

UNIVERSIDAD MAYOR DE SAN ANDRES
FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA ELECTRÓNICA



PROYECTO DE GRADO

**SISTEMA DE MONITOREO Y REGISTRO EN BASE DE DATOS
DE EVENTOS DE RED PARA TERMINALES Y USUARIOS
USANDO EL PROTOCOLO ICMP Y NOTIFICACIÓN POR SMS**

POSTULANTE: RENE ALONZO CHOQUE SAIRE

TUTOR: ING. MARCELO RAMIREZ MOLINA

LA PAZ - BOLIVIA

2022



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE INGENIERIA**



LA FACULTAD DE INGENIERIA DE LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) Visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) Copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) Copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la cita o referencia correspondiente en apego a las normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADAS EN LA LEY DE DERECHOS DE AUTOR.

AGRADECIMIENTOS

A mi familia, por haberme dado la oportunidad de formarme en esta prestigiosa universidad y haber sido mi apoyo durante todo este tiempo.

De manera especial a mi tutor, por haberme guiado, no solo en la elaboración de este trabajo de titulación, sino a lo largo de mi carrera universitaria y haberme brindado el apoyo para desarrollarme profesionalmente y seguir cultivando mis valores.

A la Facultad de Ingeniería de la Universidad Mayor de San Andrés, por haberme brindado tantas oportunidades y enriquecerme en conocimiento.

ÍNDICE

RESUMEN (ABSTRACT)	6
1. INTRODUCCIÓN	6
1.1. ANTECEDENTES DEL PROYECTO	6
1.2. OBJETIVOS	9
1.2.1. OBJETIVO GENERAL	9
1.2.2. OBJETIVOS ESPECÍFICOS	9
1.3. MARCO DE REFERENCIA	10
1.3.1. PROBLEMÁTICA.....	10
1.3.2. JUSTIFICACIÓN DEL PROYECTO	10
1.3.2.1. JUSTIFICACIÓN TÉCNICA.....	10
1.3.2.2. JUSTIFICACIÓN ECONÓMICA	11
1.3.2.3. IMPACTO SOCIAL.....	12
1.3.3. FUNDAMENTO TEÓRICO DE LA TECNOLOGÍA UTILIZADA	12
1.3.3.1. MENSAJES DE ICMPV4 Y ICMPV6.....	12
CONFIRMACIÓN DE HOST	13
DESTINO O SERVICIO INACCESIBLE	13
TIEMPO SUPERADO	14
1.3.3.2. FUNDAMENTOS DE LAS CPU DE MÚLTIPLES NÚCLEOS Y EL CONCEPTO DE HYPER- THREADING	14

MÚLTIPLES NÚCLEOS	15
HYPER-THREADING	16
1.3.3.3. SOFTWARE CON PROCESOS EN SEGUNDO PLANO.....	18
1.3.3.4. MÓDULO GSM/GPRS SIM 800 L	20
CARACTERÍSTICAS:	20
DESCRIPCIÓN DE PINES.....	20
APLICACIONES.....	20
COMANDOS AT.....	21
2. DESARROLLO E INGENIERÍA DEL PROYECTO	23
2.1. HERRAMIENTAS DE REFERENCIA.....	23
2.2. BASE DE DATOS DEL PROYECTO	24
2.2.1. GESTOR DE BASE DE DATOS	24
2.2.2. MODELO DE LA BASE DE DATOS.....	24
2.3. PROGRAMA DE CAPTURA DE DATOS Y ACTUALIZACIÓN DE ALARMAS	29
2.3.1. LENGUAJE DE PROGRAMACIÓN E IDE.....	29
2.3.2. INTERFAZ DE USUARIO MÓDULO CAPTURA DE DATOS	30
2.3.3. FLUJOGRAMA DEL PROCESO DE CAPTURA.....	31
.....	31
.....	32
2.4. SISTEMA DE GESTIÓN HTTP Y MONITOREO TIPO USUARIO	33
2.4.1. INGRESO AL SISTEMA	33
2.4.2. MENÚ PRINCIPAL DEL SISTEMA TIPO USUARIO	33

2.4.3.	MENÚ PRINCIPAL DEL SISTEMA TIPO ADMINISTRADOR.....	34
2.4.4.	FLUJOGRAMA DE MONITOREO TIPO USUARIO	35
	35
2.4.5.	MÓDULO DE MONITOREO DE RESULTADOS	35
2.4.6.	MÓDULO DE GESTIÓN DE IPS	37
2.4.7.	MÓDULO GESTIÓN DE USUARIOS	38
2.5.	SISTEMA DE ENVÍO DE ALERTAS POR SMS.....	38
2.5.1.	FLUJOGRAMA DEL PROCESO DE ENVÍO DE ALERTA POR SMS	38
2.5.2.	SISTEMA PRINCIPAL DE NOTIFICACIÓN POR SMS.....	40
2.5.2.1.	INTERFAZ DE USUARIO.	40
2.5.2.2.	CIRCUITO DE COMUNICACIÓN	41
2.5.2.3.	FLUJOGRAMA COMANDOS AT AL MÓDULO SIM800L.....	42
2.5.3.	SISTEMA SECUNDARIO DE NOTIFICACIÓN POR SMS.	43
2.5.3.1.	LENGUAJE DE PROGRAMACIÓN E IDE.....	43
2.5.3.2.	INTERFAZ DE USUARIO.	43
3.	EVALUACIÓN.....	44
3.1.	ANÁLISIS DE RESULTADOS	44
3.2.	ANÁLISIS DE COSTOS	46
3.2.1.1.	COSTOS DE DESARROLLO E IMPLEMENTACIÓN.....	46
3.2.1.2.	COSTOS DE PRESTACIÓN DE SERVICIOS	46

3.2.1.3. COSTOS DE INSTALACIÓN Y OPERACIÓN.....	47
3.3. CONCLUSIONES Y RECOMENDACIONES	48
BIBLIOGRAFÍA.....	49
WEBGRAFÍA.....	49
ANEXOS.....	50
CÓDIGO FUENTE MÓDULO DE CAPTURA DE EVENTOS LENGUAJE DE PROGRAMACIÓN VISUAL BASIC.NET.....	50
CÓDIGO FUENTE MÓDULO DE GESTIÓN DE DATOS Y CONFIGURACIÓN DE SISTEMA LENGUAJE DE PROGRAMACIÓN HTML Y JAVASCRIPT CON JQUERY	67
CÓDIGO FUENTE MÓDULO DE NOTIFICACIÓN POR SMS EN LENGUAJE DE PROGRAMACIÓN JAVA PARA ANDROID.	71
CÓDIGO FUENTE INTERFAZ DE PROGRAMACIÓN DE APLICACIONES (API) PARA GESTIÓN DE ENVÍOS DE SMS EN LENGUAJE DE PROGRAMACIÓN PHP	79
CÓDIGO FUENTE SOFTWARE INTERNO DE RETRANSMISIÓN SERIAL PLACA ARDUINO	82
CÓDIGO FUENTE PARA ENVÍO DE SMS USANDO EL MÓDULO SIM800L	82

RESUMEN (ABSTRACT)

El propósito principal del presente proyecto es diseñar e implementar un sistema alternativo a los actuales sistemas de monitoreo de conectividad presentes en el mercado tecnológico de redes de datos. El aporte principal del presente proyecto es dar a los usuarios y administradores de red una herramienta fácil de manejar al momento de evaluar o diagnosticar la red de datos en momentos que se requiere mayor seguridad, tolerancia a fallas y escalabilidad dentro de una empresa o institución.

1. INTRODUCCIÓN

1.1. ANTECEDENTES DEL PROYECTO

Desde siempre los administradores de red en general buscan tener un control y registro de eventos relacionados con la conectividad de los distintos hosts¹ presentes a nivel LAN y WAN dentro de una institución o empresa. El presente proyecto implementa cuatro funcionalidades que permiten satisfacer estos requerimientos:

- Monitoreo de conectividad de terminales.
- Gestión y configuración de la información registrada.
- Control mediante autenticación de terminales.
- Notificación de Eventos por SMS²

El proyecto permite a cada administrador de red tener una visión general del estado de las conexiones a puntos determinados.

El método utilizado está basado en los conocimientos de redes de datos y programación obtenidos tanto para la gestión de bases de datos como para la captura de eventos dentro

¹ HOST. El término host es usado en informática para referirse a las computadoras conectadas a una red, que proveen y utilizan servicios de ella. De forma general un host es todo equipo informático que posee una dirección IP y que se encuentra interconectado con uno o más equipos. Un host o anfitrión es un ordenador que funciona como el punto de inicio y final de las transferencias de datos.

² SMS. El servicio de mensajes cortos o servicio de mensajes simples, más conocido como SMS (por las siglas del inglés Short Message Service), es un servicio disponible en los teléfonos móviles que permite el envío de mensajes cortos, conocidos como mensajes de texto.

de una red. De forma específica se usará los conocimientos obtenidos en programación orientada a objetos aplicada a networking³.

El proceso seguirá cuatro etapas:

Primera Etapa: Diseñar un programa de captura de eventos y almacenamiento en base de datos.

Segunda Etapa: Diseñar una interfaz http, para usuarios administradores, que tendrá como propósito la configuración del sistema y la lectura de datos.

Tercera Etapa: Diseñar una interfaz http para usuarios con restricciones de tipo cliente que deberán autenticarse en el sistema con el fin de hacer seguimiento a las terminales que requieren una seguridad alta y que solo son de uso permitido para determinados usuarios de la red

Cuarta Etapa: Diseñar un sistema de notificación por SMS para eventos de red los cuales estarán configurados por el o los administradores del sistema.

Actualmente existen varios sistemas que permiten el monitoreo de conexiones de red. A continuación, se presenta uno que fue escogido en función a su parecido funcional con el proyecto propuesto.

³ NETWORKING Término utilizado para referirse a las redes de telecomunicaciones en general y a las conexiones entre ellas.

- **Advanced Host Monitor de KS-Soft:** HostMonitor es una herramienta multiplataforma de administración de sistemas que monitorea continuamente la disponibilidad y el rendimiento de los servidores. En caso de errores de red, HostMonitor alertará al administrador de la red.

Test name	Recurrences	Status	Reply	Last test time	Alive %	Dead %	Last status	Last reply	Test
www.yahoo.com	22	Host is alive	1342 ms	29/04/2002 05:17:26 PM	67 %	0 %	Host is alive	1142 ms	URL
Root\Asia\Ping tests\									
192.52.233.10	110	Checking..		29/04/2002 05:17:20 PM	48 %	52 %	No answer		ping
192.52.233.11	22	Host is alive	301 ms	29/04/2002 05:17:25 PM	66 %	34 %	Host is alive	300 ms	ping
192.52.233.12	22	Host is alive	311 ms	29/04/2002 05:17:25 PM	66 %	34 %	Host is alive	320 ms	ping
192.52.233.13	22	Host is alive	331 ms	29/04/2002 05:17:25 PM	66 %	34 %	Host is alive	331 ms	ping
192.52.233.14	455	Checking..		29/04/2002 05:17:20 PM	0 %	100 %	No answer		ping
192.52.233.15	22	Host is alive	331 ms	29/04/2002 05:17:25 PM	66 %	34 %	Host is alive	341 ms	ping
192.52.233.25	22	Host is alive	411 ms	29/04/2002 05:17:25 PM	66 %	34 %	Host is alive	361 ms	ping
192.52.233.26	22	Host is alive	351 ms	29/04/2002 05:17:25 PM	66 %	34 %	Host is alive	351 ms	ping
192.52.233.27	455	Checking..		29/04/2002 05:17:20 PM	0 %	100 %	No answer		ping
192.52.233.29	455	Checking..		29/04/2002 05:17:20 PM	0 %	100 %	No answer		ping
192.52.233.35	22	Host is alive	411 ms	29/04/2002 05:17:25 PM	66 %	34 %	Host is alive	361 ms	ping
192.52.233.36	22	Host is alive	421 ms	29/04/2002 05:17:25 PM	66 %	34 %	Host is alive	361 ms	ping
192.52.233.37	22	Host is alive	411 ms	29/04/2002 05:17:25 PM	66 %	34 %	Host is alive	401 ms	ping
192.52.233.38	22	Host is alive	401 ms	29/04/2002 05:17:25 PM	66 %	34 %	Host is alive	371 ms	ping
192.52.233.39	22	Host is alive	411 ms	29/04/2002 05:17:25 PM	66 %	34 %	Host is alive	411 ms	ping
Root\Europe\									
Main router	373	Host is alive	0 ms	29/04/2002 05:17:25 PM	100 %	0 %	Host is alive	0 ms	ping
Root\Europe\CPU Usage\									
CPU (critical - 90%)	58	Ok	25 %	29/04/2002 05:17:26 PM	97 %	3 %	Ok	39 %	CPU
CPU \server1	368	Unknown		29/04/2002 05:17:26 PM	0 %	0 %	Unknown		CPU
CPU <local computer>	3	Checking..		21/05/2002 04:04:55 PM	93 %	7 %	Ok	43 %	CPU
Root\us\									
vdocs	72	Ok		29/04/2002 05:17:26 PM	81 %	19 %	Ok		File
\syslog.htm	234	Ok		29/04/2002 05:17:26 PM	9 %	91 %	Ok		File
192.52.233.48	1	Wait for Master		29/04/2002 05:17:19 PM	2 %	98 %	No answer		ping
DNS: 123.32.23.4	20	Wait for Master		21/05/2002 04:05:14 PM	0 %	91 %	No answer		DNS
IMAP: pop.cadvision.com	21	Wait for Master		21/05/2002 04:04:56 PM	0 %	87 %	No answer		IMA
LDAP: bigfoot.com	20	Wait for Master		21/05/2002 04:05:13 PM	0 %	82 %	No answer		LDA
ODBC: dBASE Files	37	Ok	www.yahoo.c...	29/04/2002 05:17:26 PM	100 %	0 %	Ok	www.yahoo...	ODE
POP3: pop.cadvision.com	22	Bad		29/04/2002 05:17:25 PM	0 %	91 %	Bad		PDF

Figura 1: Resultados Monitoreo Host Monitor

Similitudes

- Simplifica la administración remota mediante servicios web http generando un listado de hosts y mostrando el estado en que se encuentra la conectividad con cada uno.
- Mediante la aplicación ping usa el protocolo ICMP para el monitoreo de conectividad
- Entre las alertas que genera también incorpora el SMS para la notificación.

Diferencias

- No controla la autenticación de usuarios en terminales específicas.

1.2. OBJETIVOS

1.2.1. OBJETIVO GENERAL

Diseñar e implementar un sistema de monitoreo de conectividad de host de red y control de sesiones de usuarios, notificando a los administradores y usuarios eventos de la red vía SMS.

1.2.2. OBJETIVOS ESPECÍFICOS

Para alcanzar el objetivo será necesario llevar a cabo las tareas descritas a continuación que, individualmente, establecen objetivos secundarios específicos.

- Objetivo A: Diseñar un programa y una base de datos que permitan el registro de eventos de conectividad de la red. Para alcanzar este objetivo se diseñará el subsistema mediante el uso de VisualBasic.Net y una librería que usa el protocolo ICMP. Para la base de datos se usará el motor de base de datos MYSQL.
- Objetivo B: Diseñar un sistema de lectura de la información recopilada por el software de monitoreo y registro. Para el diseño e implementación se usará un Framework que usa HTML5, JavaScript, CSS3 y PHP
- Objetivo C: Diseñar un módulo de registro de sesiones de usuario usando el mismo Framework del punto anterior.
- Objetivo D: Diseñar e implementar dos módulos de alerta por SMS uno principal utilizando VisualBasic.Net dentro del sistema de registro de eventos conectado al módulo SIM800L y otro auxiliar usando el lenguaje de programación JAVA para Android instalado en un smartphone.

Al final del proyecto se presentará un prototipo funcional que permita mostrar toda la información antes mencionada.

1.3. MARCO DE REFERENCIA

1.3.1. PROBLEMÁTICA

Las redes deben admitir una amplia variedad de aplicaciones y servicios, como así también funcionar con diferentes tipos de infraestructuras físicas. El término arquitectura de red, en este contexto, se refiere a las tecnologías que admiten la infraestructura y a los servicios y protocolos programados que pueden trasladar los mensajes en toda esa infraestructura. Debido a que Internet evoluciona, al igual que las redes en general, sabemos que existen características básicas que una arquitectura de red necesita para cumplir con las expectativas de los usuarios: tolerancia a fallas, escalabilidad y seguridad.

- **Tolerancia a Fallas:** Una red tolerante a fallas es la que limita el impacto de una falla del software o hardware y puede recuperarse rápidamente cuando se produce dicha falla. La detección oportuna de estos eventos es importante para cumplir este objetivo.
- **Escalabilidad:** Una red escalable puede expandirse rápidamente para admitir nuevos usuarios y aplicaciones sin afectar el rendimiento del servicio enviado a los usuarios actuales.
- **Seguridad:** La seguridad de la infraestructura de una red consiste en el aseguramiento físico y lógico de los dispositivos que proporcionan conectividad y en evitar el acceso no autorizado al software administrativo que reside en ellos.

1.3.2. JUSTIFICACIÓN DEL PROYECTO

1.3.2.1. Justificación técnica

Tomando en cuenta la problemática que debe enfrentar una infraestructura de red que busca ser ampliable, segura y funcional, el proyecto justifica su implementación aportando información oportuna que beneficia y contribuye al cumplimiento de los requerimientos mencionados anteriormente.

- **Tolerancia a Fallas:** El sistema desarrollado permite la captura de eventos de latencia, conexión y desconexión de terminales de usuario y red de forma independiente, lo cual permite que el administrador de red se enfoque en el punto específico donde se produjo un problema y pueda solucionarlo en el menor tiempo posible.

- **Escalabilidad:** El sistema brinda una gestión remota de la información en tiempo real, lo cual da a los administradores de red una forma de ver el estado de la conexión de cada terminal nueva o instalada anteriormente. Aparte para un mejor orden y clasificación, en la medida que aumenten las terminales estas pueden clasificarse en grupos para hacer más fácil el seguimiento de la red en general.
- **Seguridad:** Para la seguridad física de los dispositivos el sistema incorpora una alerta en caso de desconexión no planificada (Traslado intempestivo, pérdida de energía, hurto, etc.). Para la seguridad lógica la configuración del sistema permite establecer una alerta en el caso de que una terminal se conecte en horarios fuera de lo habitual.

Por último, para la notificación a los administradores de red, el sistema implementa un subsistema de alertas por SMS, el cual hace uso de la red de un proveedor de telefonía móvil externo que es independiente de la red que se monitorea, esto asegura que la pérdida de la conexión de internet principal no afectará el funcionamiento del sistema.

1.3.2.2. Justificación económica

En base a las características requeridas para que una red en general pueda funcionar sin problemas es decir tolerancia a fallas, escalabilidad y seguridad, a continuación, se detalla algunos de los efectos económicos de no cumplirse estos requisitos.

Dichos efectos pueden concentrarse en la capa 7, capa de Aplicación del Modelo OSI.

Terminales finales o de usuario.

- Si hablamos de una terminal, independiente de la causa, el tiempo de inactividad de las aplicaciones de red que usa siempre conllevará altos costes en productividad que repercute directa o indirectamente en pérdidas económicas para la institución o empresa. Ponemos solo como ejemplos terminales de cajeros automáticos o cajeros convencionales en entidades financieras.

Nodos intermedios o de red.

- La desconexión de un nodo intermedio a nivel LAN implica la desconexión de todas las terminales finales dependientes de este nodo intermedio, lo cual conlleva costos por separado en productividad y pérdidas económicas mayores.
- La desconexión de un nodo intermedio a nivel WAN en el caso de un ISP no solamente genera un problema masivo de todos los usuarios o nodos conectados además genera multas por parte de las entidades reguladoras.

1.3.2.3. Impacto social

El segmento de la población afectada con la instalación del sistema dependerá de si se instala a nivel LAN (Red de área local) o a nivel WAN (Red de área extendida).

En lo referente a redes LAN se tiene dos opciones, alerta desde una terminal final o de usuario que afectaría solo a una terminal o desde un nodo intermedio o de red cuya afectación llegaría a todos los usuarios conectados a este nodo intermedio.

Referente a redes WAN en el caso de que se instale el sistema en un ISP (Proveedor de Servicio de Internet) que monitorea la conexión hacia sus Nodos de distribución (DSLAMs XDSL u OLTs GPON) una alerta de desconexión implica la pérdida de tráfico desde un NODO y la correspondiente pérdida de servicio de todos los usuarios dependientes de este NODO.

1.3.3. FUNDAMENTO TEÓRICO DE LA TECNOLOGÍA UTILIZADA

El proyecto propuesto busca satisfacer las anteriores necesidades mediante el uso de un sistema principal programado en VisualBasic.Net que verifica, mediante el protocolo ICMP, direccionamiento IPV4 e IPV6, la conectividad con un determinado punto de red y luego lo registra dentro de una base de datos hecha en MYSQL. Adicionalmente este sistema genera alarmas, programadas por el usuario, según los eventos, por ejemplo: pérdida de respuesta, aumento en el tiempo promedio o establecimiento de conexión.

En ese sentido la primera parte de la fundamentación teórica es referente al protocolo ICMP

1.3.3.1. Mensajes de ICMPv4 y ICMPv6⁴

El protocolo TCP/IP proporciona mensajes que se deben enviar en caso de que se produzcan determinados errores. Estos mensajes se envían mediante los servicios de ICMP. El objetivo de estos mensajes es proporcionar respuestas acerca de temas relacionados con el procesamiento de paquetes IP bajo determinadas condiciones. Los mensajes de ICMP no son obligatorios y, a menudo, no se permiten dentro de una red por razones de seguridad.

El protocolo ICMP está disponible tanto para IPv4 como para IPv6. El protocolo de mensajes para IPv4 es ICMPv4. ICMPv6 proporciona estos mismos servicios para IPv6, pero incluye

⁴ Principios básicos de enrutamiento y switching. CCNA1 V5. Material oficial de la Academia Cisco. Capítulo 8

funcionalidad adicional. En este proyecto, el término ICMP se utilizará para referirse tanto a ICMPv4 como a ICMPv6.

Existen muchos tipos de mensajes de ICMP y muchos motivos por los cuales se envían estos mensajes. Analizaremos algunos de los mensajes más comunes.

Los mensajes ICMP comunes a ICMPv4 y a ICMPv6 usados en este proyecto son los siguientes:

- Confirmación de host (Echo Request – Echo Reply)
- Destino o servicio inaccesible (Destination Unreachable)
- Tiempo superado (Time Exceeded)

Confirmación de host

Se puede utilizar un mensaje de eco de ICMP para determinar si un host está en funcionamiento a nivel de red. El host local envía una petición de eco de ICMP a un host. Si el host se encuentra disponible, el host de destino responde con una respuesta de eco. Este uso de los mensajes de eco de ICMP es la base de la utilidad ping.

Destino o servicio inaccesible

Cuando un host o gateway recibe un paquete que no puede entregar, puede utilizar un mensaje de destino inalcanzable de ICMP para notificar al origen que el destino o el servicio es inalcanzable. El mensaje incluye un código que indica el motivo por el cual no se pudo entregar el paquete.

CÓDIGOS DESTINO INALCANZABLE ICMPv4

0. Red inalcanzable
1. host inalcanzable
2. Protocolo inalcanzable
3. puertos inalcanzables
4. Se necesita fragmentación, pero no fragmente el conjunto de bits
5. La ruta de origen falló
6. Red de destino desconocida
7. Host de destino desconocido

8. Error aislado del host de origen (solo para uso militar)
9. La red de destino está prohibida administrativamente
10. El host de destino está prohibido administrativamente
11. No se puede acceder a la red para el tipo de servicio
12. No se puede acceder al host para el tipo de servicio
13. Comunicación prohibida administrativamente (el filtrado administrativo evita el reenvío de paquetes)
14. Infracción de precedencia de host (indica que la precedencia solicitada no está permitida para la combinación de host o red y puerto)
15. Corte de precedencia vigente (la precedencia del datagrama está por debajo del nivel establecido por los administradores de red).

CÓDIGOS DESTINO INALCANZABLE ICMPv6.

0. sin ruta al destino
1. comunicación con el destino prohibido administrativamente
2. más allá del alcance de la dirección de origen
3. direcciones inalcanzables
4. puertos inalcanzables
5. dirección de origen fallida política de entrada / salida
6. rechazar ruta a destino
7. Error en el encabezado de enrutamiento de origen

Tiempo superado

Los routers utilizan los mensajes de tiempo superado de ICMPv4 para indicar que un paquete no puede reenviarse debido a que el campo Tiempo de vida (TTL) del paquete se disminuyó a 0. En el caso de IPv6 se usa el campo Limite de Saltos (Hope Limit).

