta monografía.

UPDOCI incluye en su código instrucciones SQL, cuando la aplicación es invocada por el agente SNMP recibe de éste una cadena de caracteres la cual procesa, inmediatamente se conecta a la Base de datos y realiza modificaciones en la tabla de Petición de Servicios del SATplus, donde se almacenan los registros de solicitudes para servicios suplementarios por parte de los clientes o suscriptores de la empresa telefónica.

UPDOCI tiene como función principal la de registrar las modificaciones que se realicen durante la configuración automática de los servicios suplementarios por parte de los usuarios o suscriptores en la base de datos. Cuando el usuario del servicio SGCST ha realizado una modificación en alguno de sus servicios suplementarios, UPDOCI después de ser invocado por el agente SNMP realiza las siguientes operaciones:

- **Conectar la Base de datos:**

UPDOCI crea un objeto de la clase COciCursor para establecer una conexión a la base de datos, definida por el Objeto "db".

// Define una conexión a la base de datos

```
void UpdateMethod( COciDatabase& db ){
    COciCursor      cur( db );
```

- **Procesar Cadena de Caracteres:**

La aplicación recibe del agente SNMP una cadena de caracteres con los datos de usuario, del servicio y la operación a realizar en la base de datos.

// Captura valor de número telefónico

```
strcpy(mensaje, argv[1]); // Pasa toda la cadena a mensaje
strncpy(telefonos,mensaje,6); // Pasa los 6 primeros términos a teléfonos
telefonos[6]='\0';
```

// Captura términos que definen la operación a realizar y el servicio

```
for(m=6;m<8;m++)
    servicios[m-6]=mensaje[m]; //Pasa el valor a servicios desde la posición 6
    servicios[2]='\0';
```

- **Operación a realizar en la base de datos**

Una vez define la opción entre modificar el estado del servicio o modificar el valor del servicio, realiza la instrucción que actúa sobre la base de datos.

// Modifica estado del servicio

```
if(opcion==ESTADO) {
    cur.Exec("UPDATE sersusc SET estado = :val WHERE telefonos = :tel
    AND servicios = :serv", &valor, &telefono, &servicio, NULL );}
```

// Modifica Valor del servicio o el número telefónico

```
if(opcion==NUMERO) {
    cur.Exec("UPDATE sersusc SET numero = :val WHERE telefonos = :tel
    AND servicios = :serv", &valor, &telefono, &servicio, NULL );
    db.Commit(); }
```

4.4 TABLA DE PETICIÓN SERVICIOS SUPLEMENTARIOS.

La tabla PETISERV o tabla de Petición de Servicios Suplementarios se encuentra en la base de datos del SATplus. En ella se definen los parámetros necesarios para realizar el registro de las peticiones de los servicios suplementarios que realizan los usuarios del servicio telefónico. A PETISERV accede la aplicación UPDOCI para registrar las peticiones de los usuarios del sistema SGCST (ver Tabla 6).

COLUMNA	TIPO	LONGITUD	DESCRIPCION	No
PESEDEPA	NUMBER	4	Departamento del país	1
PESELOCA	NUMBER	6	Localidad	2
PESEPETI	NUMBER	8	Número de la petición	3
PESESUS	NUMBER	8	Suscripción	4
PESESERV	NUMBER	4	Servicio	5
PESENUSE	NUMBER	8	Número del servicio	6
PESETIPE	NUMBER	2	Tipo de petición	7
PESETISE	NUMBER	2	Tipo de servicio especial	8
PESECODE	NUMBER	2	Conexión / Desconexión	9
PESEFEPR	DATE		Fecha de Conexión / Desconexión	10
PESEFEAP	DATE		FECH_AP	11
PESEFEAT	DATE		FECH_AT	12
PESEFEVS	DATE		FECH_VS	13
PESEESPE	NUMBER	2	Estado de la petición	14
PESECARA	NUMBER	8	Carta de radicación	15
PESENUME	NUMBER	8	Número de petición asociada	16
PESEOBSV	VARCHAR2	70	Observación	17
PESECOSE	VARCHAR2	4	Código secreto	18
PESETALL	NUMBER	8	Número Transferencia	19
PESECAVO	NUMBER	4	Número mensajes	20
PESEFEOS	DATE		Fecha de la orden de servicio	21
-PESENUOR	NUMBER	8	Número de la orden	22
PESEOPER	NUMBER	6	Código del Operador	23
PESEVAPE	NUMBER	13	Valor Petición	24

TABLA 6: PETICION DE SERVICIOS SUPLEMENTARIOS

Esta Tabla fue tomada como guía de la base de datos del SATplus de la empresa TeleArmenia, de la tabla Original sólo varía por la adición de los parámetros PESETALL y PESECAVO, que serían los parámetros donde se almacenarían el número de transferencia de llamada y el número de mensajes del casillero de voz parámetros que no tiene la PETISERV de TeleArmenia.

4.4.1 Parámetros que registra UPDOCI

Los parámetros que modifica la aplicación UPDOCI en la tabla PETISERV, y que son del interés del sistema de gestión SGCST son:

- PESETISE: Código del servicio suplementario.
- PESECODE: Acción sobre el servicio, Conexión o Desconexión.
- PESENUSE: Numero telefónico del usuario.
- PESECOSE: Numero de código secreto
- PESETALL: Numero de transferencia de llamada.
- PESECAVO: Numero de mensajes del casillero de voz

UPDOCI siempre modifica tres de estos parámetros según sea el servicio y lo que se desee registrar así:

- Modificar Código secreto: PESETISE, PESENUSE y PESECOSE
- Modificar Casillero de voz: PESETISE, PESENUSE y PESECAVO
- Modificar Transferencia de llamada: PESETISE, PESENUSE y PESETALL
- Desconectar o Conectar un servicio: PESETISE, PESENUSE y PESECODE

En el momento de realizarse el registro de la petición existen campos que toman valores automáticamente mediante “Triggers”, como:

- El Número de la petición
- La Fecha de la orden de servicio
- El Tipo de petición
- El Valor Petición
- El Número de la orden

Todos los parámetros que maneja la aplicación UPDOCI son datos numéricos y dependen necesariamente del servicio suplementario. En la tabla PETISERV se definen las características de cada uno de los parámetros que maneja la aplicación UPDOCI.