Si un router recibe un paquete y disminuye el campo TTL o Hope Limit en el paquete a cero, descarta el paquete y envía un mensaje de tiempo superado al host de origen.

1.3.3.2. Fundamentos de las CPU de múltiples núcleos y el concepto de Hyper-Threading

Un procesador de núcleo múltiple es un circuito integrado (IC) al cual se ha unido dos o más procesadores para mejorar el rendimiento, reducir el consumo de energía, y lograr el procesamiento simultáneo más eficiente de múltiples tareas o procesamiento en paralelo.

Un montaje de núcleo doble es algo comparable a tener múltiples procesadores separados, instalados en la misma computadora; pero debido a que los dos procesadores están de hecho conectados al mismo socket, la conexión entre ellos es más rápida.

La unidad central de procesamiento (CPU⁵) del equipo hace básicamente el trabajo de cálculo. Los CPU modernos ofrecen múltiples núcleos e Hyper-Threading. Algunos equipos incluso utilizan varias CPU.

La velocidad de reloj de CPU ya no es una referencia completa del rendimiento debido a que una CPU con Hyper-Threading tiene significativamente mejor desempeño que una CPU de un solo núcleo de la misma velocidad que no cuenta con esta última característica y aquellas con múltiples núcleos o CPUs pueden tener una ventaja aún mayor.

Múltiples Núcleos

Originalmente, las CPU tenían un solo núcleo. Eso significaba que la CPU física tenía una sola unidad central de procesamiento. Para aumentar el rendimiento, los fabricantes adicionaron “núcleos o unidades de tratamiento. Una CPU de doble núcleo tiene dos unidades centrales de procesamiento, por lo que aparece en el sistema operativo como dos CPUs. Una CPU con dos núcleos, por ejemplo, podría ejecutar dos procesos diferentes al mismo tiempo. Esto acelera su sistema, debido a que su computadora puede hacer varias cosas a la vez.

A diferencia de Hyper-Threading, una CPU de doble núcleo, literalmente, tiene dos unidades centrales de procesamiento en el chip de la CPU. Una CPU de cuatro núcleos tiene cuatro unidades de procesamiento central, una CPU octa-núcleo tiene ocho unidades de procesamiento central, y así sucesivamente.

Esto ayuda a mejorar drásticamente el rendimiento, manteniendo la unidad de CPU física compacta y en un solo módulo por lo que cabe en una sola toma o socket ⁶de CPU. Sin embargo, si bien todos los CPU están en un solo módulo, cada uno cuenta con su propia

⁵ CPU. La unidad central de procesamiento o unidad de procesamiento central (conocida por las siglas CPU, del inglés: central processing unit), es el hardware dentro de un ordenador u otros dispositivos programables, que interpreta las instrucciones de un programa informático mediante la realización de las operaciones básicas aritméticas, lógicas y de entrada/salida del sistema. El término, y su acrónimo, han estado en uso en la industria de la Informática por lo menos desde el principio de los años 1960

⁶ El Socket de CPU o Zócalo de CPU, es un tipo de zócalo electrónico (sistema electromecánico de soporte y conexión eléctrica) instalado en la placa base, que se usa para fijar y conectar el microprocesador, sin soldarlo lo cual permite ser extraído después. Por ello, se utiliza en equipos de arquitectura abierta, donde se busca que haya modularidad en la variedad de componentes, permitiendo el cambio de la tarjeta o el integrado.

energía, refrigeración, y otro hardware. Además, ya que todos están en el mismo chip hay menos latencia debido a que los núcleos pueden comunicarse con mayor rapidez.

En general, la relación de ejecutar múltiples subprocesos en un sistema de Microprocesador con N procesadores físicos, sobre ejecución de un subproceso único, se puede expresar como:

$$\text{RelativeResponse} = \frac{T_{\text{sequential}}}{T_{\text{parallel}}} = \left(1 - P + \frac{P}{N} + O \right)$$

donde P es la fracción de la carga de trabajo que se puede paralelizar, y O representa la sobrecarga de subprocesos múltiples y puede variar entre diferentes sistemas operativos. En este caso, la ganancia de rendimiento es la inversa de la respuesta relativa.

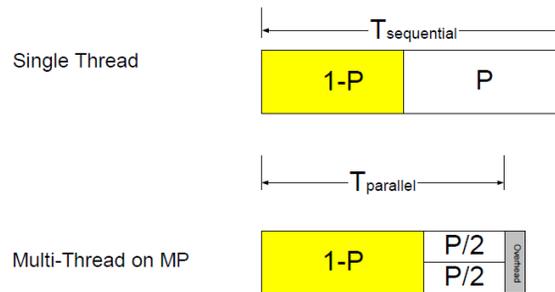


Figura 2: Ley de Amdahl para procesos de un solo hilo y procesos multihilo.

Hyper-Threading

Hyper-threading fue el primer intento de Intel para llevar la computación paralela a los PC de consumo. Debutó en las CPU de escritorio con el HT Pentium 4 en el año 2002. El Pentium 4 apareció con un solo núcleo de CPU, por lo que en realidad sólo se podía realizar una tarea a la vez, incluso si era capaz de cambiar entre tareas con la suficiente rapidez que parecía que trabajaba en multitarea. Hyper-threading trató de compensar eso.

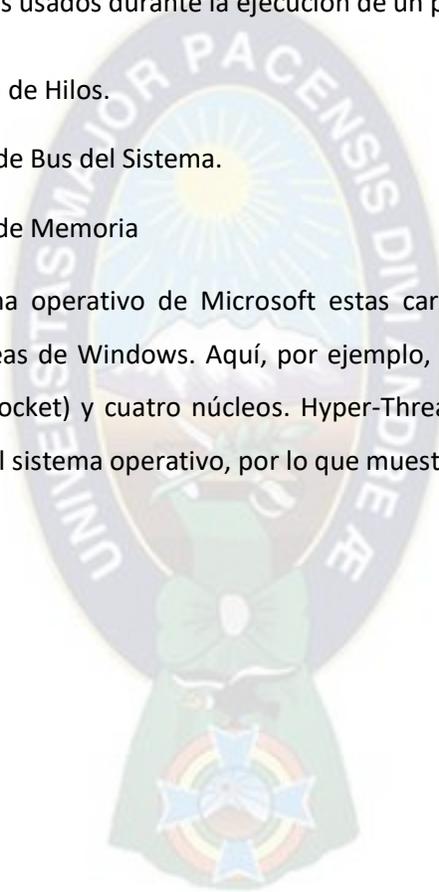
Una CPU física de un solo núcleo con Hyper-threading es reconocida como dos CPUs lógicas en el sistema operativo y utiliza su propia lógica para acelerar la ejecución del programa. En otras palabras, el Sistema Operativo ve 2 CPUs en lugar de solo una.

Hyper-threading permite que los dos núcleos de CPU lógicas puedan compartir recursos durante la ejecución física. Esto acelera la ejecución, si una CPU virtual se estancó y queda en espera, la otra CPU virtual puede tomar en préstamo sus recursos de ejecución. Hyper-threading puede ayudar a acelerar el sistema, pero no tanto como para compararse con el rendimiento de una CPU de 2 núcleos reales. Debido a ello actualmente el Hyper-threading es actualmente una característica adicional pero no un sustituto de la tecnología de múltiples núcleos.

En general los métodos usados durante la ejecución de un proceso Hyper-threading son:

- Sincronización de Hilos.
- Optimización de Bus del Sistema.
- Optimización de Memoria

En el caso del sistema operativo de Microsoft estas características se muestran en el Administrador de tareas de Windows. Aquí, por ejemplo, se puede ver que este sistema tiene una CPU real (socket) y cuatro núcleos. Hyper-Threading hace que cada núcleo se parece a dos CPU en el sistema operativo, por lo que muestra 8 procesadores lógicos.



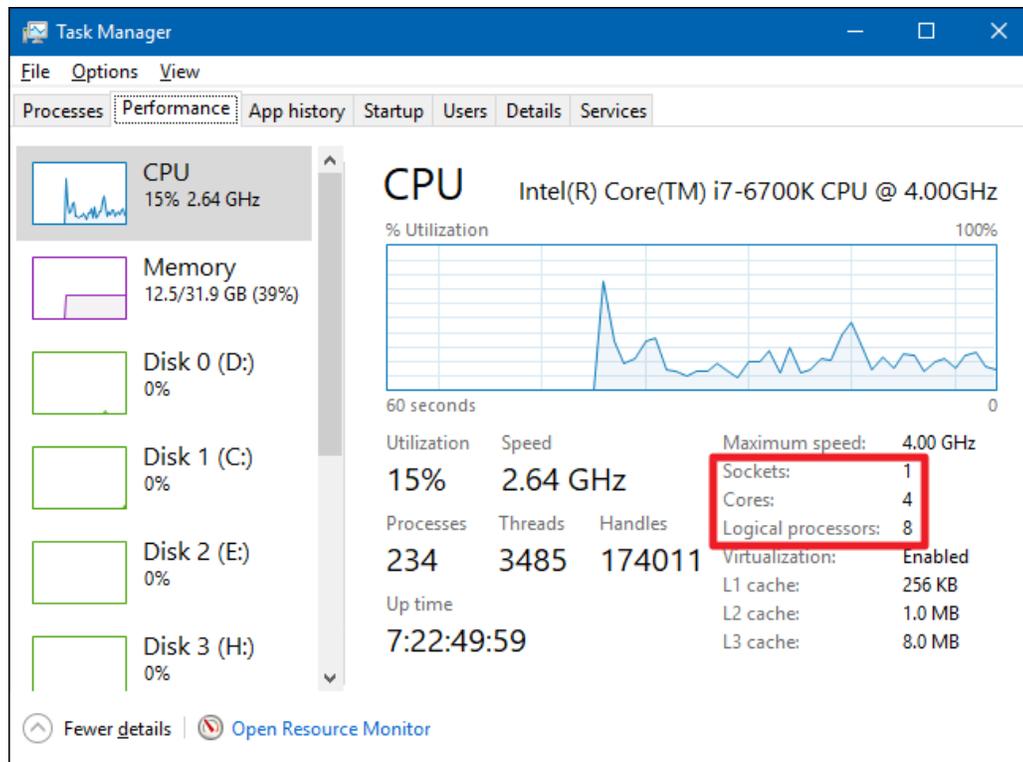


Figura 3: Captura de pantalla del Administrador de Windows que muestra el número de CPUs, Núcleos y Procesadores Lógicos

1.3.3.3. Software con procesos en Segundo Plano

Las clases ⁷en segundo plano permite ejecutar una operación en un hilo separado del programa principal y dedicado a un proceso específico. Operaciones que consumen mucho tiempo como descargas y transacciones de base de datos pueden hacer que su interfaz de usuario (UI⁸) parezca que ha dejado de responder mientras se están ejecutando. Cuando desea que una interfaz de usuario responda y se enfrente a largos retrasos asociados con tales operaciones, los procesos en segundo plano o “BackgroundWorker” proporcionan una solución conveniente.

⁷ En informática, una clase es una plantilla para la creación de objetos de datos según un modelo predefinido. Las clases se utilizan para representar entidades o conceptos. Cada clase es un modelo que define un conjunto de variables de estado, y métodos apropiados para operar con dichos datos es decir el comportamiento. Cada objeto creado a partir de la clase se denomina instancia de la clase.

⁸ La interfaz de usuario es el medio con que el usuario puede comunicarse con una máquina, equipo, computadora o dispositivo, y comprende todos los puntos de contacto entre el usuario y el equipo

Para ejecutar una operación que requiere mucho tiempo en segundo plano, se crea una instancia del BackgroundWorker el cual escucha los eventos que reportan el progreso de su operación y da una señal cuando la operación está terminada. Se puede crear el BackgroundWorker antes o durante la ejecución del programa.

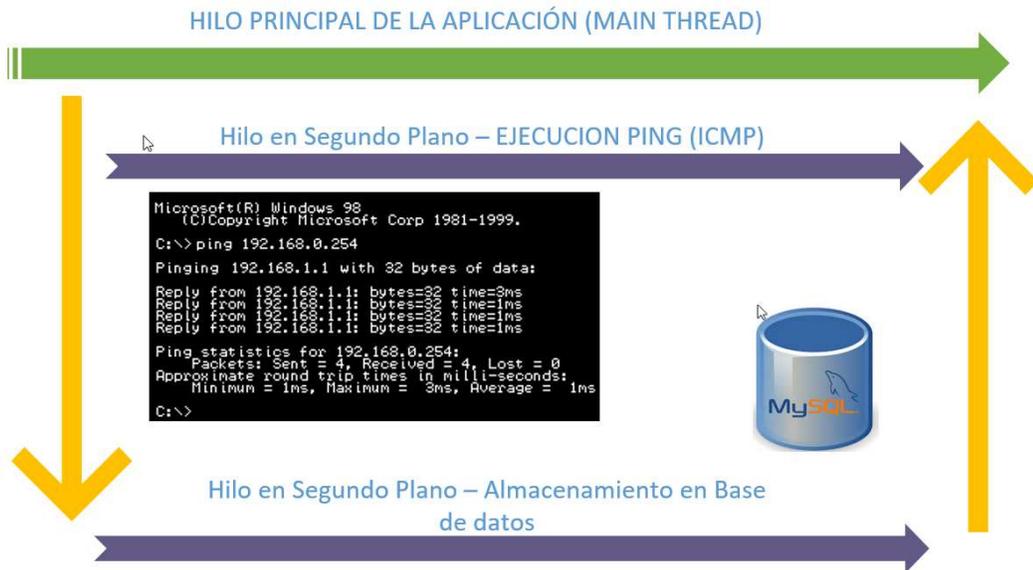


Figura 4: Diagrama de procesos en Segundo Plano

1.3.3.4. Módulo GSM/GPRS SIM 800 L

El módulo SIM800L es una solución GSM / GPRS de cuádruple banda completa con interfaz de fuente de alimentación y conexión a antena externa.

Características:

- Rango de voltaje de suministro 3.4 ~ 4.4V.
- Corriente de 1A o más (la corriente es muy importante)
- Cuenta con Bluetooth, FM y AT integrado (comandos AT)
- Banda cuádruple 850/900/1800 / 1900MHz
- Temperatura de funcionamiento: -40 ~ 85 grados centígrados.



Figura 5: Módulo SIM 800L

Descripción de PINES

- **VCC:** Alimentación Externa.
- **GND:** Conexión a tierra Externa.
- **VDD:** Entrada de voltaje de alimentación de Microcontrolador.
- **RST:** Pin de reinicio.
- **RXD:** Recepción – Comunicación Serial.
- **TXD:** Transmisión – Comunicación Serial.

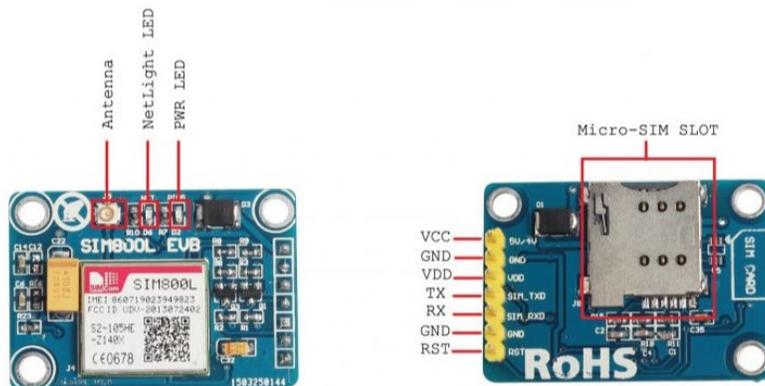


Figura 6: Disposición de PINES Módulo SIM800L

Aplicaciones

- El módulo SIM800L admite cuatro bandas 800/900/1800/1900 MHz y puede transmitir información de voz, SMS y datos con baja potencia.
- Debido a su tamaño reducido puede adaptarse a entornos compactos de diseño.

Comandos AT

Los comandos AT son comandos que se utilizan para controlar los módems en general donde AT significa Atención. Estos comandos se derivaron de los comandos de Hayes ⁹ que fueron utilizados por los módems inteligentes de Hayes. Cada conexión inalámbrica, así como los módems de acceso telefónico, requieren un comando AT para interactuar con una computadora.

Los siguientes 7 son los comandos AT más utilizados:

AT: Se utiliza para verificar la interacción entre la computadora y el módulo. Este comando generalmente se responde con un OK si el puerto y el módulo pueden conectarse correctamente, de lo contrario, vuelve con un código de resultado ERROR.

2. + CMGF: se utiliza para configurar el modo SMS. Al agregar 1 o 0 con el texto del comando o se puede seleccionar el modo PDU. Aquí el modo de texto es fácil de operar, aunque solo permite algunas funciones limitadas de SMS. Mientras que el modo PDU permite un acceso más detallado al servicio de SMS, aunque para usarlo se requieren conocimientos básicos de TDPU.

SINTAXIS: AT + CMGF = <modo>

3. + CMGS: se utiliza para enviar SMS a un número de teléfono en particular

SINTAXIS: AT + CMGS = número de serie del mensaje a enviar.

4. ATD: se usa para hacer llamadas a un número en particular

SINTAXIS: ATD; (Entrar)

5. ATA: se utiliza para responder las llamadas entrantes. Las llamadas se denotan con un mensaje "RING" que se duplica para cada timbre de la llamada. Después de que finalice la llamada, se muestra un mensaje que dice "NO HAY TRANSPORTADOR".

SINTAXIS: ATA (Enter)

6. + CMGW: Se usa para almacenar un mensaje en la SIM. Después de la ejecución del comando, el signo ">" aparece en la siguiente línea donde se puede ingresar el mensaje.

SINTAXIS: AT + CMGW = "Número de teléfono"> Mensaje a almacenar Ctrl + z

7. ATH: este comando se utiliza para desconectar a un usuario remoto con el módulo GSM.

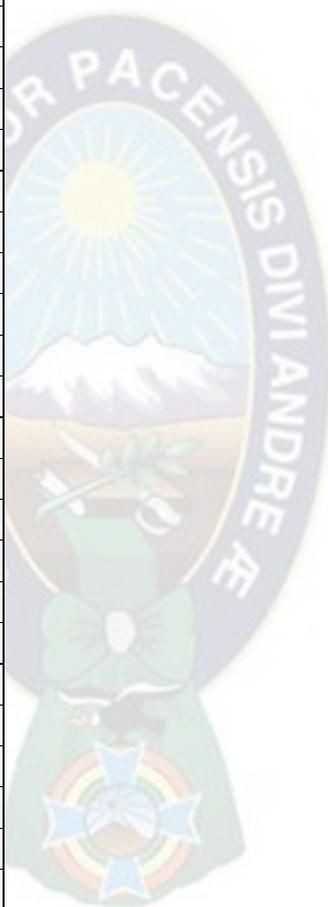
SINTAXIS: ATH (Enter).

⁹ El conjunto de comandos Hayes es un lenguaje de comandos específico desarrollado originalmente por Dennis Hayes para el módem Hayes Smartmodem 300 baudios en 1981.

8. AT+CSQ: este comando se utiliza para revisar la intensidad de la señal que recibe el dispositivo.

SINTAXIS: AT+CSQ (Enter).

Valor	RSSI ¹⁰ dBm	Condicion
2	-109	Marginal
3	-107	Marginal
4	-105	Marginal
5	-103	Marginal
6	-101	Marginal
7	-99	Marginal
8	-97	Marginal
9	-95	Marginal
10	-93	OK
11	-91	OK
12	-89	OK
13	-87	OK
14	-85	OK
15	-83	Good
16	-81	Good
17	-79	Good
18	-77	Good
19	-75	Good
20	-73	Excellent
21	-71	Excellent
22	-69	Excellent
23	-67	Excellent
24	-65	Excellent
25	-63	Excellent
26	-61	Excellent
27	-59	Excellent
28	-57	Excellent
29	-55	Excellent
30	-53	Excellent



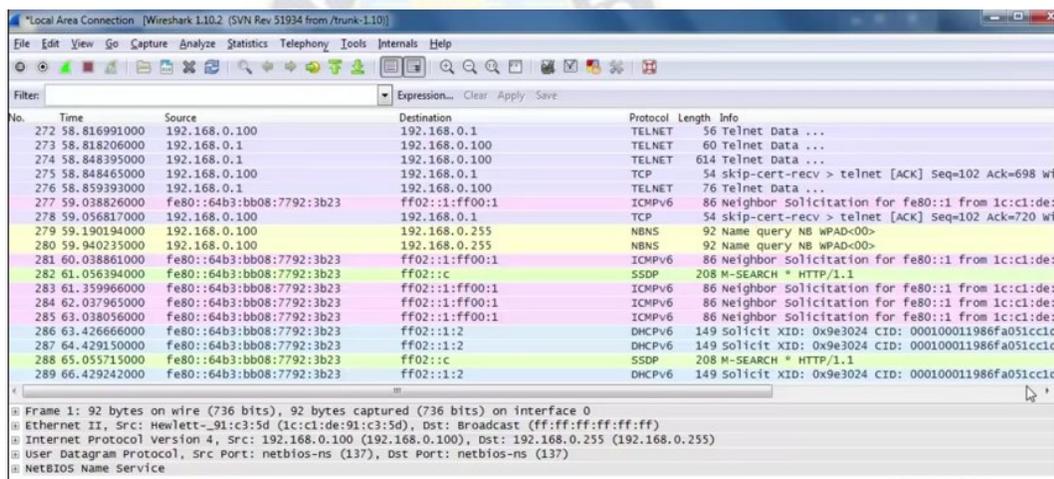
¹⁰ El indicador de fuerza de la señal recibida (RSSI por las siglas del inglés Received Signal Strength Indicator), es una escala de referencia (en relación a 1 mW) para medir el nivel de potencia de las señales recibidas por un dispositivo en las redes inalámbricas (típicamente WIFI o telefonía móvil). La escala tiene al valor 0 (cero) como centro; y representa 0 RSSI o 0 dBm. Aunque teóricamente puede darse el caso de medirse valores positivos, generalmente la escala se expresa dentro de valores negativos; cuanto más negativo, mayor pérdida de señal.

2. DESARROLLO E INGENIERÍA DEL PROYECTO

2.1. HERRAMIENTAS DE REFERENCIA

- **Wireshark patrocinada por la empresa Riverbed:** Wireshark es una herramienta multiplataforma de análisis de red. Funciona como **sniffer**¹¹, el mismo es un programa informático que registra la información que envían los periféricos, así como la actividad de red realizada en un determinado ordenador.

Wireshark no está dirigido a monitorear la conectividad solamente, sino todo el tráfico presente en la red. Por ejemplo, verificando la no presencia de una IP se puede inferir que ese punto de red no está activo.



No.	Time	Source	Destination	Protocol	Length	Info
272	58.816991000	192.168.0.100	192.168.0.1	TELNET	56	Telnet Data ...
273	58.818206000	192.168.0.1	192.168.0.100	TELNET	60	Telnet Data ...
274	58.848395000	192.168.0.1	192.168.0.100	TELNET	614	Telnet Data ...
275	58.848465000	192.168.0.100	192.168.0.1	TCP	54	skip-cert-recv > telnet [ACK] Seq=102 Ack=698 wi
276	58.859393000	192.168.0.1	192.168.0.100	TELNET	76	Telnet Data ...
277	59.038826000	fe80::64b3:bb08:7792:3b23	ff02::1:ff00:1	ICMPv6	86	Neighbor Solicitation for fe80::1 from 1c:c1:de:
278	59.056817000	192.168.0.100	192.168.0.1	TCP	54	skip-cert-recv > telnet [ACK] Seq=102 Ack=720 WI
279	59.190194000	192.168.0.100	192.168.0.255	NBNS	92	Name query NB WPAD<00>
280	59.940235000	192.168.0.100	192.168.0.255	NBNS	92	Name query NB WPAD<00>
281	60.038881000	fe80::64b3:bb08:7792:3b23	ff02::1:ff00:1	ICMPv6	86	Neighbor Solicitation for fe80::1 from 1c:c1:de:
282	61.056394000	fe80::64b3:bb08:7792:3b23	ff02::c	SSDP	208	M-SEARCH * HTTP/1.1
283	61.359966000	fe80::64b3:bb08:7792:3b23	ff02::1:ff00:1	ICMPv6	86	Neighbor Solicitation for fe80::1 from 1c:c1:de:
284	62.037965000	fe80::64b3:bb08:7792:3b23	ff02::1:ff00:1	ICMPv6	86	Neighbor Solicitation for fe80::1 from 1c:c1:de:
285	63.038056000	fe80::64b3:bb08:7792:3b23	ff02::1:ff00:1	ICMPv6	86	Neighbor Solicitation for fe80::1 from 1c:c1:de:
286	63.426666000	fe80::64b3:bb08:7792:3b23	ff02::1:2	DHCPv6	149	Solicit XID: 0x9e3024 CID: 000100011986fa051cc1c
287	64.429150000	fe80::64b3:bb08:7792:3b23	ff02::1:2	DHCPv6	149	Solicit XID: 0x9e3024 CID: 000100011986fa051cc1c
288	65.055715000	fe80::64b3:bb08:7792:3b23	ff02::c	SSDP	208	M-SEARCH * HTTP/1.1
289	66.429242000	fe80::64b3:bb08:7792:3b23	ff02::1:2	DHCPv6	149	Solicit XID: 0x9e3024 CID: 000100011986fa051cc1c

Figura 7: Ejemplo captura de información de tráfico de red en WIRESHARK

¹¹ SNIFFER. En informática, un Sniffer (anizador de protocolos) es un programa de captura de las tramas o paquetes de una red de computadoras. Los analizadores de paquetes tienen diversos usos, como monitorear redes para detectar y analizar fallos, o para realizar ingeniería inversa en protocolos de red. También es habitual su uso para fines maliciosos, como robar contraseñas, interceptar correos electrónicos, espiar conversaciones de chat, etc.