5. MODULO DE APLICACIÓN

El modulo de aplicación es importante para el proyecto por constituir la herramienta que utilizaran los usuarios para el manejo del sistema SGCST, y hacer posible de forma amigable, clara y precisa el mostrar la gestión realizada en el sistema. En este capítulo se describe de forma general la estación de gestión, haciendo énfasis en analizar las interfaces de los usuarios del sistema SGCST, y su comunicación con el modulo de gestión SNMP.

5.1 LA ESTACIÓN DE GESTIÓN

Una estación de gestión es un dispositivo que sirve como interfaz entre los usuarios de la red y el sistema de gestión de red. En general una estación de gestión debe tener como mínimo las siguientes características:

- Un conjunto de aplicaciones de gestión para análisis de datos, configuración, estadísticas, etc.
- Un grupo de interfaces a través de las cuales los usuarios puedan gestionar los elementos de gestión del sistema.
- La capacidad de trasladar los requerimientos de los usuarios del sistema a los elementos de gestión que conforman la red.
- Una base de datos de información de gestión de la red o MIB, que contiene todas las instancias de los objetos gestionados en el sistema.

El objetivo de este capítulo se centra en describir las interfaces de la estación de gestión para cada uno de los usuarios del sistema SGCST, y su funcionamiento conjunto con las aplicaciones de gestión. Se hace un diagrama de interfaces para el Usuario web y otro para el Administrador del sistema, en ellos se incluyen las páginas desarrolladas en PHP donde se utilizan llamadas a las ordenes del protocolo SNMP, que son atendidas por el agente SNMP, quien a su vez ejecuta las operaciones correspondientes en la MIB (ver Figura 5.1).

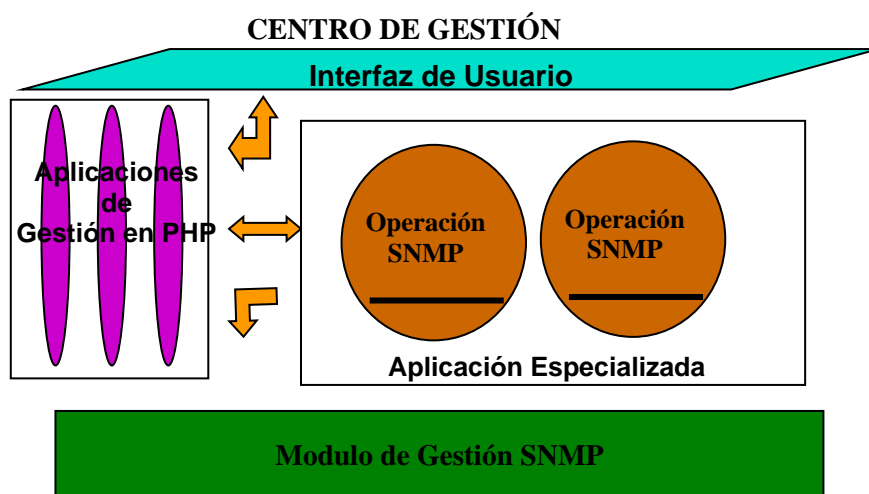


Figura 5.1: Esquema De Una Estación De Gestión

5.1.1 La Estación de Gestión del Sistema SGCST

El modulo de aplicación y el modulo de gestión conforman la denominada estación de gestión. La estación de gestión del sistema SGCST posee características similares de una estación de gestión SNMP estándar (ver Figura 5.2).

Componentes de la estación de gestión del sistema SGCST:

- Un conjunto de funciones de consulta y modificación que permiten el manejo de la información de los usuarios y de los servicios del sistema, las llamamos aplicaciones de gestión y están desarrolladas en páginas PHP.
- Un conjunto de páginas desarrolladas en PHP y HTML, para el fácil manejo del sistema SGCST por parte de los usuarios y administradores. Las interfaces le permiten al administrador monitorear, Gestionar y configurar tanto a Usuarios como a los servicios del sistema. Los usuarios a través de sus interfaces podrán gestionar la configuración de sus servicios telefónicos suplementarios.
- El protocolo SNMP, comunica las aplicaciones de gestión con el agente SNMP, en la comunicación se intercambia información de gestión y de control. Esta propiedad es sujeta a la estandarización de SNMP.

- El agente SNMP recibe las órdenes enviadas desde las aplicaciones de gestión vía protocolo SNMP, y las ejecuta en la base de información de gestión MIB.
- El agente puede integrar aplicaciones externas de gestión a la estación de gestión.
- La base de información de gestión MIB, es el lugar donde se almacena la información de gestión de los elementos del sistema SGCST.