2.2. BASE DE DATOS DEL PROYECTO

2.2.1. Gestor de Base de Datos

El gestor de base de datos relacional utilizado es el **MySQL Server Community Edition** el cual viene con licencia GPL ¹²según lo cual se puede usar de manera libre. El software de administración y diseño utilizado es el **MySQL Workbench** ambos provistos por ORACLE.



Figura 8: Software de administración de Base de Datos

2.2.2. Modelo de la Base de Datos

1. **listhost**: Tabla en la que se almacenan los datos de los Host e IPS junto con una descripción de los mismos.
2. **eventos**: En esta tabla se almacena el tipo de evento (Conexión, desconexión, etc.) además de la hora del evento
3. **listgrupo**: Esta tabla es un listado que clasifica los hosts de la red.

¹² La Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License (o simplemente sus siglas del inglés GNU GPL) es la licencia de derecho de autor más ampliamente usada en el mundo del software libre y código abierto, y garantiza a los usuarios finales (personas, organizaciones, compañías) la libertad de usar, estudiar, compartir (copiar) y modificar el software

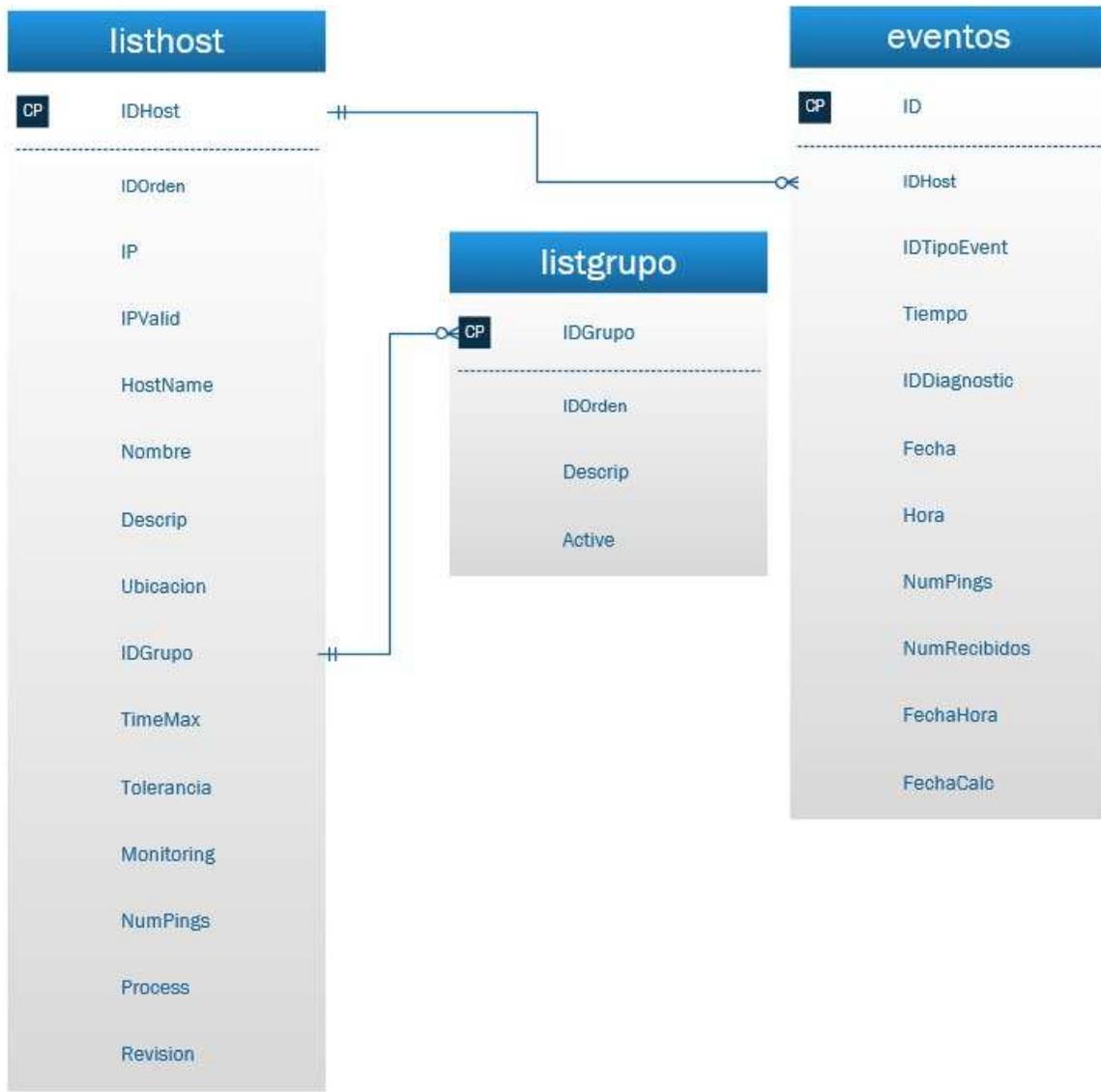


Figura 9: Esquema de la base de datos para las tablas listhost, listgrupo y eventos

4. Idtipocon: tabla secundaria que define si la conexión es de tipo cableada o inalámbrica
5. Idtipohost: Tabla secundaria que define si el host es en general portátil o fijo.

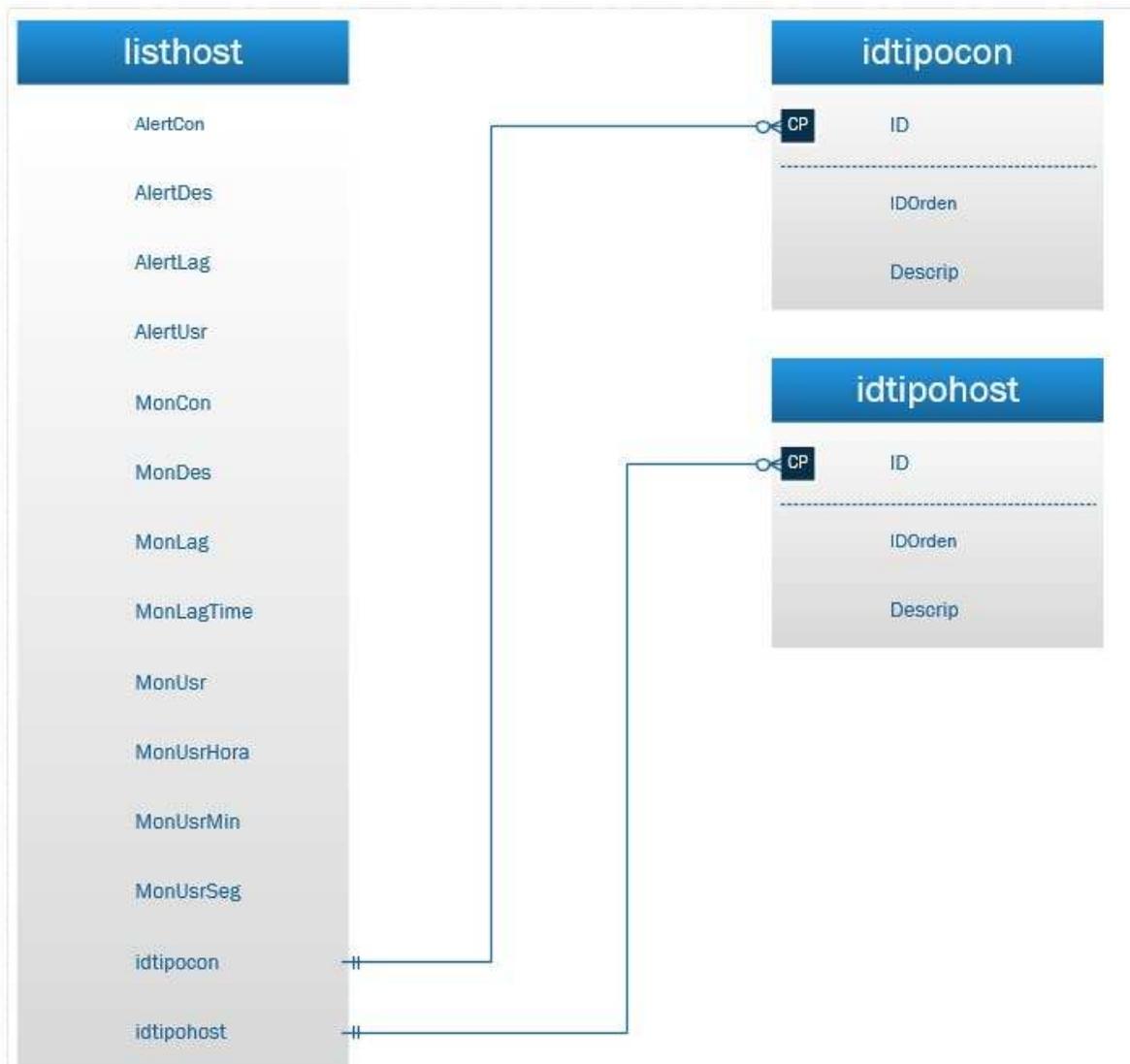


Figura 10: Esquema de la base de datos para las tablas listhost, idtipocon e idtipohost

6. **usr_list**: En esta tabla se almacena el listado de usuarios que pueden ingresar al sistema además de los permisos que tienen
7. **IDTitGrado**: Tabla secundaria usada como índice del grado de estudios del usuario
8. **IDCiExpedido**: Tabla secundaria usada para complementar la cedula de identidad del usuario.

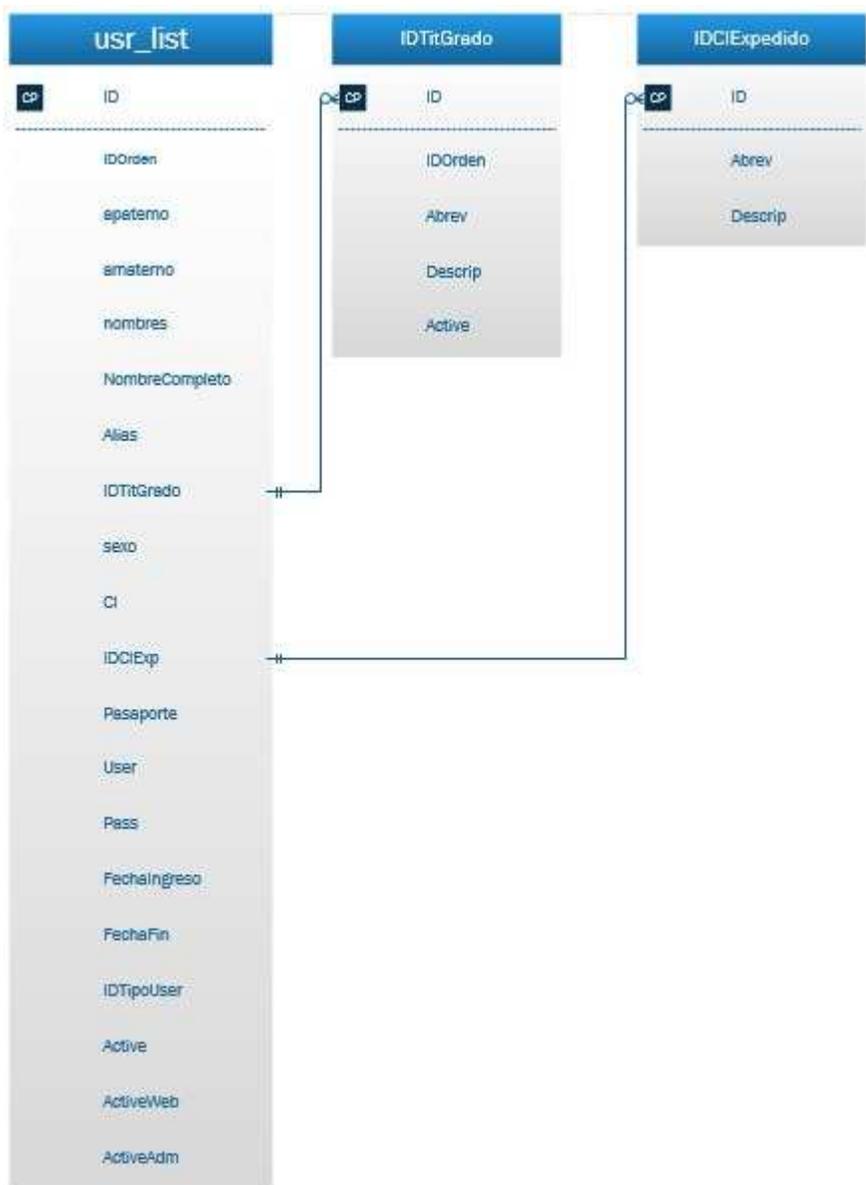


Figura 11: Esquema de la base de datos para las tablas *usr_list*, *IDTitGrado* e *IDCIEpedido*

9. **usr_area:** Indica el área en el que se encuentra el usuario (Ej Ventas, Sistemas, etc).
10. **usr_funcion:** Indica la función que cumple el usuario (Ej. Docente, Director, etc)



Figura 12: Esquema de base de datos para complementar la tabla *usr_list*, *usr_funcion* y *usr_area*



2.3. PROGRAMA DE CAPTURA DE DATOS Y ACTUALIZACIÓN DE ALARMAS

2.3.1. Lenguaje de Programación e IDE¹³

Para el software de Captura se utilizó el lenguaje Visual Basic.NET y el IDE de desarrollo proporcionado por Microsoft es decir el Visual Studio Community Edition. La razón principal de usar el mismo es que Microsoft proporciona una licencia de uso gratuito si el software es usado con fines académicos.



Figura 13: IDE de desarrollo

¹³ IDE: Es un entorno de desarrollo integrado o entorno de desarrollo interactivo, en inglés Integrated Development Environment (IDE), es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software. Normalmente, un IDE consiste de un editor de código fuente, herramientas de construcción automáticas y un depurador.

2.3.2. Interfaz de Usuario Módulo Captura de Datos

El Módulo de Captura de datos tiene una interfaz de usuario dividida en cuatro secciones

ICMP DATA BASE - MODULO CAPTURA DE DATOS - DESARROLLADO POR RENE ALONZO CHOQUE SAIRE

SISTEMA DE MONITOREO Y REGISTRO EN BASE DE DATOS DE EVENTOS DE RED PARA TERMINALES Y USUARIOS USANDO EL PROTOCOLO ICMP Y NOTIFICACION POR SMS

74.125.141.105 CONECTADO DESCONECTADO

00001 Respuesta desde 74.125.141.105 Tiempo 118 ms
00002 Respuesta desde 74.125.141.105 Tiempo 120 ms
00003 Respuesta desde 74.125.141.105 Tiempo 118 ms
00004 Respuesta desde 74.125.141.105 Tiempo 122 ms
00005 Respuesta desde 74.125.141.105 Tiempo 117 ms
00006 Respuesta desde 74.125.141.105 Tiempo 120 ms
00007 Respuesta desde 74.125.141.105 Tiempo 120 ms
00008 Respuesta desde 74.125.141.105 Tiempo 124 ms
00009 Respuesta desde 74.125.141.105 Tiempo 122 ms
00010 Respuesta desde 74.125.141.105 Tiempo 124 ms

8.8.8.8 CONECTADO DESCONECTADO

00027 Respuesta desde 8.8.8.8 Tiempo 108 ms
00028 Respuesta desde 8.8.8.8 Tiempo 110 ms
00029 Respuesta desde 8.8.8.8 Tiempo 106 ms
00030 Respuesta desde 8.8.8.8 Tiempo 105 ms
00031 Respuesta desde 8.8.8.8 Tiempo 111 ms
00032 Respuesta desde 8.8.8.8 Tiempo 107 ms
00033 Respuesta desde 8.8.8.8 Tiempo 108 ms
00034 Respuesta desde 8.8.8.8 Tiempo 108 ms
00035 Respuesta desde 8.8.8.8 Tiempo 108 ms
00036 Respuesta desde 8.8.8.8 Tiempo 111 ms

200.105.128.40 CONECTADO DESCONECTADO

00002 Respuesta desde 200.105.128.40 Tiempo 22 ms
00003 Respuesta desde 200.105.128.40 Tiempo 23 ms
00004 Respuesta desde 200.105.128.40 Tiempo 23 ms
00005 Respuesta desde 200.105.128.40 Tiempo 21 ms
00006 Respuesta desde 200.105.128.40 Tiempo 18 ms
00007 Respuesta desde 200.105.128.40 Tiempo 19 ms
00008 Respuesta desde 200.105.128.40 Tiempo 20 ms
00009 Respuesta desde 200.105.128.40 Tiempo 20 ms
00010 Respuesta desde 200.105.128.40 Tiempo 20 ms
00011 Respuesta desde 200.105.128.40 Tiempo 21 ms

Host 01 ID Host 17
Host 02 200.105.128.40
Host 03 ns1.accelerate.com

SECCION PANEL DE CONTROL

IPs Alarmas

GRUPO IPS Grp (1)(Administracion)

IDHost	IP	HostName	Monitoreo
1	74.125.141.105	www.google.com	1
2	127.0.0.1	localhost	0
6	8.8.8.8		1
7	200.87.129.195		0
17	200.105.128.40	ns1.accelerate.com	1

SECCION LISTADO DE IPS

IPs Alarmas

Tipo Alarma Todas Alarma Sonora

IP	HostName	AlertCon
74.125.141.105	www.google.com	1
8.8.8.8	8.8.8.8	1
200.105.128.40	ns1.accelerate.com	1

SECCION LISTADO DE ALARMAS

Figura 14: Interfaz de Usuario – Modulo de Captura de datos

1. **SECCIÓN PANEL DE CONTROL.** Esta sección lleva los botones para iniciar la revisión de una IP específica y el botón para iniciar y detener el proceso de captura de datos.
2. **SECCIÓN REVISIÓN DE MUESTREO.** Esta sección está compuesta por tres paneles que sirven para el seguimiento de la respuesta ICMP de una IP específica seleccionada del Listado de IPS
3. **SECCIÓN LISTADO DE IPS** Esta sección lleva el listado de IPS de la Base de Datos
4. **SECCIÓN LISTADO DE ALARMAS** Esta sección lleva el listado de Alarmas según la configuración de seguimiento de eventos guardada.

2.3.3. Flujo del Proceso de Captura

El Proceso Principal de captura de datos genera una lista de las IPS activas para monitoreo y luego crea una instancia del proceso de PING como un SUBPROCESO EN SEGUNDO PLANO para cada una de las IPS.

El Proceso Principal se repite periódicamente cada 10 segundos de manera que al momento de ingresar una nueva IP de monitoreo o deshabilitar alguna, la lista de IPS se actualice.

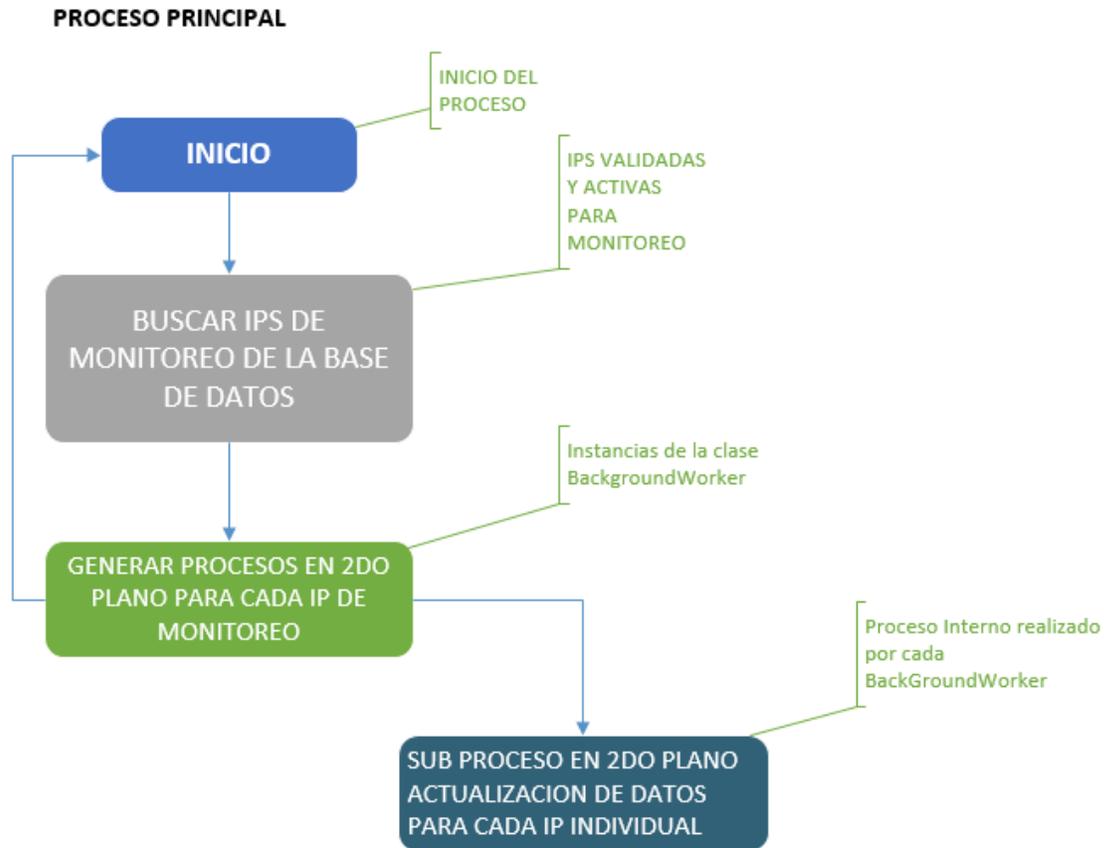


Figura 15: Proceso Principal

Una vez generados los subprocesos en segundo plano cada uno realiza un ping al IP registrado y almacena el resultado del mismo en la base de datos.

El almacenamiento se clasifica según el tipo de evento detectado

- Conexión Parcial o Completa
- Desconexión o Respuesta Nula

- Respuesta con Latencia¹⁴ alta
- Conexión de Usuario

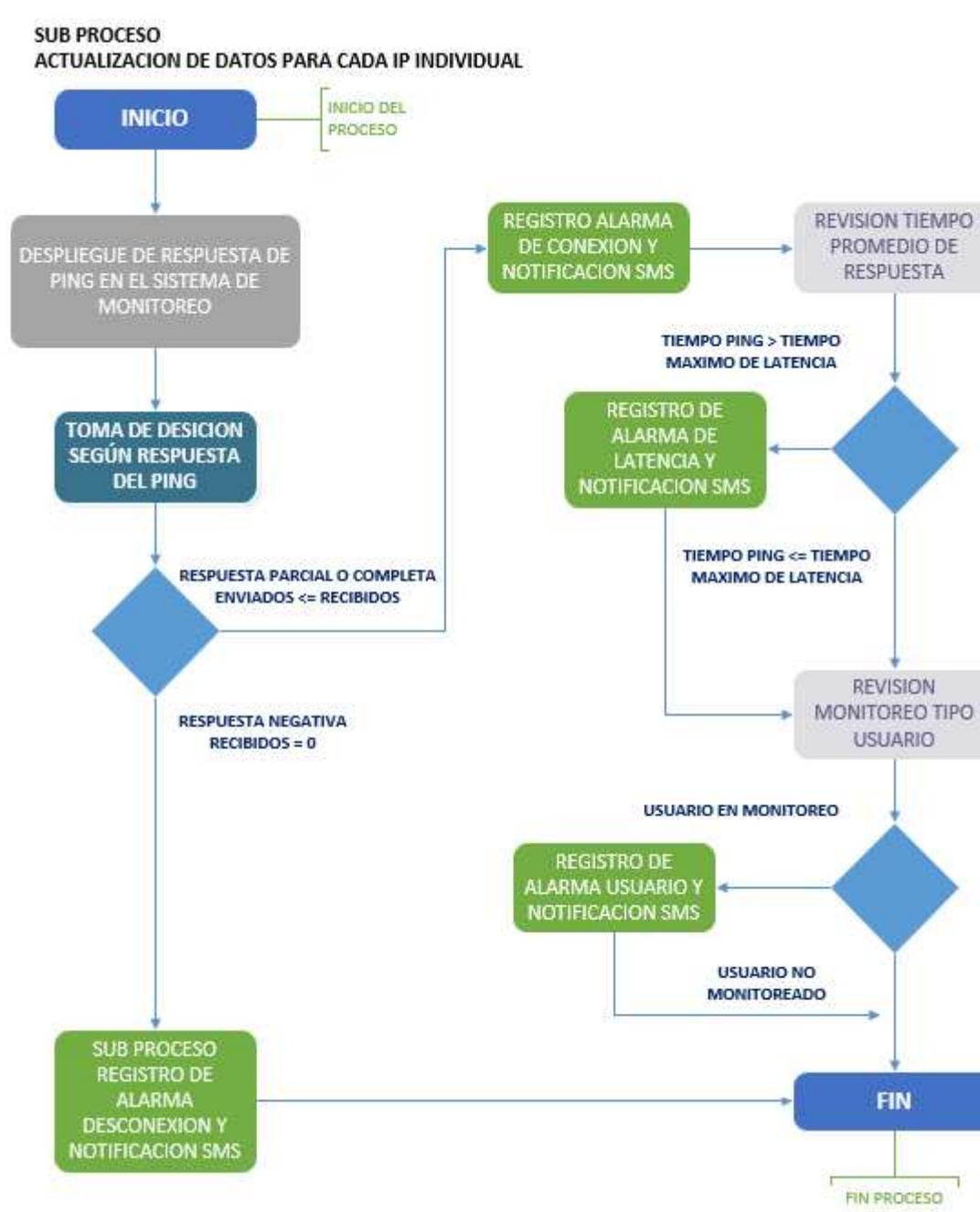


Figura 16: Proceso en segundo plano – Actualización de datos para cada IP

¹⁴ En redes informáticas de datos se denomina latencia a la suma de retardos temporales dentro de una red. Un retardo es producido por la demora en la propagación y transmisión de paquetes dentro de la red.

2.4. Sistema de Gestión HTTP y Monitoreo Tipo Usuario

2.4.1. Ingreso al sistema

Para ingresar al sistema de Gestión HTTP ¹⁵ se requiere definir la IP del servidor o el dominio asignado y luego de colocarlos en la barra de direcciones. Una vez colocados los datos de autenticación se visualizará dos tipos de sesiones la primera tipo USUARIO y la segunda tipo ADMINISTRADOR cada una con un menú de herramientas distinto.

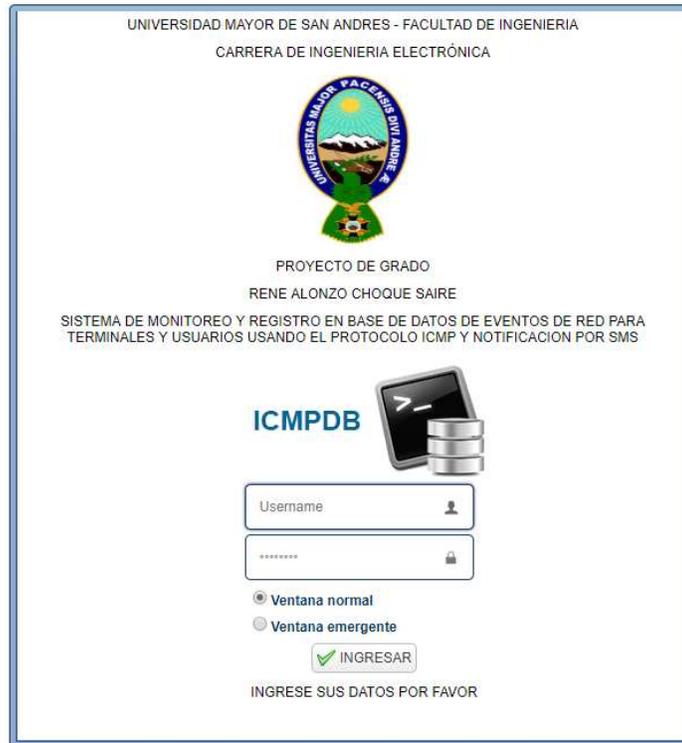


Figura 17: Ventana de Ingreso al Sistema de Gestión HTTP

2.4.2. Menú Principal del SISTEMA TIPO USUARIO

Para el módulo tipo USUARIO la pantalla principal de sistema permite solo el cambio de PASSWORD dentro del menú de CONFIGURACION.

- Configuración: Cambio de Password

¹⁵ Hypertext Transfer Protocol o HTTP (en español protocolo de transferencia de hipertexto) es el protocolo de comunicación que permite las transferencias de información en la World Wide Web. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. El cliente (se le suele llamar "usuario", en inglés user) realiza una petición enviando un mensaje, con cierto formato al servidor. El servidor (se le suele llamar un servidor web) le envía un mensaje de respuesta.



Figura 18: Menú Tipo Usuario

2.4.3. Menú Principal del SISTEMA TIPO ADMINISTRADOR

Una vez aceptados los datos de Usuario y Password vemos en la parte superior el Menú principal en el que se encuentran los distintos módulos de lectura y configuración de todo el sistema.

- Monitoreo
- Gestión de IPS
- Gestión de Usuarios
- Configuración: Cambio de Password

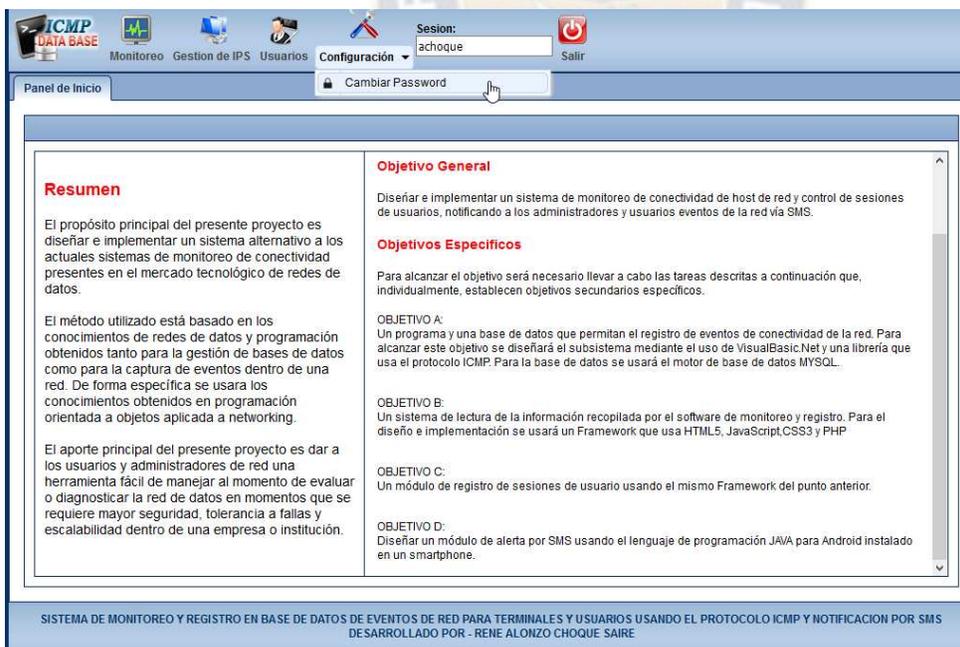


Figura 19: Menú Tipo Administrador

2.4.4. Flujograma de Monitoreo tipo usuario

Una vez el usuario autenticado en sistema se verifica el tiempo que tardó en el ingreso comparándolo con el tiempo configurado como máximo para espera

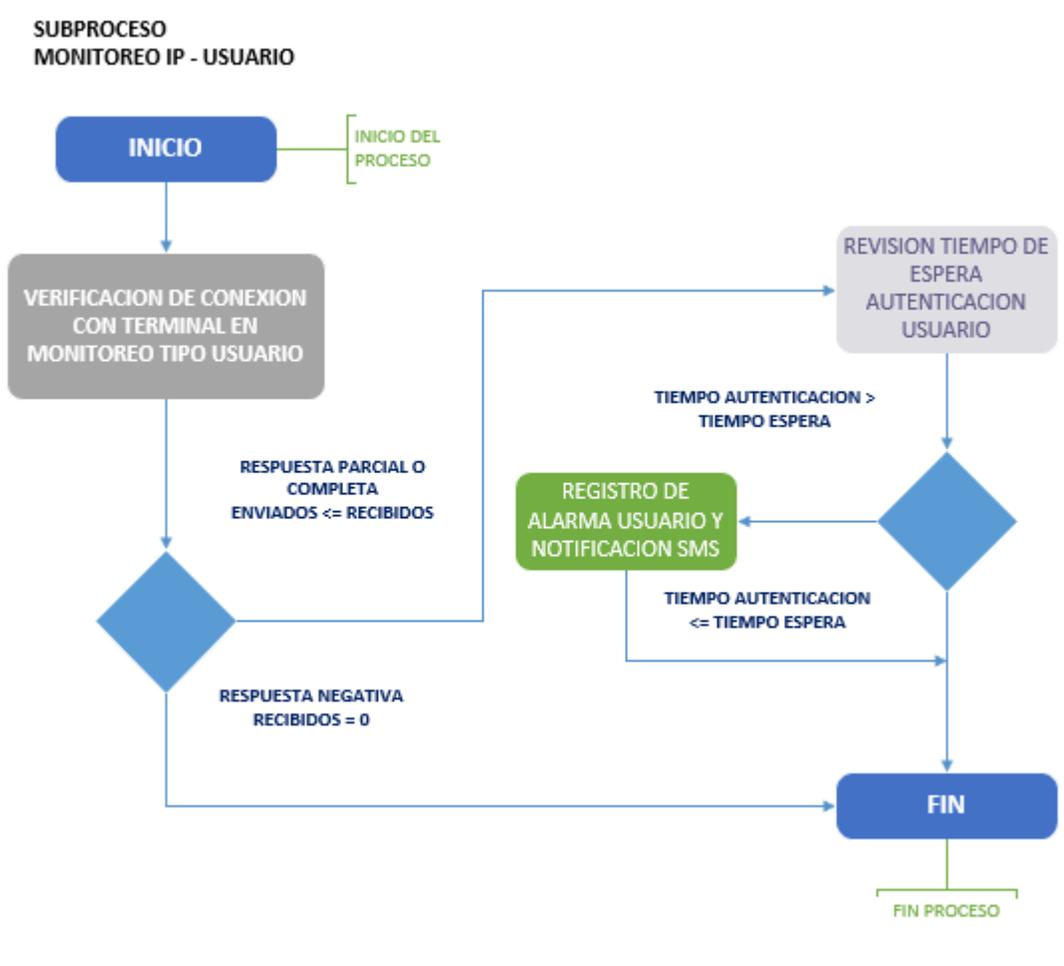


Figura 20: Flujograma Monitoreo IP - Usuario

2.4.5. Módulo de Monitoreo de Resultados

En este módulo se puede visualizar los resultados del monitoreo realizado para nuestro ejemplo para el IP DNS 8.8.8.8

IPS				IP MONITOREO >> 8.8.8.8								
IP	Nombre	Mor		Evento	Tiempo Prom.	Rango De Tiempo	FECHA	HORA	Env.	Rec.	Perd.	
1	200.105.131.57		no	1	Conexion Completa	129	< = 1 seg	2017-07-13	14:27:36	5	5	0
2	127.0.0.1		no	2	Con. Completa (Ini)	88	< = 100 ms	2017-07-13	14:27:34	5	5	0
3	200.105.128.40		no	3	Conexion Completa	161	< = 1 seg	2017-06-12	11:55:22	5	5	0
4	173.194.37.87		no	4	Conexion Parcial	150	< = 1 seg	2017-06-12	11:55:20	5	4	1
5	8.8.8.8	DNS Google	si	5	Conexion Parcial	266	< = 1 seg	2017-06-12	11:55:18	5	4	1
6	200.87.129.195		no	6	Conexion Completa	140	< = 1 seg	2017-06-12	11:55:06	5	5	0
7	200.105.128.46	Acelerate	si	7	Conexion Parcial	220	< = 1 seg	2017-06-12	11:55:04	5	4	1
8	192.168.0.4	PC Prueba	si	8	Conexion Parcial	156	< = 1 seg	2017-06-12	11:55:02	5	4	1
9	192.168.85.196	Android	si	9	Conexion Completa	141	< = 1 seg	2017-06-12	11:54:49	5	5	0
				10	Conexion Parcial	143	< = 1 seg	2017-06-12	11:54:47	5	4	1
				11	Conexion Completa	128	< = 1 seg	2017-06-12	11:54:41	5	5	0
				12	Conexion Completa	208	< = 1 seg	2017-06-12	11:54:34	5	5	0
				13	Conexion Completa	123	< = 1 seg	2017-06-12	11:54:32	5	5	0
				14	Conexion Completa	177	< = 1 seg	2017-06-12	11:54:27	5	5	0
				15	Conexion Parcial	122	< = 1 seg	2017-06-12	11:54:25	5	4	1
				16	Conexion Completa	166	< = 1 seg	2017-06-12	11:54:13	5	5	0

Figura 21: Monitoreo de Resultados

La interfaz visual indica:

- El Evento que puede ser conexión completa, parcial, desconexión total y latencia.
- El Tiempo promedio de respuesta para un muestro de 5 paquetes enviados.
- El Intervalo de respuesta, es decir menor a 1 s, menor a 100ms o menor a 10 ms.
- Fecha y Hora del muestreo
- Paquetes enviados, Recibidos y Perdidos

2.4.6. Módulo de Gestión de IPS

El módulo actual tiene 3 funciones principales separadas en 3 columnas.

- La creación y asignación de GRUPOS de IPS
- La creación y edición de IPS para su posterior monitoreo
- La asignación de Usuarios de control para cada IP y la habilitación de los mismos para envío de SMS.

The screenshot shows the 'Gestion de IPS' interface with three main panels:

- GRUPOS DE ASIGNACION:** A table with columns ID, Nombre del Grupo, and Hab. It lists groups like 'Indefinido', 'Administracion', 'Almacen', and 'ROUTERS y ACCESS POINTS'.
- LISTA DE IPS:** A table with columns Mon. ICMP, IP Asignado, and Nombre del Host. It lists IP addresses and hostnames like '74.125.141.105' and 'www.google.com'.
- Usuarios Para Monitoreo:** A table with columns Nombre, Monitoreo, SMS, and Funcion. It lists a user 'CHOQUE SAIRE, RENE ALONZO' with monitoring and SMS capabilities.

Figura 22: Gestión de IPS

Para el caso de Cada IP se tiene una ventana de edición para configurar sus parámetros de muestreo

The 'Editar IP HOST' window contains the following configuration fields:

- Grupo de IPs:** Grp (3)(ROUTERS y ACCESS POINTS)
- IP:** 192.168.1.1
- Host Name o Dominio:** (Empty)
- Tiempo Maximo de PING (ms):** 1000
- Tolerancia(ms):** 500
- Nombre:** Router ENTEL - TP Link
- Host:** (Empty)
- Ubicacion:** Bloque Secundaria - Oficina Disciplina
- IP Valida:** Estado: IP Valido
- Monitoreo ICMP (Ping):**
- Tiempo Actualizacion (ms):** 1000
- Monitoreo Conectividad:** Conexion Desconexion
- Monitoreo Latencia:** Max Latencia (ms): 1000
- Monitoreo Usuario:** Tiempo Espera - Hrs: 0 Min: 10 Seg: 0

Buttons: Guardar (Save), Cancelar (Cancel)

Figura 23: Gestión de IPS – Ventana de configuración de Parámetros de muestreo

2.4.7. Módulo Gestión de Usuarios

En este módulo se registran todos los parámetros referentes a cada usuario individual

The screenshot displays the 'Gestión de Usuarios' interface. On the left, a table lists users, with 'CHOQUE SAIRE, RENE ALONZO' selected. The main area shows the user's profile details under the 'Info. Básica' tab. The user's name is 'CHOQUE SAIRE, RENE ALONZO', and their function is 'Encargado/a'. The profile includes fields for 'Ap. Paterno' (CHOQUE), 'Ap. Materno' (SAIRE), 'Nombres' (RENE ALONZO), 'Alias' (Alonzo), 'Ci' (4376772), 'LP' (LP), and 'Pasaporte'. The 'Función' is 'Encargado/a', and the user is marked as 'Docente' and 'Administrativo'. The 'Área' is 'Indefinido', and the 'Login' is 'achoque' with a password of '123456'. The 'Comunicación' tab shows contact information: 'Celular ALERTA SMS' (78761182), 'Correo Electronico Cooperativo', and 'Correo Electronico Personal'. The interface includes a search bar, a table with columns 'Nombre' and 'Funcion', and a bottom navigation bar with buttons for 'Actualizar', 'Eliminar', 'Limpiar', and 'Adicionar Nuevo'.

Nombre Usuario	Funcion
1 CHOQUE SAIRE, RENE ALONZO	Encargado/a

Selecionado >> CHOQUE SAIRE, RENE ALONZO

Info. Básica

1 Ap. Paterno: CHOQUE Tit/Grado:
Ap. Materno: SAIRE Indef.
Nombres: RENE ALONZO Genero: Masculino
Alias: Alonzo
Ci: 4376772 LP Pasaporte:
Función: Encargado/a Docente Administrativo
Área: Indefinido
Login: achoque Password: 123456

Comunicación

Celular ALERTA SMS: 78761182
Correo Electronico Cooperativo:
Correo Electronico Personal:

Mostrando 1 a 1 de 1 items

Actualizar Elimpiar Adicionar Nuevo

Figura 24: Gestión de Usuarios

2.5. SISTEMA DE ENVÍO DE ALERTAS POR SMS

2.5.1. Flujograma del Proceso de envío de alerta por SMS

El Proceso Principal de captura de datos genera una lista alarmas activas según el monitoreo y las mismas son leídas por el sistema de Gestión HTTP y por el sistema de notificación de alertas por SMS.

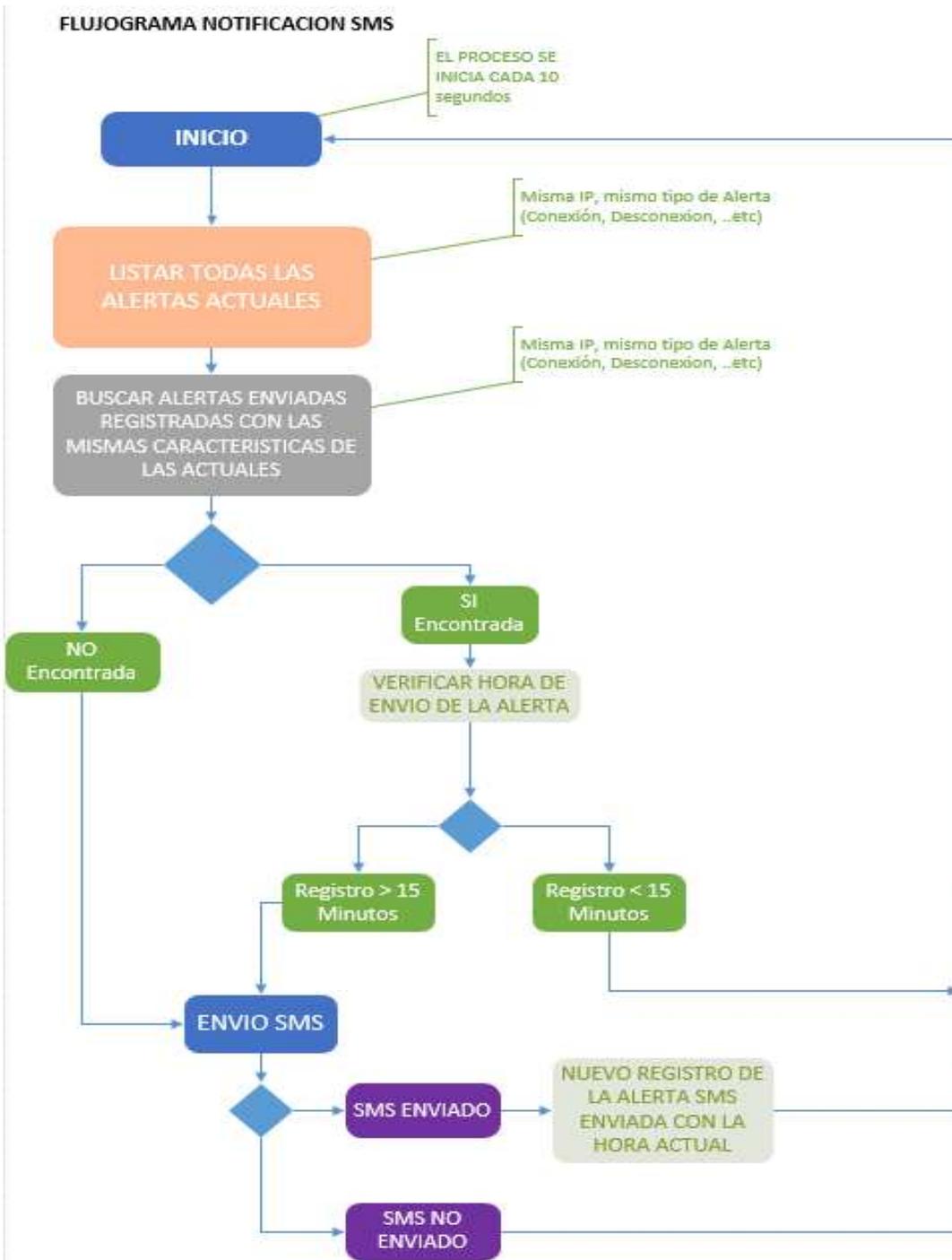


Figura 25: Flujograma Notificación SMS

El Proceso de lectura y decisión de envió se repite periódicamente cada 5 segundos de manera que al momento de que se active una nueva alarma esta sea reportada por el SMARTPHONE. Si una alarma fue enviada y el problema persiste después de 15 minutos se volverá a enviar un SMS

2.5.2. Sistema principal de notificación por SMS.

El sistema principal de notificación por SMS usa el módulo GSM SIM800L el cual se conecta al servidor de captura por USB ¹⁶ usando un protocolo SERIAL de comunicaciones y comandos AT para el envío de instrucciones al módulo SIM800L.

2.5.2.1. Interfaz de Usuario.

Dentro del módulo de captura de eventos se tiene una tercera ventana en la cual se configura la conexión SERIAL USB hacia el módulo SIM 800L definiendo la frecuencia de conexión a 9600 bps.

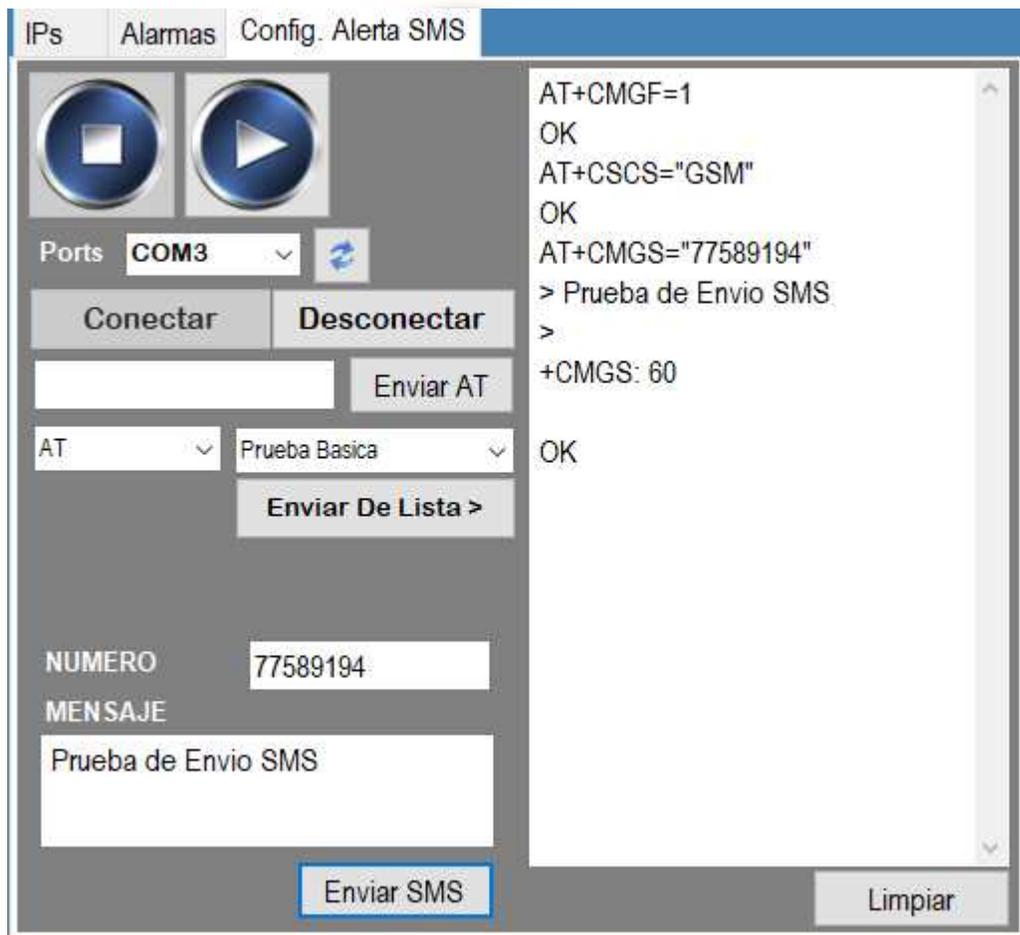


Figura 26: Interfaz de Usuario VB NET - Modulo SIM 800L

¹⁶ El Bus Universal en Serie (BUS) (en inglés: Universal Serial Bus), más conocido por la sigla USB, es un bus de comunicaciones que sigue un estándar que define los cables, conectores y protocolos usados en un bus para conectar, comunicar y proveer de alimentación eléctrica entre computadoras, periféricos y dispositivos electrónicos.