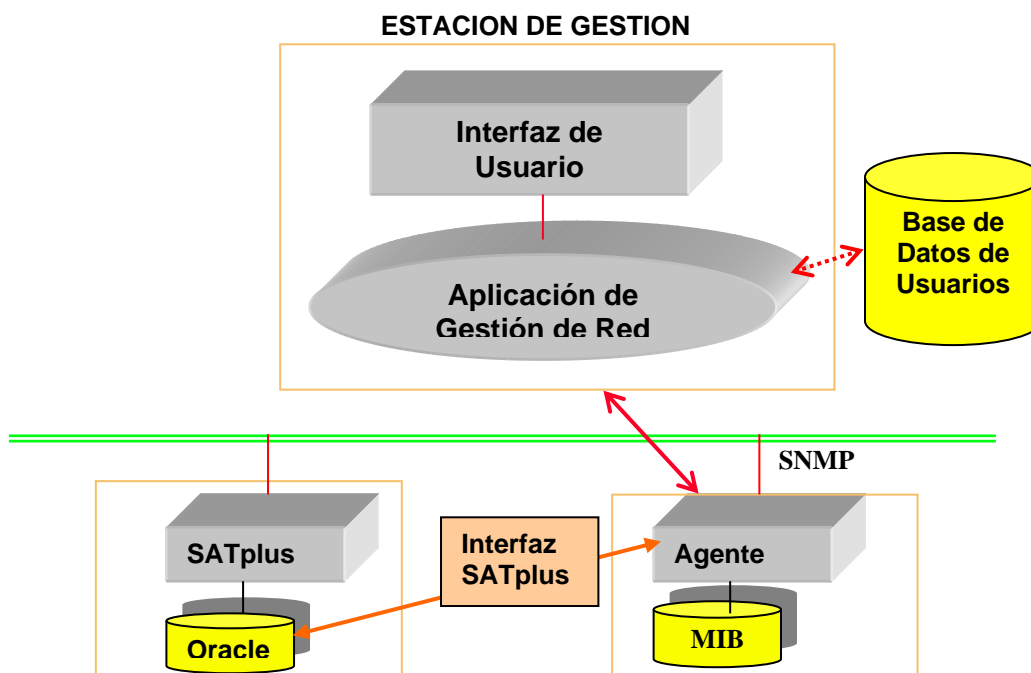


Figura 5.2: Estación de Gestión SGCST

El agente de gestión SNMP responde a las solicitudes de información o de acciones enviadas desde la estación de gestión, y puede proveer de manera asincrónica a la estación de gestión de información muy importante pero no solicitada. La estación de gestión y el agente están comunicados por el *protocolo de gestión de red SNMP*, el cual incluye las siguientes capacidades:

- *Get*: Le permite a la estación de gestión obtener los valores de objetos en el agente.
- *Set*: Permite a la estación de gestión fijar los valores de objetos en el agente.
- *Trap*: Permite a un agente notificar a la estación de gestión de los eventos significativos.

Algunos RFCs relacionados con SNMP v1 son:

Nº RFC	Descripción
RFC 1155	Estructura e identificación de la información de gestión basada en los protocolos de Internet TCP/IP
RFC 1156	Base de información de gestión MIB, para redes de gestión basadas en los protocolos de Internet TCP/IP
RFC 1157	Protocolo de gestión simple SNMP v1.
RFC 1212	Definiciones para la MIB
RFC 1213	Base de información de gestión MIB-II, para redes de gestión basadas en los protocolos de Internet TCP/IP

TABLA 7: RFC'S RELACIONADOS CON SNMP V1

5.1.2 Conexión a la base de datos de Usuarios.

Si bien no es un elemento que esta dentro de la estación de gestión, la base de datos de usuarios es muy importante, en ella se almacena la información de los usuarios del sistema, además es muy utilizada por algunas de las aplicaciones de gestión de la estación de gestión (ver Figura 5.2).

Las funciones de conexión, consulta de datos y modificación de datos en la base de datos de usuarios, que son utilizadas por las aplicaciones de gestión tanto para el usuario cliente, como para el usuario Administrador, están desarrolladas en la página AccesarOracle.php, que fue elaborada en PHP con código embebido SQL. En el siguiente fragmento se describe la función que permite ingresar datos en la base de datos:

// La función recibe los parámetros y la tabla a modificar en el arreglo \$SQLD

```
function InserUpdDatos($sqld) {
    $actualiza=0;
```

// Se Conecta a la base de datos

```
    $conn=Ora_Logon("system", "manager"); printoraerr($conn);
    $cursor=ora_open($conn); printoraerr($cursor);
    ora_parse($cursor,$sqld,0); printoraerr($cursor);
```

// Ejecuta la operación de modificar los datos

```
    ora_exec($cursor);
    $actualiza=1;
```

// Cierra la conexión con la base de datos

```
    ora_close($cursor);
    ora_logoff($conn);
    return $actualiza;
}
```

5.2 MODELO DE APLICACIÓN PARTE USUARIO.

El usuario del sistema SGCST va tener la posibilidad de gestionar y configurar sus servicios telefónicos suplementarios a través de muy pocas interfaces. En una sola página Web va tener reunidos todos los servicios que tiene disponibles para ser configurados, y de una forma sencilla podrá realizar activaciones, desactivaciones y modificaciones con total comodidad.

5.2.1 Árbol de interfaces del Usuario.

El siguiente es el esquema general de la aplicación del sistema de gestión telefónica para el usuario:

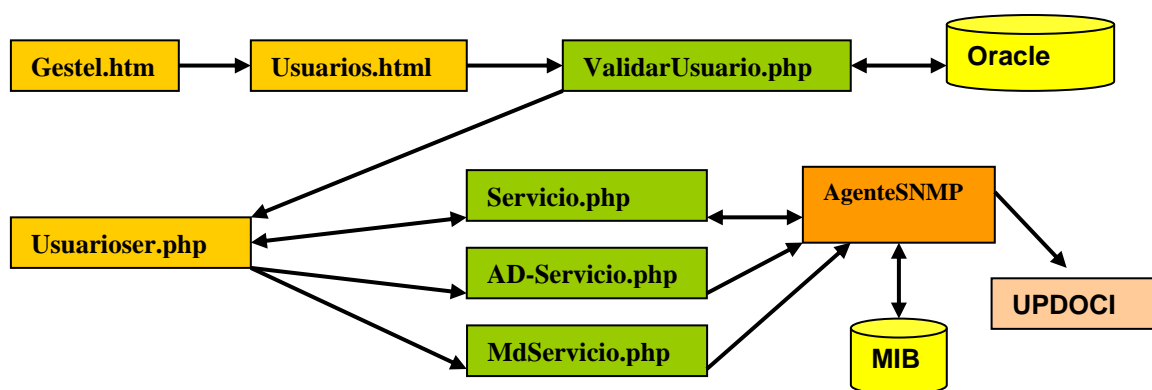


Figura 5.3: Diagrama de Interfaces para el Usuario

Acceso al Sistema

Para realizar la validación de los usuarios del sistema, se utiliza una interfaz en HTML la página **Usuarios.html**, donde el usuario Ingresa los datos (Teléfono y Clave), esta página invoca a la aplicación de gestión Validar Usuario, desarrollada en la página **ValidarUsuario.php** elaborada en PHP. Si la validación es exitosa el sistema carga la página gestión de servicios del usuario **Usuarioser.php**, esta página esta hecha en HTML con código embebido de PHP (Ver Figura 5.4).