2.5.2.2. Circuito de comunicación

El servidor de captura de eventos se conecta al módulo SERIAL USB de la placa de desarrollo ARDUINO la cual retransmite la información serial al MODULO SIM800L (Figura 27).

Los Pines 8 (Recepción) y 9 (Transmisión) en la placa ARDUINO son los encargados de enviar y recibir la información del módulo SIM800L.

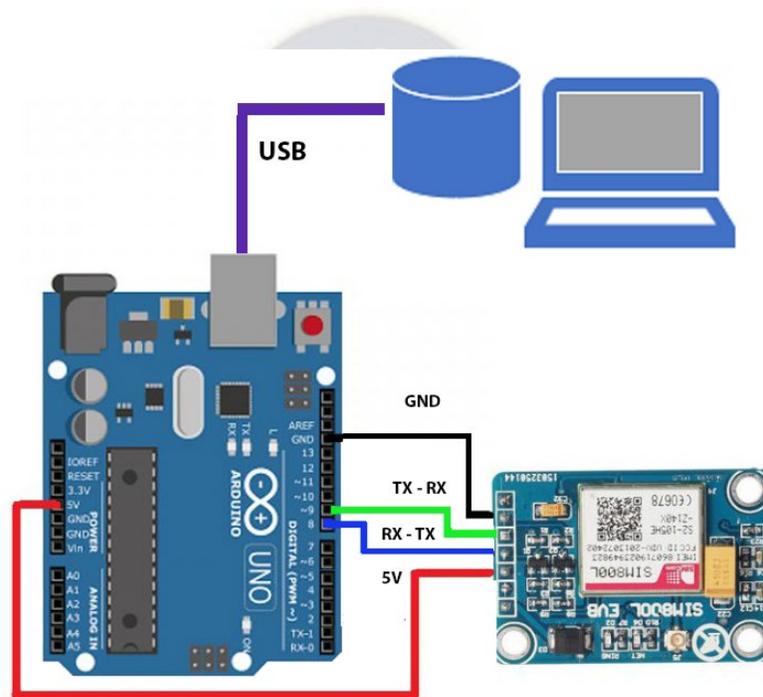


Figura 27: Esquema del circuito de comunicación PC – SIM800L



2.5.2.3. Flujograma comandos AT al Módulo SIM800L

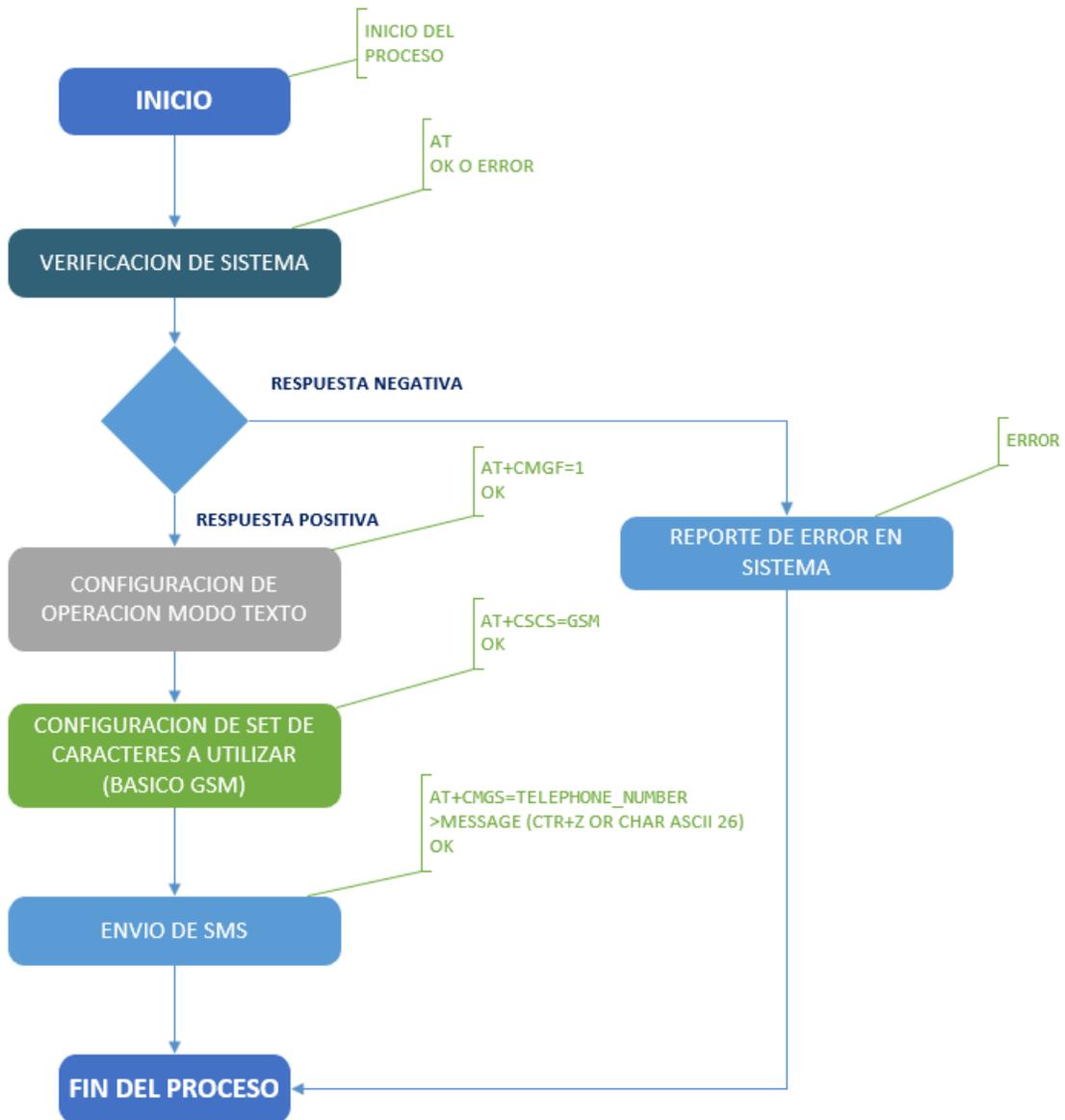


Figura 28: Flujograma envío de comandos AT Modulo SIM800L

Previo inicio del proceso de envío de SMS se debe ejecutar el comando **AT+CSQ** el cual nos indica el grado de intensidad de la señal que se está recibiendo y según la recomendación del fabricante el valor debe estar entre -53 dbm (Señal Excelente) y -109 dbm (Valor Marginal). No se ejecuta dentro del proceso cíclico dado que solo se realizará una vez al momento de la instalación del sistema de notificación.

2.5.3. Sistema secundario de notificación por SMS.

Tomando en cuenta el posible escenario según el cual el servidor de captura de eventos se encuentre en un ambiente con baja señal de telefonía celular, se desarrolló un sistema secundario que se conecta al servidor por TCP/IP y se actualiza usando una API http.

2.5.3.1. Lenguaje de Programación e IDE

Para el módulo secundario de notificación de alertas por SMS se desarrolló un programa de lectura en lenguaje JAVA para ANDROID con el IDE de Android Studio el cual es de uso gratuito.

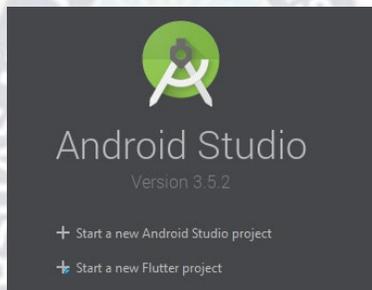


Figura 29: IDE de Desarrollo ANDROID STUDIO

2.5.3.2. Interfaz de Usuario.

Dentro del SMARTPHONE se tiene instalado el Software antes mencionado el cual permite la configuración de la URL de a API¹⁷ de lectura de datos de las alertas y mensajes además de una manera de controlar el inicio y fin de la tarea en segundo plano que realiza la actualización de información que según se mencionó es cada 5 segundos.

¹⁷ API. La interfaz de programación de aplicaciones, conocida también por la sigla API (del inglés application programming interface) es un conjunto de subrutinas, funciones y procedimientos que ofrece cierta biblioteca para ser utilizado en la comunicación entre componentes de software

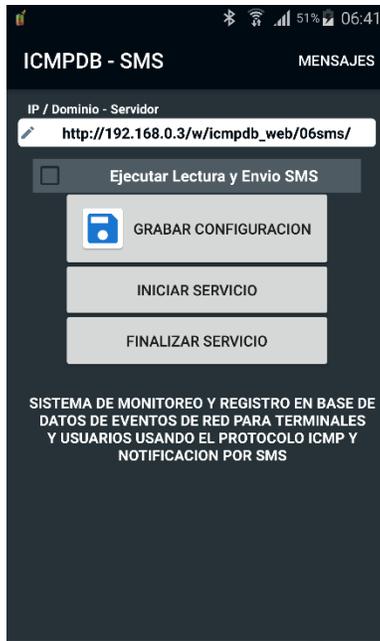


Figura 30: Panel de Control Principal

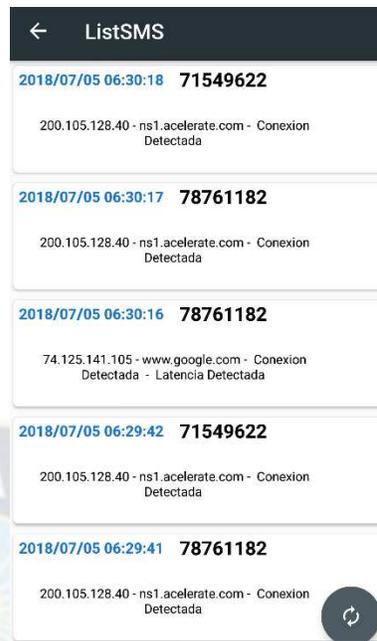
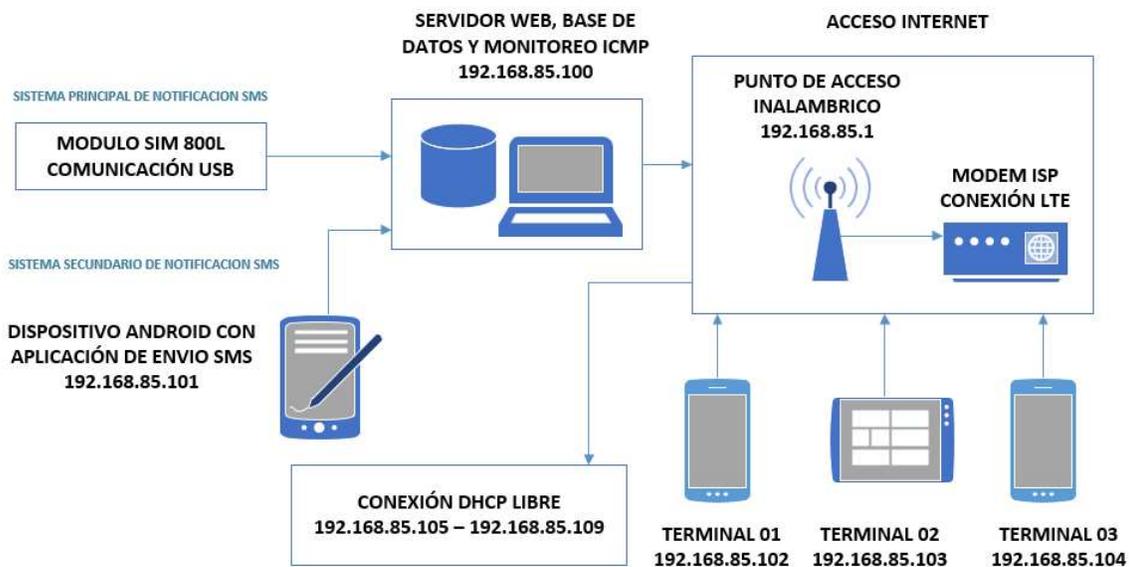


Figura 31. Listado de Mensajes Enviados desde la aplicación

3. EVALUACIÓN

3.1. ANÁLISIS DE RESULTADOS

Con el fin de probar el sistema en una red real se instaló el mismo en una red con la siguiente Topología de Red ¹⁸ para 5 dispositivos con sus respectivas IPs.



¹⁸ Topología de Red: se define como el mapa físico o lógico de una red para intercambiar datos. En otras palabras, es la forma en que está diseñada la red, sea en el plano físico o lógico

Figura 32: Esquema de Topología de RED de prueba de monitoreo

- CPU con Aplicaciones de SERVIDOR DE BASE DE DATOS Y HTTP
- 1 IP Punto de ACCESO INALAMBRICO y Modem ISP - LTE¹⁹.
- 1 IP Dispositivo Android con la Aplicación de envío de SMS.
- 3 IPS de equipos usuario.

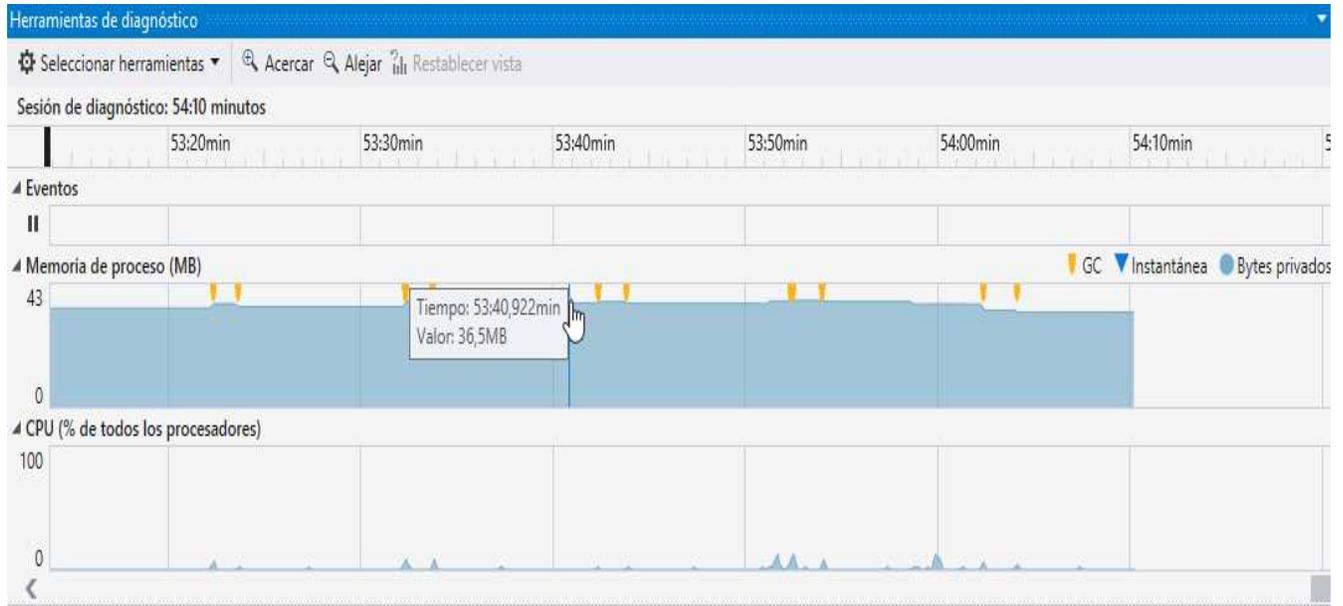


Figura 33: Recursos consumidos Memoria y Procesador

Realizado el monitoreo durante un lapso de 48 horas se puede verificar que el procesamiento no es significativo en la CPU de Monitoreo.

- PROCESAMIENTO DE CPU < 1%
- MEMORIA RAM UTILIZADA PARA 5 EQUIPOS 43 MBytes

¹⁹ LTE. (acrónimo de Long Term Evolution), es un estándar para comunicaciones inalámbricas de transmisión de datos de alta velocidad para teléfonos móviles y terminales de datos.

3.2. ANÁLISIS DE COSTOS

3.2.1.1. Costos de desarrollo e implementación

Referente al Software de Desarrollo tomando en cuenta que el proyecto fue diseñado y desarrollado usando software libre con licencia tipo GPL las licencias no tuvieron costos para su uso.

Referente al Hardware a continuación el costo de los componentes usados.

COMPONENTE	COSTO (Bs)
Protoboard	80
Placa Arduino Nano	150
Chip SIM 800L	250
Adaptador 5V	50
Cables de Conexión	20
Dispositivo Móvil prestaciones básicas	500
Equipo de Computación	6300
Total	7350

3.2.1.2. Costos de Prestación de Servicios

Los costos del servicio de SMS dependen del operador de telefonía móvil. A continuación, los costos encontrados para ENTEL, TIGO y VIVA

ENTEL – 0.20 Bs.

POSTPAGO SIMPLE

PLANES	Tarifa Básica Mensual (Bs)	Tarifa Minuto Horario Normal (Bs)	HORARIO NORMAL Días: Lunes a Domingo	Tarifa Minuto Horario Reducido (Bs)	HORARIO REDUCIDO Días: Lunes a Domingo	Tarifa por SMS (Bs)	Tarifa Fija Mensual WSLU (Bs)
			HORAS		HORAS		
Xsmall	30	1,20	5:00 a 23:59	0,66	00:00 a 4:59	0,20	80
Xsmall	50	1,10					80
Xsmallplus	75	1,00					80
Small	100	0,90					50
Smallplus	150	0,85					50
Medium	250	0,80					50
Large	350	0,80					50
Xlarge	450	0,80					50
Xxlarge	750	0,80					50

entel
ERVICIO / Junio de 2021 / ENTEL S.A.

NUEVATEL VIVA – 0.20 Bs.

SERVICIO MÓVIL - PLAN PRO WOW

Plan	Tarifa pago adelantado (Bs)	MB	Cuota de navegación	Tarifa por minuto de llamada (Bs)	Tarifa SMS	Tarifa MB adicional	Vigencia
PRO WOW	200	ilimitados	40 GB	1	0,20	0,50	28 días a partir de la habilitación

TELECEL – TIGO – 0.20 Bs.

tigo MÓVIL TELEVISIÓN INTERNET

**PLAN PREPAGO
LTE Inicial**

Disfruta desde tu activación de estas tarifas según el horario

- ✓ Tarifa Normal: Bs 1.60 el minuto.
- ✓ Tarifa Reducida: Bs 1.20 el minuto.
- ✓ Tarifa Super Reducida: Bs 0.80 el minuto.
- ✓ Tarifa por KB por 0,015 ctvs
- ✓ Tarifa por SMS a móviles nacionales por Bs 0,20

3.2.1.3. Costos de instalación y operación

HARDWARE

MODULO INSTALADO	COSTO (Bs.)
Modulo SMS armado y encapsulado	1000
Dispositivo Móvil prestaciones básicas (Opcional)	500
Equipo de Computación	2800
Total	4300

SERVICIO DE ALERTA

SERVICIO	COSTO (Bs.)
Servicio SMS promedio	0.20
TOTAL (1000 SMS)	200

3.3. CONCLUSIONES Y RECOMENDACIONES

El sistema considerará los siguientes aspectos:

- Monitoreo constante de las terminales ingresadas en sistema.
- Reporte de eventos de las terminales.
- Notificación por SMS de los eventos de red configurados.

El número de terminales a ser monitoreadas dependerá de las características del servidor de base de datos el cual como mínimo deberá usar un procesador de doble núcleo con el objetivo de buscar la máxima funcionalidad y confiabilidad del sistema de registro de eventos, que podría verse afectado si el número de terminales es demasiado grande.

Dado que el antes mencionado sistema de registro de eventos será desarrollado en VB.NET el mismo deberá ser instalado en sistemas con plataforma Microsoft que de preferencia deberá ser una versión para servidores.

El sistema de lectura de reporte de eventos solo podrá ser leído por dispositivos que tengan instalados navegadores con pleno funcionamiento y compatibilidad con HTML5 y Java Script de otra manera la gestión de esta información no se podrá garantizar.

El sistema de envío de notificaciones y eventos deberá estar instalado en un dispositivo con el sistema operativo Android y con permisos de envío de mensajería SMS.

Las terminales deben tener habilitado el protocolo ICMP para poder ser monitoreadas de otra manera el sistema las registrará como hosts ausentes. Dado que la aplicación principal del sistema estaría orientada a una intranet esto no debería afectar a las normas de seguridad de la red.

Por último, para una implementación comercial empresarial se debe tomar en cuenta que tanto la base de datos MySQL y la mayor parte de los lenguajes de programación usados son Open Source y de licencia GPL, excepto el VB.NET (Visual Basic. NET) para el cual tendría que adquirirse una licencia de desarrollo tipo empresa.