5.2.2 Descripción de Páginas

Gestel.htm: Esta es la página de inicio del servicio SGCST para el usuario creada en HTML, en ella el usuario encuentra tres botones, que le brindan tres opciones así:

- **Usuarios:** Este botón le permite cargar la página **Usuarios.html**, donde puede ingresar sus datos, para validarse en el sistema.
- **Descripción:** Este botón lo envía a la página **Descripción.html**, creada completamente en HTML, donde se describe el funcionamiento del servicio de gestión telefónica por Internet SGCST.
- **Home:** Este botón envía a la página principal de la empresa telefónica o la entidad que brinda el servicio SGCST.

Las páginas **Usuarios.html**, **Descripción.html** y **Usuarioser.php**, tienen disponibles los tres botones anteriores en su lado izquierdo.

Usuarios.html: Esta página es un formulario donde el usuario ingresa los datos para poder acceder a su página de configuración de servicios, y está completamente elaborada en HTML. El formulario utiliza dos variables: **Password** y **Teléfono**, que son enviadas a la página **ValidarUsuarios.php**, donde se realiza la validación previa consulta de los datos en la base de datos. El siguiente fragmento muestra las dos variables y la página a la que se le envían.

```
<input type="text" name="telefono" size="8" maxlength="8">  
<input type="password" name="password" size="8" maxlength="8">  
<form name="form1" method="post" action="ValidarUsuarios.php">
```

ValidarUsuarios.php: Página elaborada en PHP, en ella se realiza una consulta a la base de datos Oracle del sistema invocando la función `Consultarpassword` que se encuentra en la página `AccesarOracle.php`, y se verifica que el Número telefónico y la clave sean correctos.

Esta página ejecuta la operación de validar usuario de forma completamente transparente al usuario, una vez validado el usuario se invoca la Página **Usuarioser.php**, que es la página principal del usuario. El siguiente fragmento de código indica la acción a realizar en caso de éxito o error en la validación:

//Invoca a la página AccesarOracle.php

```
include ("AccesarOracle.php");
$et=0;
$ep=0;
$sql="select id from gesteluser where telefono = '$telefono'";
```

// Llama función consultarpasword.php

```
$id=ConsultarPassword($sql);
```

//Si es exitosa la validación llama a Usuarioser.php – página principal del usuario.

```
if (($et==1) && ($ep==1))
echo "<body bgcolor=#FFFFFF" text=#000000" background=fondo.jpg"
onLoad=window.setTimeout(location.href='usuarioser.php?idu=$id[0]',0)
>";
```

//Si no son validos los datos envía a Usuarios.htm – vuelve a preguntar datos.

```
else
echo "<body bgcolor=#FFFFFF" text=#000000" background=fondo.jpg"
onLoad=window.setTimeout(location.href='usuarios.htm',10000)>";
```

Usuarioser.php: Es la página principal del usuario, en ella el usuario puede gestionar y configurar los servicios telefónicos suplementarios que tiene disponibles. **Usuarioser.php** invoca a **Servicio.php**, quien realiza una consulta en la MIB donde obtiene los datos de los servicios del usuario, necesarios para generar la página principal.

En su página principal el usuario dispone de un listado dinámico, de este listado escoge el servicio que desea gestionar, y estando en el “Cajón” del servicio puede ver la descripción del servicio, además dispone de tres botones posibles que le permiten: Activar, Desactivar y Modificar el servicio. Esta página esta desarrollada en código HTML con PHP embebido, e incluye animaciones en **FLASH** tipo Scroll (Ver Figura 5.4).

Flash es una herramienta avanzada para el desarrollo de animaciones Web, es relativamente fácil de utilizar y muy poderosa, con unos resultados en términos de rendimiento muy favorables al generar archivos de tamaño pequeño, lo que permite una descarga mucho más rápida de las páginas Web. En la página **Usuarioser.php** se tiene una aplicación Flash que permite desplazamiento de imágenes y texto que dan sensación de movimiento, esto se logra mediante la propiedad **texto.scroll** de Flash.



Figura 5.4: Página de Gestión de servicios

5.2.3 Casos Posibles de Usuarioer.php

Activar Servicio: Si el usuario desea activar un servicio que este desactivo, oprime el botón activar, en la página **Usuarioer.php** se invoca la aplicación que permite activar los servicios, que está desarrollada en la página **AD-Servicio.php**, utilizando únicamente instrucciones en PHP. El siguiente es el fragmento de código donde se indica la operación activar o desactivar servicio:

```
<?php
// OID para los tres servicios
$oidecs = ".1.3.6.1.4.1.11.2.2.1.2";
$oidcv = ".1.3.6.1.4.1.11.2.3.1.2";
$oidetll=".1.3.6.1.4.1.11.2.4.1.2";

//Define el servicio para el usuario indicado (idus)
//Si es Código secreto
if ($servicio==0) $adservicios=$oidecs.$idus;

//Si es Casillero de voz
if ($servicio==1) $adservicios=$oidcv.$idus;

//Si es Transferencia de llamada
if ($servicio==2) $adservicios=$oidetll.$idus;

//Modifica el estado del servicio por el valor en $AD, SET en la MIB
snmpset("127.0.0.1", "public", "$adservicios", "i", $AD);
?>
```

Desactivar Servicio: Si el usuario desea desactivar un servicio que este activo, oprime el botón desactivar, la página **Usuarioser.php** invoca la aplicación que permite desactivar los servicios, que esta desarrollada en la página **AD-Servicio.php** se utiliza únicamente instrucciones en PHP (ver fragmento de código en Activar servicio).