BIBLIOGRAFÍA

- [1]. DOMINGO GUILLERMO CARRASCO MUÑOZ. “Programación Orientada a Objetos con Visual Basic.NET 2008” - Empresa Editora Macro.
- [2]. SIM COM CORPORATION “SIM800L_Hardware_Design_V1.00” – SIM800 Datasheet
- [3]. Kevin Coshic “Resumen de Cisco CCNA Discovery 4.0” Copyright © 2010 Virtual Systems.
- [4]. SAUL J.ESCALERA. “Técnicas de I & O en Ciencias y Tecnología – Segunda Edición” – Academia Nacional de Ciencias de Bolivia.
- [5]. CARLOS MUÑOZ RAZO. “Cómo elaborar y asesorar una investigación de TESIS”.
- [6]. INTEL CORPORATION. “Intel® 64 and IA-32 Architectures Optimization Reference Manual”

WEBGRAFÍA

- [1]. <http://www.visualstudio.com/es-es/products/visual-studio-express-vs>. Descripción y descarga Visual Studio Express.
- [2]. <http://www.gnu.org/licenses/old-licenses/gpl-2.0-faq.es.html>. Preguntas frecuentes acerca de la versión 2 de la GPL de GNU para software LIBRE
- [3]. video2brain.com/mx/cursos/html5-animacion-e-interactividad-con-canvas. Desarrollo en HTML 5 de Jorge González Villanueva
- [4]. <http://dev.mysql.com/doc/refman/5.0/fr/gpl-license.html>. Licencia GNU para MYSQL
- [5]. <http://www.udemy.com/programacion-de-android-desde-cero> Curso de programación en Java para Sistemas Operativos Android y Gestión de envíos SMS

ANEXOS

CÓDIGO FUENTE MODULO DE CAPTURA DE EVENTOS LENGUAJE DE PROGRAMACIÓN VISUAL BASIC.NET

```
1 Imports System.ComponentModel
2 Imports System.Collections.Generic
3 Imports System.Collections.ObjectModel
4 Imports System.Reflection
5 Imports MySql.Data.MySqlClient
6 Imports System.Data
7 Imports System.Data.SqlClient
8 Imports System.Xml
9 Imports System.Net.Sockets
10 Imports System
11 Imports System.Diagnostics
12 Imports System.IO
13 Imports System.Net.NetworkInformation 14
15 Imports System.Windows.Forms
16 Imports System.Text
17 Imports System.Net
18 Imports System.Runtime.InteropServices
19 Imports System.Management
20 Imports System.IO.Ports
21 Public Class Frm01_Con
22     'DECLARACION DE VARIABLES
23     Private frmLoad As Boolean = True
24     Private Workers() As BackgroundWorker
25     Private IPWorkers(0) As String
26     Private IniPress As Boolean
27     Private iniWorkers(0) As Boolean
28     Private NumWorkers As Integer = 0
29     Private NumPings As Byte = 5
30     Private timeWait As Integer = 1000
31     Private tiempoSMS As Integer = 10000
32     Private query As New StringBuilder("Hello world", 350)
33     Private querytemp As New StringBuilder("Hello world", 350)
34     Private queryips As New StringBuilder("Hello world", 350) 35
36     Private DS_Temp As DataSet
37     Private DS_Temp_Count As Integer
38     Private DS_Temp_Tbl As System.Data.DataTable 39
40     Private DS_UsrIP As DataSet
41     Private DS_UsrIP_Count As Integer
42     Private DS_UsrIP_Tbl As System.Data.DataTable 43
44     Private DS_Log As DataSet
45     Private DS_Log_Count As Integer
46     Private DS_Log_Tbl As System.Data.DataTable 47
48     Private DS_DGValarm As DataSet
49     Private DS_DGValarm_Count As Integer
50     Private DS_DGValarm_Tbl As System.Data.DataTable 51
52     Private DS_DGVListHost As DataSet
```

```

53 Private DS_DGVListHost_Count AsInteger
54 PrivateDS_DGVListHost_TblAsSystem.Data.DataTable 55
56 Private DS_ListHost AsDataSet
57 Private DS_ListHost_Count AsInteger
58 PrivateDS_ListHost_TblAsSystem.Data.DataTable 59
60 Private DS_Process AsDataSet
61 Private DS_Process_Count AsInteger
62 PrivateDS_Process_TblAsSystem.Data.DataTable 63
64 PrivatecontPing01AsInteger
65 PrivatecontPing02AsInteger
66 PrivatecontPing03AsInteger 67
67 Private ConAct1 As Boolean = False
68 Private ConAnt1 As Boolean = False
69 Private ConAct2 As Boolean = False
70 Private ConAnt2 As Boolean = False
71 Private ConAct3 As Boolean = False
72 Private ConAnt3 As Boolean = False
73 Private ConAnt3 As Boolean = False
74
75
76 'COMBO BOX
77 Private DS_cmb_Grupolps AsDataSet
78 'SEND SMS
79 DelegateSubSetTextCallback(ByVal[text]AsString)'Addedto prevent threading errors during receiveing ofdata
80 PrivateSubFrm01_Con_Load(senderAsObject,eAsEventArgs)Handles MyBase.Load
81 frmLoad =True
82 IniPress =False
83
84 contPing01 =0
85 contPing02 =0
86 contPing03 =0
87 Btn_Parar_SMS.Enabled =False
88 Btn_Iniciar_SMS.Enabled =True
89 Cmb_TipoAlarm.SelectedIndex =0
90 Chk_Sound.Checked=My.Settings.AlarmSound 99
100 LlenarCombo(cmb_Grupolps, DS_cmb_Grupolps, "SELECT IDGrupo AS
ID,concat('Grp(',')(',')',Descrip,')')ASDescripFROM listgrupo ORDER
BYIDOrden DESC", "ID", "Descrip",-1)
101 Update_DGVListHost()
102 'SMS
103 UpdatePorts()
104 Cmb_DescripAT.SelectedIndex =0
105 TabPrin.TabPages.Remove(TabPage3)
106 Btn_StopSystem.Location = New Point(914,4)

107 frmLoad = False
108 End
109
110 #Region "DATA SETSEXECUTE"
111 Private Sub DS_Temp_Exec(query AsString)
112 DS_Temp = MySQLDS(query)
113

```

```

114         DS_Temp_Count = CInt(DS_Temp.Tables(0).Rows.Count)
115         DS_Temp_Tbl=DS_Temp.Tables(0)
116
117     End Sub
118     Private Sub DS_UsrIP_Exec(query AsString)
119         DS_UsrIP = MySqlDS(query)
120         DS_UsrIP_Count = CInt(DS_UsrIP.Tables(0).Rows.Count)
121         DS_UsrIP_Tbl=DS_UsrIP.Tables(0) 122
122     End Sub
123     Private Sub DS_Log_Exec(query AsString)
124         DS_Log = MySqlDS(query)
125         DS_Log_Count = CInt(DS_Log.Tables(0).Rows.Count)
126         DS_Log_Tbl =DS_Log.Tables(0)
127
128
129     End Sub
130     Private Sub DS_DGVAlarm_Exec(query AsString)
131         DS_DGVAlarm = MySqlDS(query)
132         DS_DGVAlarm_Count =CInt(DS_DGVAlarm.Tables(0).Rows.Count)
133         DS_DGVAlarm_Tbl=DS_DGVAlarm.Tables(0)
134
135     End Sub
136     Private Sub DS_Process_Exec(query AsString)
137         DS_Process = MySqlDS(query)
138         DS_Process_Count =CInt(DS_Process.Tables(0).Rows.Count)
139         DS_Process_Tbl =DS_Process.Tables(0)
140     End Sub
141     Private Sub DS_DGVListHost_Exec(query AsString)
142         DS_DGVListHost = MySqlDS(query)
143         DS_DGVListHost_Count =CInt(DS_DGVListHost.Tables(0).Rows.Count)
144         DS_DGVListHost_Tbl=DS_DGVListHost.Tables(0) 145
145     End Sub
146     Private Sub DS_ListHost_Exec(query AsString)
147         DS_ListHost = MySqlDS(query)
148         DS_ListHost_Count =CInt(DS_ListHost.Tables(0).Rows.Count)
149         DS_ListHost_Tbl=DS_ListHost.Tables(0) 151
150
151     End
152 Sub 153
153     #End Region
154     Public Class ArgumentType
155         Public _Index As Integer
156         Public _IP AsString
157         Public _TimeMaxAsInteger
158         Public _TimeUpdAsInteger
159     End Class
160     Public Class ResultArgumentType
161         Public _Index AsInteger
162         Public _IP AsString
163         Public _timeResp AsInteger
164         Public _tipoResp AsString
165         Public _numRecibidos AsInteger
166     End Class
167     Private Sub Update_DGVListHost()

```

```

169         query.Remove(0, query.Length)
170         query.Append("SELECT      Listhost.IDHost AS IDHost")
171         query.Append(",listhost.IP AS IP")
query.Append(",CONCAT(listhost.Nombre,",listhost.Descrip)AS NombreDescrip ")
                AlertCon=1")
172
173         query.Append(", listhost.Monitoring AS Monitoreo")
174         query.Append(" FROM Listhost ")
175         query.Append(" WHERE IDGrupo=" & CInt(cmb_GrupoIps.SelectedValue))
176         query.Append(" ORDER BY Listhost.IDOrden ASC")
177         DS_DGVListHost_Exec(query.ToString)
178
179         DGVListHost.DataSource = DS_DGVListHost_Tbl
180         DGVListHost.Columns(0).Width = 50
181         DGVListHost.Columns(1).Width = 150
182         DGVListHost.Columns(2).Width = 200
183         DGVListHost.Columns(3).Width = 100
184     End Sub
185     Private Sub Update_DGVAlarm()
186         query.Remove(0, query.Length)
187         query.Append("SELECT")
188         query.Append(" Listhost.IP AS IP")
189         query.Append(",Listhost.HostName AS HostName")
190         query.Append(",Listhost.AlertCon AS AlertCon")
191         query.Append(",Listhost.AlertDes AS AlertDes")
192         query.Append(",Listhost.AlertLag AS AlertLag")
193         query.Append(",Listhost.AlertUsr AS AlertUsr")
194         query.Append(" FROM Listhost ")
195         Select Case CInt(Cmb_TipoAlarm.SelectedValue)
196             Case Is = 1
197                 query.Append(" WHERE AlertCon=1")
198             Case Is = 2
199                 query.Append(" WHERE AlertDes=1")
200             Case Is = 3
201                 query.Append(" WHERE AlertLag=1")
202             Case Is = 4
203                 query.Append(" WHERE AlertUsr=1")
204             Case Is = 0
205                 query.Append(" WHERE AlertUsr=1 OR AlertLag=1 OR AlertDes=1 OR
206
                Case Else
207                 Exit Select
208         End Select
209         query.Append(" ORDER BY Listhost.IDOrdenASC")
210         DS_DGVAlarm_Exec(query.ToString)
211
212         DGVAlarm.DataSource = DS_DGVAlarm_Tbl
213         DGVAlarm.Columns(0).Width=150
214         DGVAlarm.Columns(1).Width=150
215         DGVAlarm.Columns(2).Width=150
216

```

```

217
218
219     End
Sub 220
221     PrivateSub WorkerDoWork(sender As Object, e As System.ComponentModel.DoWorkEventArgs ➤
        System.ComponentModel.BackgroundWorker = DirectCast(sender, System.ComponentModel.BackgroundWorker) ➤
222         Dim MyWorker As System.ComponentModel.BackgroundWorker = DirectCast(sender, System.ComponentModel.BackgroundWorker)
223         Dim res As System.Net.NetworkInformation.PingReply
224         Dim tipoResp As String
225         Dim timeResp As Integer = 99999
226         Dim IPHost As String
227         Dim TimeUpd As Integer
228         Dim TimeMax As Integer
229         Dim eargs As ArgumentType = CType(e.Argument, ArgumentType)
230         Dim cont As Byte
231         Dim numRecibidos As Integer
232
233         IPHost = eargs._IP
234         TimeUpd = eargs._TimeUpd
235         TimeMax = eargs._TimeMax
236         Do While MyWorker.CancellationPending = False
237             timeResp = 0
238             numRecibidos = 0
239             tipoResp = "HDI"
240             For cont = 1 To NumPings
241
242                 Try
243                     res = TiempoHost(IPHost, TimeMax)
244                     If res.Status = NetworkInformation.IPStatus.Success Then
245                         timeResp = Cint(timeResp + res.RoundtripTime)
246                         tipoResp = "OK"
247                         numRecibidos = numRecibidos + 1
248
249                     Else
250                         tipoResp = "TEA"
251                     End If
252                 Catch
253                     tipoResp = "HDI"
254
255                 End Try
256             Next
257
258             If numRecibidos > 0 Then
259                 timeResp = Cint(timeResp / numRecibidos)
260             Else
261                 timeResp = -1
262             End If
263
264             Dim rargs As ResultArgumentType = New ResultArgumentType
265             rargs._timeResp = timeResp
266             rargs._tipoResp = tipoResp
267             rargs._numRecibidos = numRecibidos

```

```

268         rargs._IP = IPHost
269         rargs._Index = eargs._Index
270         MyWorker.ReportProgress(0, rargs)
271         Threading.Thread.Sleep(TimeUpd)
272
273     Loop
274     e.Cancel = True
275
276 End
Sub 277
278 PrivateSub WorkerProgressChanged(sender As Object, e As System.ComponentModel.ProgressChangedEventArgs ➤
    'FLUJOGRAMA PRINCIPAL-Revision de Tipo de Conexion y Alarmas 280 '-----
    -----
281     Dim FechaAct As String
282     Dim HoraAct As String
283     Dim FHAct As String
284     Dim FechaCalc As String
285     Dim MyWorker As System.ComponentModel.BackgroundWorker = DirectCast (sender, System.ComponentModel.BackgroundWorker) ➤
286     Dim rargs As ResultArgumentType = CType(e.UserState, ResultArgumentType) ➤
287     Dim IDTipoEvent As Byte
288     Dim IDDiagnostic As Integer
289     Dim numRecibidos As Integer = rargs._numRecibidos
290     Dim timeResp As Integer = rargs._timeResp
291     Dim IDHost As Integer = rargs._Index
292
293     Dim IDTipoEventDB As Byte
294     Dim IDDiagnosticDB As Integer
295     Dim numRecibidosDB As Integer
296     Dim timeRespDB As Integer
297     Dim diffTime As Long
298
299
300     If rargs._Index = CInt(My.Settings.selHost01) And CInt ➤
        (My.Settings.selHost01) <> 0 Then
301         Txt_Host01.Text = rargs._IP
302         ProcesarPingBackground(Txt_Ping01, rargs._IP, contPing01, ConAct1, ConAnt1, ➤
            Chk_Con01, Chk_Descon01, rargs._timeResp, rargs._tipoResp) ➤
303     End If
304     If rargs._Index = CInt(My.Settings.selHost02) And CInt ➤
        (My.Settings.selHost02) <> 0 Then
305         Txt_Host02.Text = rargs._IP
306         ProcesarPingBackground(Txt_Ping02, rargs._IP, contPing02, ConAct2, ConAnt2, ➤
            Chk_Con02, Chk_Descon02, rargs._timeResp, rargs._tipoResp) ➤
307     End If
308     If rargs._Index = CInt(My.Settings.selHost03) And CInt ➤
        (My.Settings.selHost03) <> 0 Then
309         Txt_Host03.Text = rargs._IP
310         ProcesarPingBackground(Txt_Ping03, rargs._IP, contPing03, ConAct3, ConAnt3, ➤
            Chk_Con03, Chk_Descon03, rargs._timeResp, rargs._tipoResp) ➤
311     End If

```

```

312
313
314 FechaAct = Today.ToString("yyyy-MM-dd")
315 HoraAct = TimeOfDay.ToString("HH:mm:ss")
316 FHAct=Today.ToString("yyyy-MM-dd")&" "&TimeOfDay.ToString("HH:mm")
317 FechaCalc=Today.ToString("yyyyMMdd")&TimeOfDay.ToString("HHmmss")
318 If numRecibidos = NumPingsThen
319     IDTipoEvent = 1
320     If iniWorkers(IDHost) Then
321         IDTipoEvent = 11
322     End If
323 ElseIf numRecibidos < NumPings And numRecibidos > 0 Then
324     IDTipoEvent = 2
325     If iniWorkers(IDHost) Then
326         IDTipoEvent = 12
327     End If
328 ElseIf numRecibidos = 0 Then
329     IDTipoEvent = 3
330     If iniWorkers(IDHost) Then
331         IDTipoEvent = 13
332     End If
333 End If
334 'IDDiagnostico
335 Select Case timeResp
336     Case Is < 0
337         IDDiagnostico = 0
338     Case 0 To 10
339         IDDiagnostico = 10
340     Case 11 To 100
341         IDDiagnostico = 100
342     Case 101 To 1000
343         IDDiagnostico = 1000
344     Case Is > 1000
345         IDDiagnostico = 10000
346
347 End
348
349 'REVISION DE ALARMAS
350 '-----
351 Dim MonCon, MonDes, MonLag, MonUsr As Boolean
352 Dim AlertCon, AlertDes, AlertLag, AlertUsr As Boolean
353 Dim AlertReadCon, AlertReadDes, AlertReadLag, AlertReadUsr As Boolean
354 Dim FechaCalcDB As String
355 Dim tolerancia As Integer
356 Dim esperaUsr As Integer
357 MonCon = False
358 MonDes = False
359 MonLag = False
360 MonUsr = False
361 AlertCon = False
362 AlertDes = False
363 AlertLag = False

```

```

364 AlertUsr =False
365 query.Remove(0, query.Length)
366 query.Append(" SELECT MonCon,MonDes,MonLag,MonLagTime,")
367 query.Append(" MonUsr,MonUsrHora,MonUsrMin,MonUsrSeg,Tolerancia,IP,")
368 query.Append(" AlertCon,AlertDes,AlertLag,AlertUsr")
369 query.Append(" FROM Iisthost WHERE IDHost=" & IDHost & """) 370
371 DS_Temp_Exec(query.ToString)
372 If DS_Temp_Count > 0 Then
373     MonCon=CBool(DS_Temp_Tbl.Rows(0)("MonCon"))
374     MonDes=CBool(DS_Temp_Tbl.Rows(0)("MonDes"))
375     MonLag=CBool(DS_Temp_Tbl.Rows(0)("MonLag"))
376     MonUsr=CBool(DS_Temp_Tbl.Rows(0)("MonUsr"))
377     AlertReadCon=CBool(DS_Temp_Tbl.Rows(0)("AlertCon"))
378     AlertReadDes=CBool(DS_Temp_Tbl.Rows(0)("AlertDes"))
379     AlertReadLag=CBool(DS_Temp_Tbl.Rows(0)("AlertLag"))
380     AlertReadUsr=CBool(DS_Temp_Tbl.Rows(0)("AlertUsr"))
381     esperaUsr = CInt(Conv2Digit(CInt(DS_Temp_Tbl.Rows(0)
        ("MonUsrHora"))&Conv2Digit(CInt(DS_Temp_Tbl.Rows(0)
        ("MonUsrMin")) & Conv2Digit(CInt(DS_Temp_Tbl.Rows(0)
        ("MonUsrSeg")))))
382     tolerancia = CInt(DS_Temp_Tbl.Rows(0)("Tolerancia"))
383     If MonCon Then
384         If IDTipoEvent = 1 Or IDTipoEvent = 2 Then
385             AlertCon = True
386         End If
387     End If
388     If MonDes Then
389         If IDTipoEvent = 3 Then
390             AlertDes = True
391         End If
392     End If
393     If MonLag Then
394         If timeResp >= CInt(DS_Temp_Tbl.Rows(0)("MonLagTime")) Then
395             AlertLag = True
396         End If
397     End If
398     If MonUsr Then
399         If IDTipoEvent = 1 Or IDTipoEvent = 2 Then
400
401             query.Remove(0, query.Length)
402             query.Append(" SELECT FechaCalc FROM eventos WHERE
                IDHost=" & IDHost & ""
403                 AND (IDTipoEvent = 1 OR IDTipoEvent=2) AND Fecha=" &
                FechaAct & "" ORDER BY ID ASC LIMIT 0,1")
404             DS_Temp_Exec(query.ToString)
405             If DS_Temp_Count > 0 Then
406                 FechaCalcDB = CType(DS_Temp_Tbl.Rows(0)("FechaCalc"),
                String)
407                 diffTime = CLng(FechaCalc) - CLng(FechaCalcDB)
408                 If diffTime > esperaUsr Then
409
410                     query.Remove(0, query.Length)
411                     query.Append(" SELECT IDUsr FROM usr_ip WHERE

```

```

IDHost="" & IDHost & "" AND Active = 1")
412 DS_UsrIP_Exec(query.ToString)
413 If DS_UsrIP_Count > 0 Then
414     AlertUsr = True
415     For i = 0 To DS_UsrIP_Count - 1
416         query.Remove(0, query.Length)
417         query.Append("SELECT FechaCalc FROM
usr_log WHERE idusr="" & DS_UsrIP_Tbl.Rows(i)
(IDUsr").ToString & "" AND IP="" & rargs._IP & """)
DS_Log_Exec(query.ToString)
419
418
419     If DS_Log_Count > 0 Then
420
421         diffTime = CLng(FechaCalc) - CLng
(DS_Log_Tbl.Rows(0)("FechaCalc"))
422         If diffTime < 60000 Then 'Tiempo de
Logueo maximo 6 horas
423             AlertUsr = False
424         End If
425
426     End If
427 Next
428 End If
429 End If
430 End If
431 End If
432 End If
433
434 End If
435
436 query.Remove(0, query.Length)
437 query.Append(" UPDATE listhost SET")
438 query.Append("AlertCon=" & ConvBooleanToBin(AlertCon))
439 query.Append(",AlertDes=" & ConvBooleanToBin(AlertDes))
440 query.Append(",AlertLag=" & ConvBooleanToBin(AlertLag))
441 query.Append(",AlertUsr=" & ConvBooleanToBin(AlertUsr))
442 If AlertCon And Not (AlertReadCon) Then
443     query.Append(",FHCon="" & FHAct & """)
444 End If
445 If AlertDes And Not (AlertReadDes) Then
446     query.Append(",FHDes="" & FHAct & """)
447 End If
448 If AlertLag And Not (AlertReadLag) Then
449     query.Append(",FHLag="" & FHAct & """)
450 End If
451 If AlertUsr And Not (AlertReadUsr) Then
452     query.Append(",FHUsr="" & FHAct & """)
453 End If
454 query.Append(" WHERE IDHost="" & IDHost & """)

```

```

455     MySqlQuery(query.ToString)
456     'REGISTRO DE EVENTO
457     '-----
458     'VERIFICAR DATO ANTERIOR
459     '-----
460     Dim AddReg As Boolean
461     AddReg =
False 462
463     If iniWorkers(IDHost)
Then 464
465         AddReg = True
466         iniWorkers(IDHost) = False
467     Else
468         query.Remove(0, query.Length)
469         query.Append("SELECT * FROM eventos WHERE IDHost='" & IDHost & "' ORDER BY
ID DESC LIMIT 0,1")
470         DS_Temp_Exec(query.ToString)
471         If DS_Temp_Count = 0 Then
472             AddReg = True
473         Else
474             IDTipoEventDB = CByte(DS_Temp_Tbl.Rows(0)("IDTipoEvent"))
475             IDDiagnosticDB = CInt(DS_Temp_Tbl.Rows(0)("IDDiagnostic"))
476             numRecibidosDB = CInt(DS_Temp_Tbl.Rows(0)("NumRecibidos"))
477             timeRespDB = CInt(DS_Temp_Tbl.Rows(0)("Tiempo"))
478             diffTime = System.Math.Abs(timeResp - timeRespDB)
479             If diffTime > tolerancia Or IDTipoEvent <> IDTipoEventDB Or numRecibidos
<> numRecibidosDB Then
480                 AddReg = True
481             End If
482         End If
483     End If
484 End If
485
486
487     If AddReg Then
488         'GUARDAR LA INFORMACION
489         query.Remove(0, query.Length)
490         query.Append("INSERT INTO eventos("
491         query.Append("IDHost,IDTipoEvent,Tiempo,IDDiagnostic,Fecha,Hora,NumPings,Num
Recibidos,FechaHora,FechaCalc")
492         query.Append(") VALUES ("
493         query.Append("'" & IDHost & "','" & IDTipoEvent & "','" & timeResp & "','" &
IDDiagnostic & "','"
494         query.Append(FechaAct & "','" & HoraAct & "','" & NumPings & "','" & numRecibidos
& "','" & FechaAct & " " & HoraAct & "','" & FechaCalc & "''")
495         query.Append(")")
496         MySqlQuery(query.ToString)
497     End If
498
499
500
501

```

```

502
503
504 End Sub
505 Private Sub StartButton()
506     Btn_StartSystem.Visible = False
507     Btn_StopSystem.Visible = True
508 End Sub
509 Private Sub StopButton()
510     Btn_StartSystem.Visible = True
511     Btn_StopSystem.Visible = False
512 End
513 Sub
514 Private Sub ProcesarPingBackground(ByRef Txt_Ping As TextBox, ByVal
HostStr As String, ByRef contPing As Integer, ByRef ConAct As
Boolean, ByRef ConAnt As Boolean, ByVal ChkCon As CheckBox, ByRef
ChkDescon As CheckBox, ByVal timeResp As Integer, ByVal tipoResp
As String)
515
516     contPing = contPing + 1
517     If tipoResp = "OK" Then
518         If timeResp > 1 Then
519             Txt_Ping.Text = Txt_Ping.Text & Conv5Digit(contPing) & " Respuesta desde "
& Trim(HostStr) & "Tiempo" & timeResp & " ms" & vbCrLf
520         Else
521             Txt_Ping.Text = Txt_Ping.Text & Conv5Digit(contPing) & " Respuesta desde "
& Trim(HostStr) & "Tiempo < 1 ms" & vbCrLf
522         End If
523         ConAct = True
524         ElseIf tipoResp = "TEA" Then
525             Txt_Ping.Text = Txt_Ping.Text & Conv5Digit(contPing) & "Tiempode Espera Agotado "
& vbCrLf
526             ConAct = False
527         ElseIf tipoResp = "HDI" Then
528             Txt_Ping.Text = Txt_Ping.Text & Conv5Digit(contPing) & "Hostde Destino
Inaccesible " & vbCrLf
529
530             ConAct = False
531         End If
532
533         If ConAct <> ConAnt Then
534             If ConAct Then
535                 If ChkCon.Checked Then
536                     NotifyIcon1.ShowBalloonTip(2000, "Notificacion - CPEConfig",
"CONECTADO con " & HostStr, ToolTipIcon.Info)
537
538                     My.Computer.Audio.Play(My.Resources.ConnectionOK,
AudioPlayMode.WaitToComplete)
539                     'My.Computer.Audio.Stop()
540                 End If
541             Else
542                 If ChkDescon.Checked Then