Modificar Servicio: Si el usuario desea modificar un servicio, oprime el botón modificar en la página **Usuarioser.php**, esta le solicita el nuevo valor del servicio, este lo llena en la caja de texto, al oprimir "Ok", se invoca la aplicación de gestión que permite modificar los servicios. Esta aplicación fue desarrollada en la página **MdServicio.php** solamente usando instrucciones en PHP. El siguiente es el fragmento de código donde se indica la operación modificar servicio:

```
<?php
// IOD para los tres servicios
$oidnacs=".1.3.6.1.4.1.11.2.2.1.3";
$oidnmcv=".1.3.6.1.4.1.11.2.3.1.3";
$oidntll=".1.3.6.1.4.1.11.2.4.1.3";
// Busca el servicio a modificar, para el Usuario indicado ( idus).
// Si es Código secreto
if ($servicio==0) $mdservicios=$oidnacs.$idus;
// Si es Casillero de voz
if ($servicio==1) $mdservicios=$oidnmcv.$idus;
// Si es Transferencia de llamada
if ($servicio==2) $mdservicios=$oidntll.$idus;
// Modifica el valor del servicio por el valor en $MS, orden SET en la MIB.
snmpset("127.0.0.1", "public", "$mdservicios", "i", $MS);
?>
```

Servicio.php: Una vez que **ValidarUsuario.php** valida los datos, invoca a **Usuarioser.php** y le pasa el **Index** o identificador del usuario (Indica la posición del usuario en la tabla de la MIB), esta a su vez invoca a **Servicio.php** que con el **Index** del usuario arma las consulta a las tablas de servicios en la MIB, y los datos son enviados nuevamente a **Usuarioser.php** que carga los datos del usuario. El siguiente es el fragmento donde se indica la operación de consulta de los datos de los servicios del usuario en la MIB:

```
$EstadoCodigoSecreto= snmpget("127.0.0.1", "public", "$oidecs.$idus");
$NumeroCodigoSecreto= snmpget("127.0.0.1", "public", "$oidnacs.$idus");
$EstadoCasilleroVoz= snmpget("127.0.0.1", "public", "$oidecv.$idus");
$MensajesCasilleroVoz= snmpget("127.0.0.1", "public", "$oidnmcv.$idus");
$EstadoTransferenciaLLamada= snmpget("127.0.0.1", "public", "$oidetll.$idus");
$NumeroTransferenciaLLamada= snmpget("127.0.0.1", "public", "$oidntll.$idus");
```


5.3 MODELO DE APLICACIÓN PARTE USUARIO ADMINISTRADOR.

El Administrador del sistema SGCST va tener la posibilidad de gestionar a los usuarios del sistema por medio de muy pocas interfaces de fácil uso. En su página principal el Administrador tiene la oportunidad de Adicionar o Eliminar un Usuario del sistema de forma sencilla.

5.3.1 Árbol de interfaces del Administrador

El siguiente es el esquema general de la aplicación del sistema de gestión telefónica para el usuario Administrador:

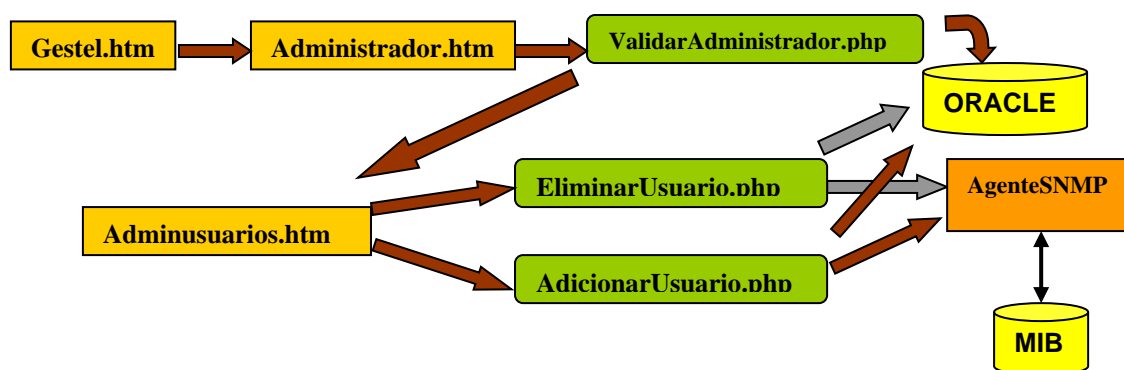
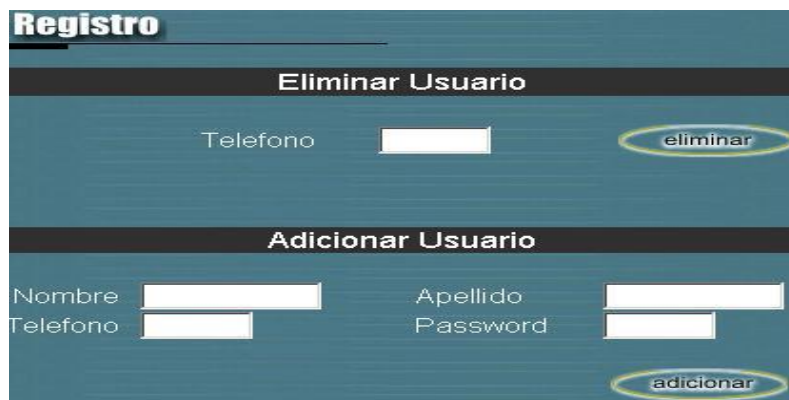


Figura 5.5: Diagrama de Interfaces para el Administrador

Acceso al Sistema

Para realizar su validación el o los Administradores del sistema utilizan la página en HTML **Administrador.html**, donde el Administrador Ingresa los datos (identificador y clave), esta página invoca a la aplicación de gestión **ValidarAdministrador.php**, que esta elaborada en PHP, si la validación es exitosa el sistema carga la página gestión de usuarios **Adminusuarios.htm**, página elaborada en HTML (Ver Figura 5.6).



The image shows a web interface titled 'Registro'. It contains two main sections: 'Eliminar Usuario' and 'Adicionar Usuario'. The 'Eliminar Usuario' section has a text input field labeled 'Telefono' and a button labeled 'eliminar'. The 'Adicionar Usuario' section has four text input fields: 'Nombre', 'Apellido', 'Telefono', and 'Password', and a button labeled 'adicionar'.

Figura 5.6: Página de Gestión de Usuarios

5.3.2 Descripción de Páginas

Gestel2.htm: Esta es la página de inicio del servicio SGCST para el Administrador creada en HTML, la única diferencia con la Gestel.htm del usuario, es que cuenta con un botón adicional (Administrador), que le envía a la página **Administrador.html**, donde puede ingresar sus datos y validarse en el sistema para que sea cargada su página principal la **Admiusuarios.htm**.