```

```

544         NotifyIcon1.ShowBalloonTip(2000, "Notificacion -
CPEConfig", "DESCONEXION con " & HostStr,
ToolTipIcon.Warning)
545         My.Computer.Audio.Play(My.Resources.Disconnected,
AudioPlayMode.WaitToComplete)
546     End If
547 End If
548 End If
549
550
551     ConAnt = ConAct
552     Txt_Ping.Select(Txt_Ping.Text.Length, 0)
553     Txt_Ping.ScrollToCaret() 554
555 End Sub
556 Private Sub WorkerCompleted(sender As Object, e As
System.ComponentModel.RunWorkerCompletedEventArgs)

567 End
Sub 568
569 Private Sub Btn_StartSystem_Click(sender As Object, e As
EventArgs) Handles Btn_StartSystem.Click
570     StartButton()
571     IniPress = True
572
573     ModificarList
a() 573
574     With Bkw_Update
575         If .IsBusy = FalseThen
576             .WorkerReportsProgress = True
577             .WorkerSupportsCancellation = True
578             .RunWorkerAsync()
579         End If
580
581     End With
582 End
Sub 583
584 Private Sub Bkw_Update_DoWork(sender As Object, e As
DoWorkEventArgs) Handles Bkw_Update.DoWork
585
586     Dim Cont As Integer
= 0 587
588     With Bkw_Update
589         Do While .CancellationPending = False
590             Cont = Cont + 1
591             .ReportProgress(Cont, "OK")
592             Threading.Thread.Sleep(5000) 593
594         Loop
595
596     End
597
598

```

```

614         ip = IPAddress.Parse(IPHost)

608
609
610     End Sub
611     Private Function TiempoHost(ByVal IPHost As String, ByVal TimeMax As
        Integer) As System.Net.NetworkInformation.PingReply
612         Dim eco As New System.Net.NetworkInformation.Ping
613         Dim ip As IPAddress
615         TiempoHost = eco.Send(ip, TimeMax)
616     End Function
617     Private Function Conv5Digit(ByVal Num As Integer) As String
618         Conv5Digit = Trim(Str(Num))
619
620         Do While Conv5Digit.Length < 5
621             Conv5Digit = "0" & Conv5Digit
622         Loop
623     End Function
624     Private Function Conv2Digit(ByVal Num As Integer) As String
625         Conv2Digit = Trim(Str(Num))
626
627         Do While Conv2Digit.Length < 2
628             Conv2Digit = "0" & Conv2Digit
629         Loop
630     End Function
631
632     Private Sub Btn_StopSystem_Click(sender As Object, e As EventArgs) Handles
        Btn_StopSystem.Click
633         StopButton()
634         Call DetenerWorkers()
635         Bkw_Update.CancelAsync()
636     End Sub
637     Private Sub DetenerWorkers()
638         Dim NumRows As Integer
639         Dim IDHost As Integer
640         For NumRows = 0 To DS_ListHost_Count - 1
641             Dim eargs As ArgumentType = New ArgumentType()
642             IDHost =
CInt(DS_ListHost_Tbl.Rows(NumRows)("IDHost")) 643
                Workers(IDHost).CancelAsync()
644         Next
645
646
647     End Sub
648
649     Private Sub Bkw_Update_ProgressChanged(sender As Object, e
        As ProgressChangedEventArgs) Handles
        Bkw_Update.ProgressChanged
650
651         Dim LanIP As IPAddress
652         Dim IPStr As String
653         Dim Cont As Integer
654         Dim IDHost As Integer

```

```

655         Dim host As IPHostEntry
656
657     If IniPress Then
658         Dim iniWorkerRes As Boolean = False
659         Dim iniWorkerAct As Boolean = False
660         For i = 0 To NumWorkers
661             iniWorkerAct = iniWorkers(i)
662             iniWorkerRes = iniWorkerAct Or iniWorkerRes 663
663         Next
664         If iniWorkerRes = False
665             IniPress = False
666         End If
667     End If
668     queryips.Remove(0, queryips.Length)
669     queryips.Append("SELECT Listhost.IDHost AS
IDHost") 670 queryips.Append(",Listhost.IP AS IP")
671 queryips.Append(",Listhost.HostName AS
HostName") 672 queryips.Append(",Listhost.Monitoring AS
Monitoreo") 673 queryips.Append(" FROM Listhost ")
674 queryips.Append(" WHERE Process = 0 ")
675 queryips.Append(" ORDER BY Listhost.IDOrden
ASC") 676 DS_Process_Exec(queryips.ToString)
677     If DS_Process_Count > 0 Then
678         Call DetenerWorkers()
679
680         For Cont = 0 To DS_Process_Count -1
681             IDHost = CInt(DS_Process_Tbl.Rows(Cont)("IDHost"))
682             IPStr = CType(DS_Process_Tbl.Rows(Cont)("IP"),String)
683             queryips.Remove(0, queryips.Length)
684             'VALIDACION DE IPS
685             Try
686                 LanIP = IPAddress.Parse(IPStr)
687                 queryips.Append("UPDATE listhost SET IP=" & LanIP.ToString &
688                 "",IPValid=1,Revision=IP
689                 Valido',Process=1,AlertCon=0,AlertDes=0,AlertLag=0 WHERE IDHost=" &
690                 IDHost)
691
692             Catch ex1 As Exception
693             Try
694                 host = Dns.GetHostEntry(IPStr)
695                 Dim ip As IPAddress() = host.AddressList
696                 queryips.Append("UPDATE listhost SET IPValid=1,IP=" &
697                 ip(0).ToString() & "',Revision='Host Validado por el DNS',HostName=" &
698                 & IPStr & "',Process=1,AlertCon=0,AlertDes=0,AlertLag=0
699                 ")
700                 queryips.Append(" WHERE IDHost=" & IDHost)
701                 Call ModificarLista()
702             Catch ex2 As Exception
703                 queryips.Append("UPDATE listhost Set IPValid=0,Process=1,
704                 Revision='Error de IP',AlertCon=0,AlertDes=0,AlertLag=0
705                 WHERE IDHost=" & IDHost)
706             End Try
707         End Try
708     End Try

```

```

700
701         MySQLQuery(queryips.ToString)
702     Next
703     Call ModificarLista()
704
705 End If
706 Call Update_DGVAlarm()
707 If DS_DGVAlarm_Count > 0 Then
708     If Chk_Sound.Checked Then
709         NotifyIcon1.ShowBalloonTip(2000,"Notificacion-ICMPDB", "Alarmas
710             Activas ", ToolTipIcon.Warning)
711         My.Computer.Audio.Play(My.Resources.Disconnected,
712             AudioPlayMode.WaitToComplete)
713     End If
714 End Sub
715
716 Private Sub ModificarLista() 717
717     Dim NumRows As Integer
718     Dim IDHost As Integer
719     Dim IPHost As String
720     Dim TimeMax As Integer
721     Dim TimeUpd As Integer
722     Dim IPValid As Byte
723     Dim Monitoreo As Byte
724
725
726     query.Remove(0, query.Length)
727     query.Append("SELECT Listhost.IDHost AS
728 IDHost") 728 query.Append(",Listhost.IP AS IP")
729     query.Append(",Listhost.HostName AS
730 HostName") 730 query.Append(",Listhost.IPValid AS
731 IPValid")
732     query.Append(",Listhost.Monitoring AS
733 Monitoreo") 732 query.Append(",Listhost.TimeMax AS
734 TimeMax")
735     query.Append(",Listhost.TimeUpd AS
736 TimeUpd") 734 query.Append(" FROM Listhost ")
737     query.Append(" ORDER BY Listhost.IDHost
738 ASC") 736 DS_ListHost_Exec(query.ToString)
739
740     NumWorkers = DS_ListHost_Tbl.Rows(DS_ListHost_Count -
741 1)("IDHost") 739 ReDim Workers(NumWorkers)
742     ReDim InIWorkers(NumWorkers)
743     For i = 0 To
744 NumWorkers 742 InIWorkers(i) =
745 False 743 Next
746     For NumRows = 0 To DS_ListHost_Count - 1
747     Dim eargs As ArgumentType = New ArgumentType()
748     IDHost = DS_ListHost_Tbl.Rows(NumRows)("IDHost")
749     Monitoreo =
750 DS_ListHost_Tbl.Rows(NumRows)("Monitoreo") 748 IPHost =

```

```

DS_ListHost_Tbl.Rows(NumRows)("IP")
749         IPValid = DS_ListHost_Tbl.Rows(NumRows)("IPValid")
750         TimeMax = DS_ListHost_Tbl.Rows(NumRows)("TimeMax")
751         TimeUpd =
DS_ListHost_Tbl.Rows(NumRows)("TimeUpd") 752
753         eargs._Index = IDHost
754         eargs._IP = IPHost
755         eargs._TimeMax = TimeMax
756         eargs._TimeUpd =
TimeUpd 757
758
759         'IPWorkers(IDHost) = IPHost
760         Workers(IDHost) = NewBackgroundWorker
761         Workers(IDHost).WorkerReportsProgress = True
762         Workers(IDHost).WorkerSupportsCancellation = True
763         AddHandlerWorkers(IDHost).DoWork,AddressOfWorkerDoWork 764
AddHandlerWorkers(IDHost).ProgressChanged,AddressOf
WorkerProgressChanged
765         AddHandler Workers(IDHost).RunWorkerCompleted,
AddressOf WorkerCompleted
766
767
768         If Monitoreo = 1 And IPValid = 1
Then 769         If IniPress Then
770             InIWorkers(IDHost) =
True 771         End If
772
773         If Not (Workers(IDHost).IsBusy)Then
774             Workers(IDHost).WorkerReportsProgress = True
775             Workers(IDHost).WorkerSupportsCancellation = True
776             Workers(IDHost).RunWorkerAsync(eargs)
777         End If
778     End If
779     Next
780 End Sub
781
782 PrivateSubcmb_Grupolps_SelectedIndexChanged(senderAsObject,eAs EventArgs)
Handlescmb_Grupolps.SelectedIndexChanged
783     If frmLoad Then Exit Sub
784     Update_DGVListHost()
785 End Sub
786
787 Private Sub btn_Refresh_Click(sender As Object, e As EventArgs) Handles
btn_Refresh.Click
788     Update_DGVListHost()
789 End Sub
790
791 Private Sub DGVListHost_CellClick(sender As Object, e As
DataGridViewCellEventArgs) Handles
DGVListHost.CellClick 792     txt_IP.Text =
DGVListHost.Item("IP",
DGVListHost.CurrentRow.Index).Value
793     txt_HostName.Text =

```

```

        DGVListHost.Item("NombreDescrip",
        DGVListHost.CurrentRow.Index).Value
794     txt_IDHost.Text =
        DGVListHost.Item("IDHost",
        DGVListHost.CurrentRow.Index).Value
795     End Sub
796
797     Private Sub btn_SendHost01_Click(sender As Object, e As EventArgs)
        Handles btn_SendHost01.Click
798         If txt_IDHost.Text = "" Then Exit
Sub 799         Txt_Host01.Text = txt_IP.Text
800         My.Settings.selHost01 =
Val(txt_IDHost.Text) 801 My.Settings.Save()
802
803     End Sub
804
805     Private Sub btn_SendHost02_Click(sender As Object, e As EventArgs)
        Handles btn_SendHost02.Click
806         If txt_IDHost.Text = "" Then Exit
Sub 807         Txt_Host02.Text = txt_IP.Text
808         My.Settings.selHost02 =
Val(txt_IDHost.Text) 809 My.Settings.Save()
810     End Sub
811
812     Private Sub btn_SendHost03_Click(sender As Object, e As EventArgs)
        Handles btn_SendHost03.Click
813         If txt_IDHost.Text = "" Then Exit Sub
814         Txt_Host03.Text = txt_IP.Text
815         My.Settings.selHost03 = Val(txt_IDHost.Text)
816         My.Settings.Save()
817     End Sub
818
819     Private Sub Chk_Sound_CheckedChanged(sender As Object, e As EventArgs)
        Handles Chk_Sound.CheckedChanged
820         If frmLoad Then Exit Sub
821         My.Settings.AlarmSound =
Chk_Sound.Checked 822 My.Settings.Save()
823     End Sub
824
825
826
992 End Class

```

CÓDIGO FUENTE MÓDULO DE GESTIÓN DE DATOS Y CONFIGURACIÓN DE SISTEMA LENGUAJE DE PROGRAMACIÓN HTML Y JAVASCRIPT CON JQUERY

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8" lang="es-Es">
5 <title>icmpdb</title>
6 <link rel="stylesheet" type="text/css" href="../../jquery-easyui-1.4.4_lon/themes/azulito/easyui.css">
7 <link rel="stylesheet" type="text/css" href="../../jquery-easyui-1.4.4_lon/themes/icon.css">
8 <link rel="stylesheet" type="text/css" href="../../jquery-easyui-1.4.4_lon/themes/color.css">
9 <link rel="stylesheet" type="text/css" href="../../jquery-easyui-1.4.4_lon/demo/demo.css">
10 <script type="text/javascript" src="../../jquery-easyui-1.4.4_lon/jquery.min.js"></script>
11 <script type="text/javascript" src="../../jquery-easyui-1.4.4_lon/jquery.easyui.min.js"></script>
12 <script type="text/javascript" src="../../jquery-easyui-1.4.4_lon/datagrid-detailview.js"></script>
13 <script type="text/javascript" src="../../js/jqueryui.js"></script>
14 <script type="text/javascript" src="gesips.js"></script>
15 <script type="text/javascript" src="gesipsLoad.js"></script>
16
17 </head>
18 <body>
19 <div id="layoutp" class="easyui-layout" style="width:1010px;height:565px;">
20 <div region="west" split="true" title="GRUPOS DE ASIGNACION" style="width:350px;height:565px;" iconCls="icon-search" >
21
22 <table id="dg_grp"
23 class="easyui-datagrid" style="width:340px;height:515px" url="get_listgrp.php"
24 toolbar="#tbar_grp" pagination="true" showFooter="false"
25 rownumbers="true" fitColumns="false" singleSelect="true" >
26 <thead>
27 <tr>
28 <th field="dgvid" width="35px">ID</th>
29 <th field="dgvdescrip" width="200px">Nombre del Grupo</th>
30 <th field="dgvactive" width="27px">Hab</th>
31 </tr>
32 </thead>
33 </table>
34 <div id="tbar_grp">
35 <a href="javascript:void(0)" class="easyui-linkbutton easyui-tooltip"
36 title="Adicionar Grupo" iconCls="icon-add" plain="true" onClick="newGrp()">Adicionar</a>
37
38 <a href="javascript:void(0)" class="easyui-linkbutton easyui-tooltip"
39 title="Editar Grupo" iconCls="icon-edit" plain="true" onClick="editGrp()">Editar</a>
40
41 <a href="javascript:void(0)" class="easyui-linkbutton easyui-tooltip"
42 title="Eliminar Grupo" iconCls="icon-remove" plain="true" onClick="delGrp()">Eliminar</a>
43
44 </div>
45 <div id="dlg_grp" class="easyui-dialog" style="width:250px;height:350px;;padding:0px 0px"
46 closed="true" buttons="#dlgbtn_grp">
47 <!--<div class="ftitle">Datos de Registro</div-->
48 <form id="fm_grp" method="post" novalidate>
49 <table border="0" align="center" cellpadding="2" cellspacing="2" style="font-weight: bold; color: #000;">
50 <tr><td>Activar Edicion<input id="grpactive" name="grpactive" type="checkbox"> </td></tr>
51 <tr><td>Nombre del Grupo </td></tr>
52 <tr><td>
53 <input id="dgvdescrip" name="dgvdescrip" class="easyui-textbox" style="width:190px" prompt="Ej.
54 Administracion">
55 </td></tr>
56 </table>
57 </form>
58 </div>
59 <div id="dlgbtn_grp">
60 <a href="javascript:void(0)" class="easyui-linkbutton" iconCls="icon-ok" onClick="saveGrp()" style="width:90px">Guardar</a>
61 <a href="javascript:void(0)" class="easyui-linkbutton" iconCls="icon-cancel"
62 onClick="javascript:$('#dlg_grp').dialog('close')" style="width:90px">Cancelar</a>
63 </div><!--id="dlg-buttonsf"-->
64 </div><!--region="west"-->
```

```

65 <div id="content" region="center" title="LISTA DE IPS" style="padding:0px;">
66
67 <table id="dg_ips"
68 class="easyui-datagrid" style="width:620px;height:515px" url="get_listhost.php"
69 toolbar="#tbar_ips" pagination="true" showFooter="false"
70 rownumbers="true" fitColumns="false" singleSelect="true" >
71 <thead data-options="frozen:true">
72 <tr>
73 <th field="dgvip" width="100px">IP Asignado</th>
74 <th field="dgvhostname" width="100px">Nombre del Host</th>
75 </tr>
76 </thead>
77 <thead>
78 <tr>
79 <th field="dgvnombre" width="100px">Nombre BD</th>
80 <th field="dgvtimeamax" width="80px">Tiempo Max.</th>
81 <th field="dgvtimeupd" width="80px">Tiempo Upd.</th>
82 <th field="dgvtolerancia" width="80px">Tolerancia</th>
83 <th field="dgvvipvalid" width="55px">IP Valida</th>
84 <th field="dgvmonitoreo" width="65px">Mon.ICMP</th>
85 <th field="dgvmoncon" width="105px">Mon.Conexion</th>
86 <th field="dgvmondes" width="105px">Mon.Desconexion</th>
87 <th field="dgvmonlag" width="105px">Mon.Latencia</th>
88 <th field="dgvmonusr" width="105px">Mon.Usuario</th>
89 <th field="dgvipdescrip" width="100px">Descripcion</th>
90 <th field="dgvubicacion" width="100px">Ubicacion</th>
91 <th field="dgvvidgrp" width="50px">ID GRP</th>
92 </tr>
93 </thead>
94 </table>
95
96 <div id="tbar_ips">
97 <a href="javascript:void(0)" class="easyui-linkbutton easyui-tooltip"
98 title="Adicionar IP" iconCls="icon-add" plain="true" onClick="addIPs()">Adicionar</a>
99
100 <a href="javascript:void(0)" class="easyui-linkbutton easyui-tooltip"
101 title="Editar IP" iconCls="icon-edit" plain="true" onClick="editIPs()">Editar</a>
102
103 <a href="javascript:void(0)" class="easyui-linkbutton easyui-tooltip"
104 title="Eliminar IP" iconCls="icon-remove" plain="true" onClick="delIPs()">Eliminar</a>
105 <br>
106 <span>Mon. ICMP</span>
107 <input id="tbcmb_mon" style="width:100px" class="easyui-combobox"
108 name="tbcmb_mon"
109 url="cmbs/tbcmb_mon.php"
110 valueField="id" textField="text">
111 <a href="#" class="easyui-linkbutton" plain="true" onClick="IpsDoSearch()" iconCls="icon-search"></a>
112 </div>
113
114 <div id="dlg_ips" class="easyui-dialog" style="width:600px;height:500px;padding:0px 0px"
115 closed="true" buttons="#dlgbtn_ips">
116 <!--<div class="ftitle">Datos de Registro</div-->
117
118 <form id="fm_ips" method="post" novalidate>
119
120
121
122 <table border="0" align="center" cellpadding="3" cellspacing="0" style="font-weight: bold; color: #000;">
123
124 <tr><td><span style="padding:3px">Grupo De IPS</span></td><td>
125 <input id="dgvvidgrp" style="width:270px"
126 name="dgvvidgrp"
127 url="cmbs/cmb_listgrupo.php"
128 valueField="id" textField="text">
129 </td></tr>
130 <tr><td style="background-color:#FFC107"><span style="padding:2px">IP</span></td>
131 <td style="background-color:#FFC107">
132 <input id="dgvip" name="dgvip" class="easyui-textbox" style="width:200px">
133 </td></tr>
134 <tr><td><span style="padding:2px">Host Name o Dominio</span></td>
135 <td>
136 <input id="dgvhostname" name="dgvhostname" class="easyui-textbox" style="width:200px">
137 </td></tr>
138 <tr><td><span style="padding:2px">Tiempo Maximo de PING (ms)</span></td>
139 <td>
140 <input id="dgvtimeamax" name="dgvtimeamax" class="easyui-numberspinner" data-
141 options="min:1000,max:9000,increment:100" style="width:80px">
142 <span style="padding:2px">Tolerancia(ms)</span>
143 <input id="dgvtolerancia" name="dgvtolerancia" class="easyui-numberspinner" data-
144 options="min:10,max:9000,increment:5" style="width:80px">
145 </td></tr>

```

```

144 <tr><td><span style="padding:2px">Nombre</span></td>
145 <td>
146 <input id="dgvnombre" name="dgvnombre" class="easyui-textbox" style="width:200px">
147 </td></tr>
148 <tr><td><span style="padding:2px">Descripcion</span></td>
149 <td>
150 <input id="dgvipdescrip" name="dgvipdescrip" class="easyui-textbox" style="width:200px">
151 </td></tr>
152 <tr><td>Ubicacion</td><td>
153 <input id="dgvubicacion" name="dgvubicacion" class="easyui-textbox" style="width:270px;height:75px"
154 data-options="multiline:true"
155 prompt="(Campo Opcional) Ej. 2do Piso" >
156 </td></tr>
157 <tr><td style="background-color: #BDBDBD"><span style="padding:3px">IP Valida</span><input id="ipvalid"
158 name="ipvalid" type="checkbox" disabled="disabled">
159 <span style="padding:3px">Estado</span></td>
160 <td style="background-color: #BDBDBD">
161 <input id="dgvrevision" name="dgvrevision" class="easyui-textbox" style="width:150px;text-align: center"
162 readonly>
163 <span style="padding:3px">Monitoreo ICMP (Ping)</span><input id="monitoreo" name="monitoreo"
164 type="checkbox">
165 </td></tr>
166 <tr>
167 <td><span style="padding:3px">Tiempo Actualizacion (ms)</span></td>
168 <td><input id="dgvtimeupd" name="dgvtimeupd" class="easyui-numberspinner" data-
169 options="min:1000,max:100000,increment:100" style="width:80px"></td>
170 </tr>
171 <tr>
172 <td><span style="padding:3px">Monitoreo Conectividad</span></td>
173 <td>
174 Conexion<input id="moncon" name="moncon" type="checkbox">
175 Desconexion<input id="mondes" name="mondes" type="checkbox">
176 </td>
177 </tr>
178 <tr>
179 <td><span style="padding:3px">Monitoreo Latencia</span></td>
180 <td><input id="monlag" name="monlag" type="checkbox"></td>
181 </tr>
182 <tr>
183 <td><span style="padding:3px">Monitoreo Usuario</span><input id="monusr" name="monusr" type="checkbox">
184 </td>
185 <td>
186 <span style="padding:2px">Tiempo Espera - Hrs:</span>
187 <input id="dgvhora" name="dgvhora" class="easyui-numberspinner" data-
188 options="min:00,max:23,increment:1" style="width:50px">
189 <span style="padding:2px">Min:</span>
190 <input id="dgvmin" name="dgvmin" class="easyui-numberspinner" data-
191 options="min:00,max:59,increment:1" style="width:50px">
192 <span style="padding:2px">Seg:</span>
193 <input id="dgvseg" name="dgvseg" class="easyui-numberspinner" data-
194 options="min:00,max:59,increment:1" style="width:50px">
195 </td>
196 </tr>
197 </table>
198 </form>
199
200 </div>
201 <div id="dlgbtn_ips">
202 <a id="btn_saveIPs" href="javascript:void(0)" class="easyui-linkbutton" iconCls="icon-ok"
203 onClick="saveIPs()" style="width:90px">Guardar</a>
204
205 <a href="javascript:void(0)" class="easyui-linkbutton" iconCls="icon-cancel"
206 onClick="javascript:$('#dlg_ips').dialog('close')" style="width:90px">Cancelar</a>
207 </div>
208
209 </div>

```

```

210 <div data-options="region:'east',split:true" title="Usuarios Para Monitoreo" style="width:400px">
211 <table id="dg_usr"
212 class="easyui-datagrid" style="width:380px;height:515px" url="get_usrips.php"
213 toolbar="#tbar_usr" pagination="true" showFooter="false"
214 rownumbers="true" fitColumns="false" singleSelect="true" >
215 <thead>
216 <tr>
217 <th field="dgvnomcom" formatter="formatter_dgvnomcom" width="150px">Nombre</th>
218 <th field="dgvactive" width="70px">Monitoreo</th>
219 <th field="dgvactivesms" width="50px">SMS</th>
220 <th field="dgvfuncion" width="100px">Funcion</th>
221 <th field="dgvarea" width="150px">Area</th>
222 </tr>
223 </thead>
224 </table>
225 <div id="tbar_usr">
226 <a href="javascript:void(0)" class="easyui-linkbutton easyui-tooltip"
227 title="Adicionar Usuario" iconCls="icon-add" plain="true" onClick="newUsr()">Adicionar</a>
228 <a href="javascript:void(0)" class="easyui-linkbutton easyui-tooltip"
229 title="Editar Usuario" iconCls="icon-edit" plain="true" onClick="editUsr()">Editar</a>
230 <a href="javascript:void(0)" class="easyui-linkbutton easyui-tooltip"
231 title="Eliminar Usuario" iconCls="icon-remove" plain="true" onClick="delUsr()">Eliminar</a>
232
233 </div>

234 <div id="dlg_usr" class="easyui-dialog" style="width:350px;height:200px;padding:0px 0px"
235 closed="true" buttons="#dlgbtn_usr">
236 <!--<div class="ftitle">Datos de Registro</div-->
237 <form id="fm_usr" method="post" novalidate>
238 <table border="0" align="center" cellpadding="2" cellspacing="2" style="font-weight: bold; color: #000;">
239 <tr><td>Activar Monitoreo <input id="usractive" name="usractive" type="checkbox"> </td></tr>
240 <tr><td>Activar Envio SMS <input id="smsactive" name="smsactive" type="checkbox"> </td></tr>
241 <tr><td>Nombre Usuario</td></tr>
242 </tr>
243 <tr>
244 <td>
245 <input id="dgvidusr" style="width:250px"
246 name="dgvidusr">
247 </td>
248 </tr>
249 </form>
250 </div>

251 <div id="dlgbtn_usr">
252 <a href="javascript:void(0)" class="easyui-linkbutton" iconCls="icon-ok" onClick="saveUsr()" style="width:90px">Guardar</a>
253 <a href="javascript:void(0)" class="easyui-linkbutton" iconCls="icon-cancel"
254 onClick="javascript:$('#dlg_usr').dialog('close');" style="width:90px">Cancelar</a>
255 </div>
256 </div>
257
258
259 <style type="text/css">
260 <div>
261 #fm{
262 margin:0;
263 padding:0px 0px;
264 }
265 .ftitle{
266 font-size:12px;
267 font-weight:bold;
268 padding:4px 0;
269 margin-bottom:2px;
270 border-bottom:1px solid #ccc;
271 }
272 .fitem{
273 }
274 .fitem label{
275 display:inline-block;
276 width:80px;
277 }
278 .fitem input{
279 }
280 }
281 #tblidtiposervicio{
282 border: black 0px solid;
283 }
284 </style>
285 </body>
286 </html>

```

CÓDIGO FUENTE MÓDULO DE NOTIFICACIÓN POR SMS EN LENGUAJE DE PROGRAMACIÓN JAVA PARA ANDROID.

```
package com.alonzochoque.amdfxsony.icmpdb.Service;

import android.Manifest;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.AsyncTask;
import android.os.Build;
import android.os.IBinder;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.support.v4.app.ActivityCompat;
import android.support.v4.app.JobIntentService;
import android.telephony.SmsManager;
import android.util.Log;
import android.widget.Toast;

import com.alonzochoque.amdfxsony.icmpdb.MisClases.ParamWebService;
import com.alonzochoque.amdfxsony.icmpdb.Modelos.ConfigApp;
import com.alonzochoque.amdfxsony.icmpdb.Modelos.RegSms;
import com.alonzochoque.amdfxsony.icmpdb.R;
import com.alonzochoque.amdfxsony.icmpdb.Volley.AppController;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Map;

import io.realm.Realm;

public class MyJobIntentService extends JobIntentService {
    private String TAG = MyJobIntentService.class.getSimpleName();
    private MyJobTask myTask;
    Context context;
    private String runService;
    int tiempo = 5000;
    int contMax = 6;
    int contCiclo = 1;
}
```

```

//Realm
private Realm realm;
private ConfigApp configApp;
private SharedPreferences confSystem;
String serverIP;
//SMS-----
private SmsManager smsManager;
private static final int PERMISSIONS_SEND_SMS = 123;
//SMS-----

public static void enqueueWork(Context context, Intent intent){
    enqueueWork(context, MyJobIntentService.class, 17, intent);
}

@Override
public void onCreate() {
    super.onCreate();
    context = getApplicationContext();
    //Toast.makeText(this, "Servicio creado!",
Toast.LENGTH_SHORT).show();
    myTask = new MyJobTask();
}

@Override
public int onStartCommand(@Nullable Intent intent, int flags, int
startId) {
    return super.onStartCommand(intent, flags, startId);
}

@Override
public IBinder onBind(@NonNull Intent intent) {
    return super.onBind(intent);
}

@Override
protected void onHandleWork(@NonNull Intent intent) {
    myTask.execute();
}

private class MyJobTask extends AsyncTask<String, String, String> {

    private SimpleDateFormat dateFormat;
    private String date;

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        dateFormat = new SimpleDateFormat("HH:mm:ss");
        //Toast.makeText(getApplicationContext(), "Tarea iniciada!",
Toast.LENGTH_SHORT).show();
        confSystem = getSharedPreferences("confSystem",
getApplicationContext().MODE_PRIVATE);
        runService = confSystem.getString("runService", "0");
    }
}

```

```

@Override
protected String doInBackground(String... params) {

    Log.d(TAG + " INI",runService);
    while (runService.equals("1")){

        date = dateFormat.format(new Date());
        try {
            publishProgress(date);
            // Stop 5s
            Thread.sleep(tiempo);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

        runService = confSystem.getString("runService","0");
        Log.d(TAG + " CICLE " + date ,runService);
    }

    return null;
}

@Override
protected void onProgressUpdate(String... values) {
    Toast.makeText(getApplicationContext(), "ICMPDB Running\nHora
actual: " + values[0], Toast.LENGTH_SHORT).show();

    SolicitarActualizacion();
}

@Override
protected void onCancelled() {
    super.onCancelled();
}

@Override
protected void onPostExecute(String s) {
    super.onPostExecute(s);
    Toast.makeText(getApplicationContext(), "Tarea Finalizada ",
Toast.LENGTH_SHORT).show();
}

private void SolicitarEnvioSMS(String cel, String msg){
    if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.M){
        //requestPermissions(new
String[]{Manifest.permission.SEND_SMS},PERMISSIONS_SEND_SMS);
        ExecuteNewerVersions(cel,msg);
    }else{
        ExecuteOlderVersions(cel,msg);
    }
}

private boolean CheckPermission (String permission){
    int result =
context.checkCallingOrSelfPermission(permission);
    return result == PackageManager.PERMISSION_GRANTED;
}

```

```

private void ExecuteOlderVersions (String cel, String msg){
    if (CheckPermission(Manifest.permission.SEND_SMS)){
        EnviarSMS (cel,msg);
    } else {
        Toast.makeText (context, "Permiso Denegado para Envio de
SMS", Toast.LENGTH_SHORT).show();
    }
}
private void ExecuteNewerVersions(String cel, String msg){
    if (ActivityCompat.checkSelfPermission(context,
Manifest.permission.SEND_SMS)
    == PackageManager.PERMISSION_GRANTED) {
        EnviarSMS (cel,msg);
    } else {
        Toast.makeText (context, "Permiso Denegado para Envio de
SMS - Nueva Version", Toast.LENGTH_SHORT).show();
    }
}

}
//-----Procesar JSON-----//
private void ProcJSON_SMS(String result) {
    realm = Realm.getDefaultInstance();
    JSONObject respuestaJSON = null; //Creo un JSONObject a
partir del StringBuilder pasado a cadena
    try {
        respuestaJSON = new JSONObject(result);
        String total = respuestaJSON.getString("total");
        if (!total.equals("0")) {
            //
            //
            //
            //
            //
        }

        //Variable de Llegada
        int idsms;
        String ip;
        String hostName;
        String msg;
        String cel;
        String alias;
        String fhsendstr;
        //Variables de Confirmacion
        String idsmsconf = "";

        JSONArray rowsJSON = respuestaJSON.getJSONArray("sms");
        for (int i = 0; i < rowsJSON.length(); i++) {
            idsms =
Integer.parseInt (rowsJSON.getJSONObject(i).getString("idsms"));
            ip = rowsJSON.getJSONObject(i).getString("ip");
            hostName =
rowsJSON.getJSONObject(i).getString("hostName");
            msg = rowsJSON.getJSONObject(i).getString("msg");
            cel = rowsJSON.getJSONObject(i).getString("cel");
            alias = rowsJSON.getJSONObject(i).getString("alias");
            DateFormat dateFormat = new
SimpleDateFormat ("yyyy/MM/dd HH:mm:ss");
            Date date = new Date();
            fhsendstr = dateFormat.format (date).toString();

```

```

Log.d(TAG + " SMS:" ,msg);

RegSms regSms = new RegSms(idsms
    , ip
    , hostName
    , msg
    , cel
    , alias
    , fhsendstr
);

realm.beginTransaction();
realm.copyToRealmOrUpdate(regSms);
realm.commitTransaction();

try {
    Thread.sleep(1000);
    SolicitarEnvioSMS(cel,msg);
    if (i>0) idsmsconf += ",";
    idsmsconf += idsms;

} catch (InterruptedException e) {
    e.printStackTrace();
}

}
Log.d(TAG,idsmsconf);
ParamWebService paramWebService = new ParamWebService();
paramWebService.addParam("codsmsgs",idsmsconf);
VolleyDatos(serverIP + "conf_sms.php",paramWebService);

} catch (JSONException e) {
    e.printStackTrace();
    //etxtUser.setText(e.toString());
}
}
private void deleteRegSms() {
    realm = Realm.getDefaultInstance();
    realm.beginTransaction();
    realm.delete(RegSms.class);
    realm.commitTransaction();
}
private void EnviarSMS(String cel ,String msg){
    smsManager = SmsManager.getDefault();
    smsManager.sendTextMessage(cel, null, msg, null, null);
}
private void SolicitarActualizacion() {
    //Verificar Activacion ENVIO SMS
    //-----
    String sendSMS = "0";
    realm = Realm.getDefaultInstance();
    configApp = realm.where(ConfigApp.class).equalTo("param",
"sendSMS").findFirst();
    if (configApp != null){
        sendSMS = configApp.getValor();
    }
}

```

```

        Log.d(TAG + " SEND SMS" + date , sendSMS);
    }
    if (sendSMS.equals("0")) return;
    //-----
    serverIP=getResources().getString(R.string.ipserver1);
    configApp = realm.where(ConfigApp.class).equalTo("param",
"ipServer").findFirst();
    if (configApp != null){
        serverIP = configApp.getValor();
        Log.d(TAG + " SERVER IP" + date , serverIP);
    }
    date = dateFormat.format(new Date());
    //Toast.makeText(getApplicationContext(), "ICMPDB is
Running\nHora actual: " + date.toString() , Toast.LENGTH_SHORT).show();

    ParamWebService pupd = new ParamWebService();
    //pupd.addParam("cods","");

    ConnectivityManager cm = (ConnectivityManager)
context.getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo activeNetwork = cm.getActiveNetworkInfo();
    if (activeNetwork != null) { // connected to the internet

        if (activeNetwork.getType() ==
ConnectivityManager.TYPE_WIFI) {
            // connected to wifi
            //Toast.makeText(context,
activeNetwork.getTypeName(), Toast.LENGTH_SHORT).show();

            if (contCiclo > contMax){
                contCiclo = 0;
                //Toast.makeText(getApplicationContext(),
"UEELP", Toast.LENGTH_SHORT).show();
                //pupd.addParam("fupds","");
                Log.d(TAG,pupd.getURLParam(serverIP +
"send_sms"));
                VolleyUpdate(pupd.getURLParam(serverIP +
"send_sms"),1);
            }

        } else if (activeNetwork.getType() ==
ConnectivityManager.TYPE_MOBILE) {
            // connected to the mobile provider's data plan

            if (contCiclo > contMax*4){
                contCiclo = 0;
                //Toast.makeText(getApplicationContext(),
"UEELP", Toast.LENGTH_SHORT).show();
                Log.d(TAG,pupd.getURLParam(serverIP +
"send_sms"));
                VolleyUpdate(pupd.getURLParam(serverIP +
"send_sms"),1);
            }
        }
    } else {
        // not connected to the internet
    }
}

```

```

    }
    contCiclo++;
}

private void VolleyUpdate(String webURL, final int metodoJSON) {
    ConnectivityManager connMgr = (ConnectivityManager)
        getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();
    if (networkInfo != null && networkInfo.isConnected()) {
        // Tag used to cancel the request
        String tag_json_obj = "json_obj_req";

        String url = webURL;
        //Log.d("webURL",webURL);
        JsonObjectRequest jsonObjReq = new
        JsonObjectRequest(Request.Method.GET,
            url, null,
            new Response.Listener<JSONObject>() {

                @Override
                public void onResponse(JSONObject response) {
                    Log.d(TAG,response.toString());

                    switch (metodoJSON) {
                        case 1:
ProcJSON_SMS(response.toString());
                            break;
                        case 2:
//ProcJSON_Search(response.toString());
                            break;
                    }

                }
            }, new Response.ErrorListener() {

                @Override
                public void onErrorResponse(VolleyError error) {
                    Toast.makeText(getApplicationContext(), "Error de
Conexion", Toast.LENGTH_SHORT).show();
                    //CharSequence textError = "Error de Conexion";
                    //int duration = Toast.LENGTH_LONG;
                    //Toast toast = Toast.makeText(context,
textError, duration);
                    //toast.show();
                }
            });

        // Adding request to request queue
        ApplicationController.getInstance().addToRequestQueue(jsonObjReq,
tag_json_obj);
    }

}
//-----Volley -----//

```

```

private void VolleyDatos(String webURL, final ParamWebService
paramWebService) {
    ConnectivityManager connMgr = (ConnectivityManager)
        getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();
    if (networkInfo != null && networkInfo.isConnected()) {

        String url = webURL;

        RequestQueue requestQueue =
Volley.newRequestQueue(getApplicationContext());
        StringRequest stringRequest = new
StringRequest(Request.Method.POST, url, new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                Log.i("NORMAL VOLLEY", response);
                Log.d(TAG, response);
            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                Log.e("ERROR VOLLEY", error.toString());
                Toast.makeText(getApplicationContext(), "Problema
de Conexion", Toast.LENGTH_LONG).show();
            }
        }) {
            @Override
            protected Map<String, String> getParams() {
                return paramWebService.getMapParam();
            }
        };
        requestQueue.add(stringRequest);

    } else {
        Toast toast = Toast.makeText(context, "Dispositivo
Desconectado", Toast.LENGTH_SHORT);
        toast.show();
    }
}

```

CÓDIGO FUENTE INTERFAZ DE PROGRAMACIÓN DE APLICACIONES (API) PARA GESTIÓN DE ENVIOS DE SMS EN LENGUAJE DE PROGRAMACIÓN PHP

```
1 <?php
2 include '../@dbcon.php';
3 $minsWait = 1;
4 //REVISAR SI EXISTEN ALARMAS
5 $sql = "SELECT";
6 $sql .= " listhost.IDHost AS IDHost";
7 $sql .= ",listhost.IP AS IP";
8 $sql .= ",listhost.HostName AS HostName";
9 $sql .= ",listhost.Nombre AS Nombre";
10 $sql .= ",listhost.Descrip AS Descrip";
11 $sql .= ",listhost.AlertCon AS AlertCon";
12 $sql .= ",listhost.AlertDes AS AlertDes";
13 $sql .= ",listhost.AlertLag AS AlertLag";
14 $sql .= ",listhost.AlertUsr AS AlertUsr";
15 $sql .= ",listhost.FHCon AS FHCon";
16 $sql .= ",listhost.FHDes AS FHDes";
17 $sql .= ",listhost.FHLag AS FHLag";
18 $sql .= ",listhost.FHUsr AS FHUsr";
19 $sql .= " FROM listhost ";
20 $sql .= " WHERE AlertUsr=1 OR AlertLag=1 OR AlertDes=1 OR AlertCon=1";
21 $sql .= " ORDER BY listhost.IDOrden ASC";
22
23
24 $contSMS = 0;
25 $jsonsms = "sms":[';
26 $rsAlert = mysqli_query( $con, $sql );
27 while ( $rowAlert = mysqli_fetch_assoc( $rsAlert ) ) {
28     $idHost = $rowAlert[ 'IDHost' ];
29     $ip = $rowAlert[ 'IP' ];
30     $hostName = trim( utf8enc( $rowAlert[ 'HostName' ] ) );
31
32     if ( $hostName == "" ) {
33         $hostName = trim( utf8enc( $rowAlert[ 'Nombre' ] ) ) . " " . trim( utf8enc( $rowAlert[ 'Descrip' ] ) );
34     } else {
35         $hostName .= " " . trim( utf8enc( $rowAlert[ 'Nombre' ] ) ) . " " . trim( utf8enc( $rowAlert[ 'Descrip' ] ) );
36     }
37     $sqlUsr = " SELECT ";
38     $sqlUsr .= " usr_list.ID AS idusr ";
39     $sqlUsr .= " ,usr_list.Alias AS alias ";
40     $sqlUsr .= " ,usr_list.cel_alert AS cel ";
41     $sqlUsr .= " FROM usr_ip,usr_list ";
42     $sqlUsr .= " WHERE ";
43     $sqlUsr .= " usr_ip.IDUsr=usr_list.ID ";
44     $sqlUsr .= " AND usr_ip.IDHost=$idHost ";
45     $sqlUsr .= " AND usr_ip.ActiveSMS=1 ";
46     $rsUsr = mysqli_query( $con, $sqlUsr );
```

```

47 ▼ while ( $rowUsr = mysqli_fetch_assoc( $rsUsr ) ) {
48     $cel = $rowUsr[ 'cel' ];
49     $alias = $rowUsr[ 'alias' ];
50     $idUsr = $rowUsr[ 'idusr' ];
51     $alertCon = $rowAlert[ 'AlertCon' ];
52     $alertDes = $rowAlert[ 'AlertDes' ];
53     $alertLag = $rowAlert[ 'AlertLag' ];
54     $alertUsr = $rowAlert[ 'AlertUsr' ];
55     $fhCon = $rowAlert[ 'FHCon' ];
56     $fhDes = $rowAlert[ 'FHDes' ];
57     $fhLag = $rowAlert[ 'FHLag' ];
58     $fhUsr = $rowAlert[ 'FHUsr' ];
59     $msg = Msg( $ip, $hostName, $alertCon, $alertDes, $alertLag, $alertUsr, $fhCon, $fhDes, $fhLag, $fhUsr );
60     $sqlSMS = " SELECT ";
61     $sqlSMS .= " sms_reg.ID AS ID ";
62     $sqlSMS .= ", sms_reg.FHRead AS FHRead ";
63     $sqlSMS .= ", sms_reg.FHConf AS FHConf ";
64     $sqlSMS .= " FROM sms_reg ";
65     $sqlSMS .= " WHERE ";
66     $sqlSMS .= " sms_reg.IDHost=$idHost ";
67     $sqlSMS .= " AND sms_reg.IDUsr=$idUsr ";
68     $sqlSMS .= " AND sms_reg.IP='$ip' ";
69     $sqlSMS .= " AND sms_reg.AlertCon=$alertCon ";
70     $sqlSMS .= " AND sms_reg.AlertDes=$alertDes ";
71     $sqlSMS .= " AND sms_reg.AlertLag=$alertLag ";
72     $sqlSMS .= " AND sms_reg.AlertUsr=$alertUsr ";
73     $sqlSMS .= " ORDER BY sms_reg.FHRead DESC LIMIT 0,1";
74     //echo $sqlSMS;
75     $idSMS = 0;
76     $contRegs = 0;
77     $send = true;
78     $add = true;
79     $fhAct = date( 'Y-m-d' ) . ' ' . date( 'H:i:s' );
80     $rsSMS = mysqli_query( $con, $sqlSMS );
81 ▼ while ( $rowSMS = mysqli_fetch_assoc( $rsSMS ) ) {
82     $contRegs++;
83     $add = false;
84     $idSMS = $rowSMS[ 'ID' ];
85     $fhConf = $rowSMS[ 'FHConf' ];
86 ▼ if ( $fhConf == '' ) {
87     $send = true;
88 ▼ } else {
89     $datetime1 = date_create( $fhConf );
90     $datetime2 = date_create( $fhAct );
91     $interval = $datetime1->diff( $datetime2 );
92     //echo $interval->format( "%H:%I:%S" );
93     $difMins = $interval->format( "%i" );
94 ▼ if ( $difMins < $minsWait ) {
95     $send = false;
96     $difMins = $interval->format( "%I:%S" );
97     $sqlUpd = "UPDATE sms_reg SET FHDiff='$difMins' WHERE ID=$idSMS ";
98     $rsUpd = mysqli_query( $con, $sqlUpd );
99
100 ▼ } else {
101     $send = true;
102     $add = true;
103 }
104 }
105 }
106

```

```

107 ▼ if ( $add ) {
108     //ID
109     $sqlID = "SELECT sms_reg.ID AS ID FROM sms_reg ORDER BY ID DESC LIMIT 0,1";
110     $rsID = mysql_query( $con, $sqlID );
111     $rowID = mysql_fetch_assoc( $rsID );
112     $idSMS = $rowID[ 'ID' ] + 1;
113     //ADD REG
114     $sqlAdd = " INSERT INTO sms_reg ";
115     $sqlAdd .= "(ID,IDHost,IDUsr,IP,HostName,Msg,Cel";
116     $sqlAdd .= ",AlertCon,AlertDes,AlertLag,AlertUsr";
117     $sqlAdd .= ",FHRead) ";
118     $sqlAdd .= "VALUES";
119     $sqlAdd .= "($idSMS,$idHost,$idUsr,'$ip','$hostName','$msg','$cel";
120     $sqlAdd .= ",$alertCon,$alertDes,$alertLag,$alertUsr";
121     $sqlAdd .= ",'$fhAct') ";
122     $rsAdd = mysql_query( $con, $sqlAdd );
123
124 }
125 ▼ if ( $send && $idSMS != 0 ) {
126     //-----
127 ▼ if ( $contSMS > 0 ) {
128     $jonsms .= ',';
129     }
130     $contSMS++;
131     //-----
132 ▼ $jonsms .= '{';
133     $jonsms .= '"idsms":"' . $idSMS . '"';
134     $jonsms .= ', "ip":"' . $ip . '"';
135     $jonsms .= ', "hostName":"' . $hostName . '"';
136     $jonsms .= ', "msg":"' . $msg . '"';
137     $jonsms .= ', "cel":"' . $cel . '"';
138     $jonsms .= ', "alias":"' . $alias . '"';
139     $jonsms .= '}';
140 }
141 }
142 }
143 $jonsms .= ']';
144 echo '{' . $jonsms . '}';
145 ▼ function Msg( $ip, $hostName, $alertCon, $alertDes, $alertLag, $alertUsr, $fhCon, $fhDes, $fhLag, $fhUsr ) {
146     $msg = $ip . ' - ' . $hostName;
147 ▼ if ( $alertCon == 1 ) {
148 ▼ if ( $msg != '' ) {
149     $msg .= ' - ';
150     }
151     $msg .= $fhCon . ' Conexion Detectada ';
152 }
153
154 ▼ if ( $alertDes == 1 ) {
155 ▼ if ( $msg != '' ) {
156     $msg .= ' - ';
157     }
158     $msg .= $fhDes . ' Desconexion Detectada ';
159 }
160 ▼ if ( $alertLag == 1 ) {
161 ▼ if ( $msg != '' ) {
162     $msg .= ' - ';
163     }
164     $msg .= $fhLag . ' Latencia Detectada ';
165 }
166 ▼ if ( $alertUsr == 1 ) {
167 ▼ if ( $msg != '' ) {
168     $msg .= ' - ';
169     }
170     $msg .= $fhUsr . ' PC Sin Usuario Autenticado ';
171 }
172 return $msg;
173 }
174 ?>

```

CÓDIGO FUENTE SOFTWARE INTERNO DE RETRANSMISIÓN SERIAL PLACA ARDUINO

```
USB_SERIAL$
#include<SoftwareSerial.h>
SoftwareSerial myserial(8,9);//RX,TX pin 8 receptor y pin 9 Transmisor
void setup() {
  Serial.begin(9600);
  myserial.begin(9600);
}

void loop() {
  if(myserial.available()){
    Serial.write(myserial.read());
  }
  if(Serial.available()){
    myserial.write(Serial.read());
  }
}
```

CÓDIGO FUENTE PARA ENVIO DE SMS USANDO EL MÓDULO SIM800L

```
Private Sub Btn_SendMsg_Click(sender As Object, e As EventArgs) Handles Btn_SendMsg.Click
    Txt_Read.Text = ""

    Dim str As String

    If SPort.IsOpen Then

        str = "AT+CMGF=1"
        SPort.Write(str & vbCrLf)
        Threading.Thread.Sleep(500)
        str = "AT+CSCS=""GSM""
        SPort.Write(str & vbCrLf)
        Threading.Thread.Sleep(500)
        str = "AT+CMGS="" & Txt_Numero.Text & """"
        SPort.Write(str & vbCrLf)
        Threading.Thread.Sleep(500)
        str = Txt_Mensaje.Text
        SPort.Write(str & vbCrLf)
        Threading.Thread.Sleep(500)
        SPort.Write(Chr(26))

    End If
End Sub
```