Administardor.html: Es la página que recibe los datos del Administrador para validarse y acceder al sistema. Esta Página es un formulario completamente elaborado en HTML, que utiliza dos variables: **Login** y **Password**. El siguiente fragmento muestra las dos variables y la página a la que se le envían los parámetros.

```
<input type="text" name="Login" size="8" maxlength="8">  
<input type="password" name="password" size="8" maxlength="8">  
<form name="form1" method="post" action="ValidadorAdministrador.php">
```

ValidadorAdministrador.php: Página elaborada en PHP, en ella se lleva a cabo una consulta a la base de datos Oracle del sistema invocando la función Consultarpassword que se encuentra en la página AccesarOracle.php, allí se verifica que el Identificador y la Clave del administrador sean correctos.

Esta página ejecuta la operación de Validar Administrador de forma completamente transparente al Administrador. Una vez validado el Administrador se carga la Página **Adminusuarios.htm**. El siguiente fragmento de código indica la acción a realizar en caso de éxito o error en la validación:

```
//Si es exitosa la validación carga Adminusuarios.htm – página principal del Administrador
if (($el==1) && ($ep==1))
echo "<body bgcolor=\"#FFFFFF\" text=\"#000000\" background=\"fondo.jpg\"
onLoad=window.setTimeout(\"location.href='adminusuarios.htm'\",0)>";
//Si no son validos los datos carga Administrador.htm – vuelve a preguntar datos
else
echo "<body bgcolor=\"#FFFFFF\" text=\"#000000\"background=\"fondo.jpg\"
onLoad=window.setTimeout(\"location.href='administrador.htm'\",10000)>";
```

5.3.3 Casos Posibles

Eliminar Usuario: En la página principal del Administrador **Adminusuarios.htm**, si este desea eliminar un usuario del servicio, ingresa el número telefónico correspondiente y oprime el botón eliminar. Aparece una ventana de confirmación. Si oprime “aceptar” la página **Adminusuarios.htm** invoca la aplicación que permite eliminar usuarios, que está desarrollada en la página **EliminarUsuario.php** hecha completamente en PHP. Esta a su vez invoca la función **InserUpdDatos** que se encuentra en la página **AccesarOracle.php**, para realizar los cambios en la base de datos.

Si el usuario ha sido correctamente eliminado se carga la página con el aviso “El usuario ha sido Eliminado”. Si no, se despliega el mensaje “El usuario no existe “. En ambos casos las páginas tienen el botón “Regresar” que les permite regresar a la página principal del administrador. El siguiente es el fragmento en PHP de la página **EliminarUsuario.php**, donde se realiza la operación de Eliminar usuario:

```
//Llama a la función que se conecta con la base de datos Oracle
include("AccesarOracle.php");
$ue=0;
//Realiza la consulta del Index(id) del usuario con el número telefónico
$sql="select id from gesteluser where telefono = '$telefonoe'";
$id=ConsultarPassword($sql);
//Verifica que existe el usuario y que el index (id) es distinto de cero
if (($id[0]!="") && ($id[0]!="0")) {
    $ue=1;
    $oidt=".1.3.6.1.4.1.11.2.1.1.2";
```

//Realiza la eliminación en la MIB y en la Base de datos

```
snmpset("127.0.0.1", "public", "$oidt.$id[0]", "i", "0");
$sql="update gesteluser set id = 0 where telefono = '$telefonoe'";
```

//Llama a la función InserupDatos

```
$iu=InserUpdDatos($sql);
```

Adicionar usuario: Estando en su página principal, si el Administrador desea adicionar un nuevo usuario al servicio, ingresa: el número telefónico, Nombre, Apellido y una clave; luego oprime el botón “Adicionar”, aparece un aviso de confirmación, al oprimir “Aceptar” la página **Adminusuarios.htm** invoca la aplicación que permite Adicionar usuarios desarrollada en la página **AdicionarUsuario.php** hecha completamente en PHP. Ésta a su vez invoca la función InserUpdDatos que se encuentra en la página AccesarOracle.php, para realizar los cambios en la base de datos (Ver fragmento), como respuesta del sistema existen estas posibilidades:

- Si el usuario es adicionado correctamente se carga la página con el aviso “Usuario Adicionado”, y con el botón “Regresar” se regresa a la página principal.
- Si se ingresa uno de los datos del usuario mal el sistema limpia los datos de las casillas al oprimir “aceptar”.
- Si el usuario que se ingresa ya esta registrado, se carga la página con el aviso “Usuario ya existe”.

El siguiente es el fragmento en PHP de la página **AdicionarUsuario.php**, donde se realiza la operación de Adicionar usuario:

// Llama a la función que se conecta a la base de datos Oracle

```
include("AccesarOracle.php");
$ue=0;
```

// Realiza la consulta del usuario según el teléfono que ingreso y verifica si existe o no

```
$sql="select nombre from gesteluser where telefono = '$telefonoa'";
$u=ConsultarPassword($sql);
if($u[0]==""){
```

// Si Usuario no existe, crea un nuevo usuario

```
$ue=1;
```

// OID tabla MIB - Gestionar Usuarios

```
$oidt=".1.3.6.1.4.1.11.2.1.1.2";
```

// Realiza una lista con todos los usuarios en la MIB – Gestionar usuarios

```
$listausuarios = snmpwalk("127.0.0.1", "public", "$oidt" );
$numerousuarios=count($listausuarios);
```

```
// Busca un lugar vacío en la tabla – parámetro teléfono = 0
for ($i=0; $i < $numerousuarios; $i++) {
    if($listausuarios[$i] == 0){
        $id=$i+1;
// Cuando lo ubica almacena el nuevo numero telefónico en la MIB – Gestionar usuarios
        snmpset("127.0.0.1", "public", "$oidt.$id", "i", "$telefonoa");
        $i = $numerousuarios;
    }
}
// Guarda los valores del nuevo usuario en la base de datos según el index de la MIB
$sql="insert into gesteluser values
($id, '$nombre', '$apellido', $telefonoa, '$password')";
$iu=InserUpdDatos($sql);
}
```

6. RECOMENDACIONES Y CONCLUSIONES

EL PROTOCOLO SNMP.

SNMP es un protocolo que es capaz de satisfacer muchas expectativas y cumplir muchas misiones de gestión, pero debemos ser claros a este protocolo no se le pueden exigir funciones que resultan irreales para sus características.

SNMP es una plataforma de gestión poco exigente, esto quiere decir que el volumen de recursos del sistema necesarios para soportarlo no es muy grande, normalmente el protocolo SNMP no ocasiona mayor congestión en la Red. Hay que tener en cuenta que debe existir un control sobre el numero de traps generados por el protocolo SNMP para evitar que un numero muy grande ocasionen problemas de trafico.

Para SNMP cada vez que sea necesario reiniciar el sistema, se hace indispensable almacenar la información de gestión en una base de datos, esto debido a que la información que se tenga en la MIB se vacía y se puede perder. Seria muy importante tener un mecanismo Software, que permita arrancar nuevamente el sistema con la información actualizada, para lograr esto es necesario implementar un sistema bidireccional de intercambio de información entre la base de datos diferente de la MIB y el agente SNMP.

Una gran desventaja de SNMP está en el hecho de que este protocolo es sumamente estático. Esto por que no es capaz de crear o eliminar objetos ni instancias en la MIB de forma dinámica, se necesita compilar nuevamente la MIB para adicionar o eliminar un objeto o una instancia. CMIP es un protocolo que si permite crear nuevas instancias y nuevos objetos dinámicamente, pero es una plataforma extremadamente compleja, con muy poca acogida. Se espera que en el futuro las nuevas generaciones de SNMP lo hagan mucho mas dinámico y aumente su aceptación.

Es importante buscar alternativas al sistema Operativo en los futuros proyectos de SNMP. En la aplicación SGCST se utiliza Windows 2000, con algunas propiedades adicionales de Windows NT, entre ellas el servicio SNMP, que es fácil de configurar, de este servicio se encuentra mucho material de soporte en la Internet. Pero la alternativa de Linux es válida, también es necesario para este sistema operativo instalar el componente UCD-SNMP que lo pone al nivel del servicio SNMP de Windows NT (ver <http://net-snmp.sourceforge.net/>).

EL AGENTE SNMP.

Para crear el agente SNMP es necesario definir claramente la MIB del sistema, tomando en cuenta una correcta relación entre los diferentes objetos que en ella se encuentran. Es muy importante a la hora de programar el agente tener presente el número de objetos y su posición en el árbol de la MIB, definir que objeto es el primero y cual es el último, este último debe apuntar a null (vacío). Esto es necesario para poder desarrollar de manera adecuada las ordenes SET y GET, por que estas operaciones realizan la búsqueda los elementos en la MIB uno por uno en todo el árbol de la MIB.

En futuras aplicaciones se hace importante buscar alternativas más “fáciles” y eficientes de generar el agente SNMP. En la aplicación se hizo necesario desarrollar el agente “a pie”, solamente se utilizó el archivo de la MIB como referencia para cada uno de los objetos que la conformaban y para visualizarla compilada en el Browser; Las operaciones se desarrollaron con el método de ensayo y error, gracias a ejemplos de referencia encontrados en Internet. Resulta adecuado estudiar y adquirir compiladores de MIB que generen de forma total o parcial el agente SNMP teniendo únicamente el archivo MIB.

BASE DE GESTIÓN DE INFORMACIÓN MIB.

Es importante diseñar la MIB de forma correcta, solo así será posible obtener un agente SNMP que cumpla con los objetivos que se le proponen. Para lograr esto resulta indispensable seguir las normas propuestas en los RFC's 1213, 1155 y 1156. Pero no se debe despreciar el material existente como referencia y los ejemplos de MIB's ya creadas con sus respectivos agentes. Son de mucha utilidad y le brindan un soporte que es difícil de encontrar por otras fuentes.

Sería interesante analizar y estudiar a fondo las nuevas MIB-II, que son bases de información que soportan el monitoreo y la gestión de red desde la capa de transporte hacia abajo, tratando de resolver algunos interrogantes, por ejemplo: cómo se consigue establecer las conexiones, cómo son enrutados los paquetes, y problemas similares. Para gestionar recursos como servidores de correo e impresoras, la MIB necesita ser mejorada. Lo que SNMP y la MIB-II proveen es una base para ganar experiencia en la ejecución de gestión de redes heterogéneas, así como un conjunto limitado de características de gestión de uso general.

LA ESTACIÓN DE GESTIÓN

Sin duda entender la estación de gestión como la forma más adecuada de implantar un sistema de gestión nos da un gran aval para tener éxito en este proyecto, aspectos en los que cabe resaltar son los siguientes:

- No existen recomendaciones específicas en los estándares acerca del número de agentes de gestión o la relación de estaciones de gestión a agentes, esto queda en libertad del desarrollador.
- En general, es prudente tener al menos dos sistemas capaces de realizar la función de estación de gestión, para proveer redundancia en caso de fallas.

- Otro problema a resolver es de tipo práctico, y está relacionado con la cantidad de agentes que puede manejar una sola estación de gestión. Gracias a que SNMP es relativamente “simple”, ese número puede ser bastante grande, hasta los centenares.
- La estación de gestión se puede ver como el “sistema operativo” de las Aplicaciones de Gestión.
- La estación de gestión se valora por los servicios que ofrece a las aplicaciones de gestión a través de un conjunto de APIs.
- La estación de gestión ofrece a las Aplicaciones de Gestión una única infraestructura común de acceso a la información de gestión de los recursos
- Hacen más simple el desarrollo de aplicaciones específicas de gestión.
- Permiten a los administradores de redes configurar a “su medida” las herramientas de gestión con las que van a trabajar.
- Posibilitan la integración de aplicaciones de gestión específicas, y de diferentes fabricantes, en una única aplicación global de gestión.

LA COMUNICACIÓN ENTRE LA BASE DE DATOS DEL SATplus Y EL SISTEMA SGCST

Basados en recomendaciones recibidas y comprobadas durante el desarrollo del proyecto se concluyó lo siguiente:

La comunicación entre la base de datos del SATplus y la aplicación del sistema SGCST, se debe realizar directamente con la base de datos mediante una aplicación que sirva de interfaz entre estos dos elementos, por las siguientes razones:

- Fácil manejo de tablas por medio de SQL, iniciando una conexión remota a la base de datos.

- El SATplus es una aplicación propietaria que no brinda facilidades de comunicación con otras aplicaciones.
- Es mucho más sencillo realizar pruebas y mejoras que no dependan de una plataforma o sistema cerrado.

Una gran ventaja para el proyecto resulta la propiedad del agente SNMP que permite invocar aplicaciones externas cuando se requiera como la que nos permite comunicarnos con la base de datos del SATplus. En este proyecto se desarrolla una aplicación denominada UPDOCI que cumple con ese objetivo.

LA COMUNICACIÓN CON LA TERMINAL DE OPERACIÓN Y MANTENIMIENTO OMT.

Para el futuro y previendo la continuidad de este proyecto, es necesario tener en cuenta las posibilidades que se tienen para comunicar aplicaciones de gestión con las centrales telefónicas. Es importante:

- Conocer la central telefónica y las facilidades de comunicación que esta brinda para la terminal OMT, como es el caso de la comunicación con RS-232 en el puerto VT-100 de las centrales ITALTEL.
- Realizar un análisis de las ordenes MML (lenguaje Hombre maquina), que se crean necesarias para realizar cualquier aplicación de gestión, de forma que se tengan un conjunto preciso de ordenes que serán enviadas a la central.
- Tener en cuenta las compatibilidades en las operaciones que se desean llevar a cabo en la central, en particular para este proyecto se debe tener claro los servicios que no son compatibles.
- Tener un nivel de seguridad que garantice la integridad de la central telefónica y la información que en ella se maneja.

SISTEMA DE ATENCIÓN AUTOMÁTICA DE SERVICIOS SAAS.

Sin duda algo sumamente importante resulta el hecho que ya existan aplicaciones que permiten una comunicación directa con la central telefónica; este es el caso del Sistema de Atención Automática de Servicios SAAS, desarrollado por la empresa TELEARMENIA, este proyecto fue llevado a cabo por el Ing. Germán Darío Guevara, actualmente desvinculado de esta empresa.

Este sistema permite extraer de la base de datos del SATplus la tabla que contiene las ordenes de solicitudes a servicios que están pendientes, una vez que obtiene este listado es capaz de conectarse a la central telefónica y ejecutar las ordenes que le permiten resolver las solicitudes a los servicios tanto básicos como suplementarios.

LA SEGURIDAD.

Es importante mejorar en algunos aspectos para las próximas aplicaciones de gestión que se desarrollen dentro del grupo de ingeniería telemática. En particular el aspecto de seguridad resulta muy importante, la confianza en un producto seguro, que preserve la integridad de la información y del sistema en general puede ser la diferencia entre tener o no tener éxito:

La seguridad es un aspecto a mejorar en la aplicación SGCST, el agente SNMP tiene limitantes en seguridad y estabilidad propias de la versión del protocolo de gestión SNMP v1. En las aplicaciones que presidan este proyecto y soportadas en la red Internet, es clave utilizar un protocolo mas avanzado que incluya aspectos mas destacados en lo que a seguridad se refiere, este es el caso de SNMP v3, que permitirá el desarrollo de sistemas de gestión relativamente más seguros y muy estables, además los nuevos agentes poseerán nuevas operaciones de gestión y configuraciones que lo hacen mucho más robusto.

Algo muy importante que mejoraría el nivel de seguridad para el sistema SGCST, deriva en el hecho que es una aplicación pensada para implantarse dentro de una red empresarial tipo Intranet, donde se tengan políticas rigurosas de seguridad.

No hay que desestimar el uso de dispositivos Hardware que permitan un sistema mucho mas seguro, la implementación de Firewall es una muy buena opción, para impedir que agentes extraños al sistema puedan invadir y causar daños, o que logren apoderarse total o parcialmente del mismo.

BIBLIOGRAFIA.

- Ejemplo de la MIB TOASTER en la Página de la Universidad Politécnica de Valencia.
<http://www.redes.upv.es/mer/practicass.htm>
- El MIB BROWSER en la Página de la Corporación MG-SOFT.
<http://www.mg-soft.com/mgMibBrowserPE.html>
- Teoría de SNMP en la Página de la Universidad de Alicante, Departamento de Física, Ingeniería de Sistemas y Teoría de la Señal.
<http://www.disc.ua.es/asignaturas/rc/trabajos/snmp/tema1.html>
- Página de Visual C++ en Microsoft.
<http://www.microsoft.com/latam/visualc/producto/caracteristicass/>
- Página de oficial de PHP
<http://www.php.net/manual/es/>
- Manual de PHP
Autores: Stig Sæther Bakken, Alexander Aulbach, Egon Schmid, Jim Winstead, Lars Torben Wilson, Rasmus Lerdorf, Zeev Suraski, Andrei Zmievski, y Jouni Ahto
Editado por Rafael Martínez, Publicado 15-04-2001
Copyright © 1999, 2000, 2001 por el Grupo de documentación de PHP
- Microsoft Visual C++ 6.0 – Manual del programador.
Beck Zaratian, Mc Graw Hill, 1999.
- Oracle8i para Windows NT - Edición de aprendizaje
Brobrowski, Osborne Mc Graw Hill, 1999.
- Oracle8 - Diseño de bases de datos con UML
Paul Dorsey - Joseph Hudicka, Osborne MC Graw Hill, 1999.
- SNMP SNMPv2 and RMON
William Stallings, Addison Wesley, 1996.
- Simple Network Management Protocol Agent Support (for Windows)
<http://msdn.microsoft.com/library/en-us/wcecomm/html/snmp.asp>
- RFC 1155 - "Structure and Identification of Management Information for TCP/IP-based Internets"
- RFC 1156 – "Management Information Base for Network Management of TCP/IP-based internets"

- RFC 1157 – “A Simple Network Management Protocol”.
- Manual SATplus – Open Systems.
- Sitios de información de TEXT-SCROLL de flash
http://www.kadazuro.com/tutoriales/txt_scroll/
http://www.htmlpoint.com/flash/flash_23.htm
- Documento SNMP y NMS (Network Management System, Sistema de Gestión de Redes)
<http://www.geocities.com/Heartland/Plains/3102/SNMP.htm>

