

Universidad de Cienfuegos “Carlos Rafael Rodríguez”
Facultad de Ingeniería
Carrera de Ingeniería Informática



**“Sistema de Gestión Básica para la Red DOCSIS de Telecable
Internacional de Cayo Santa María”**

Trabajo de diploma para optar por el título de Ingeniería en Informática

Autor:

Miguel Angel Cortés Moya

Tutores:

**Lic. Yaima Toledo Guerra. Universidad de Cienfuegos
Ing. Alejandro León Rodríguez. Universidad de Cienfuegos**

**Cienfuegos, Cuba
Curso 2012-2013**

DECLARACIÓN DE AUTORÍA

Yo, Miguel Angel Cortés Moya declaro que soy el único autor del trabajo de diploma titulado: "Sistema de Gestión Básica para la Red DOCSIS de Telecable Internacional de cayo Santa María" y autorizo al Centro TELECABLE internacional del cayo Santa María, Villa Clara, para que hagan el uso que estimen pertinente del presente trabajo de diploma.

Para que así conste firmo la presente a los 19 días del mes de junio del 2012

Firma Autor

Miguel Angel Cortés Moya

Los abajo firmantes certificamos que el presente trabajo ha sido revisado según acuerdo de la dirección de nuestro centro y el mismo cumple los requisitos que debe tener un trabajo de esta envergadura referente a la temática señalada.

Firma Tutor

Lic. Yaima Toledo

Firma Tutor

Ing. Alejandro León

Firma ICT

Firma Vicedecano

AGRADECIMIENTOS

Ante todo a mis padres por su apoyo en cada momento de mi vida y por hacerme la persona que soy.

A mis tutores por su empeño, dedicación.

A todos los profesores que de una forma u otra influyeron en mi formación como profesional.

A mis amistades por su ayuda y por todos los momentos que hemos pasado juntos.

A todos mis amigos del aula y en especial a los más allegados a mí.

En general quisiera agradecerles a todos por haber cumplido mi sueño.

RESUMEN

En la cabecera de Telecable se cuenta con el equipamiento necesario para brindar los servicios DOCSIS, pero debido al alto costo del hardware, la inversión en software de gestión es prácticamente inviable, las actividades de gestión de la red exigen del chequeo y configuración de una variedad de parámetros en tiempo real, por lo que se hace necesario la elaboración de un software como alternativa de solución, para eliminar las deficiencias en la gestión de la información de los dispositivos que integran la red, este ha contribuido a la reducción del tiempo de atención a fallas, facilidad de configuración, monitoreo constante y en tiempo real de la red, en el cayo Santa María y con un costo insignificante comparado con los software propietarios existentes. En este trabajo se propone un Sistema de Gestión nombrado "DATA_Cable" que permite influir proactivamente en la calidad de los servicios prestados.

ABSTRACT

Telecable's headend in Key Santa Maria has the elemental equipment to deliver DOCSIS services, but due to hardware's high cost, software investment is very difficult. Management activities demand checking and configuring a wide range of parameters in real-time, which makes the creation of a software an alternate solution, to eliminate deficiencies in information management on the devices conforming the network. This contributes to save time in fail attendance, configuration activities, and real time monitoring in Telecables's network in Key Santa Maria. This work proposes a system named DATA_Cable which permits to influence in a proactive way in the quality of services offered.

ÍNDICE

INTRODUCCIÓN	10
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.	13
1.1 Introducción.	13
1.2 Redes Híbridas de Fibra óptica y Cable Coaxial (HFC).....	13
1.2.1 Arquitectura de una red HFC.....	14
1.2.2 Transmisión de Datos en Redes de CATV/HFC.	15
1.3 Especificación DOCSIS.	17
1.3.1 Características y Fundamentos general.....	20
1.3.2 Operación Básica del Protocolo.....	22
1.4 Gestión de redes DOCSIS.....	23
1.4.1 Protocolo Simple de Administración de Red (SNMP).	24
1.5 Análisis de los Software Existentes.	26
1.6 Flujo actual de los procesos y análisis de la ejecución.....	27
1.7 Metodología de Desarrollo de Software y Lenguaje de Modelado.	27
1.8 Uso de Lenguaje de Programación y Sistema Gestor de Base de Datos.....	31
1.8.1 Java	31
1.8.2 PostgreSQL.	32
1.8.2.1 SQL.....	34
1.9 Herramientas utilizadas.	34
1.10 Conclusiones del Capítulo.....	36
CAPÍTULO 2: PLANIFICACIÓN Y DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.	38
2.1 Introducción.	38
2.2 Pila del Producto.	38
2.3 Requerimientos no funcionales del sistema.	44
2.4 Planeación de los Sprint.	47
2.4.1 Listado de los Sprint.....	48
2.4.2- Técnica de Estimación de un Sprint.....	49
2.4.3 Descripción de los Sprint.	49
2.5 Conclusiones del Capítulo.....	62
CAPÍTULO 3: CONTRUCCIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.....	63
3.1 Introducción.	63
3.2 Historias Técnicas.....	63
3.3 Principios de Diseño.	64
3.3.1 Estándares en la Interfaz de Aplicación.....	64
3.3.2 Tratamiento de Errores.	65

3.3.3 Concepción General de la Ayuda.	65
3.4 Beneficios Tangibles e Intangibles.	66
3.4.1 Análisis de los Costos y Beneficios.	66
3.5 Validación de la Solución Propuesta.	66
3.6 Conclusiones del Capítulo.	69
CONCLUSIONES	70
RECOMENDACIONES.....	71
REFERENCIAS BILIOGRÁFICAS.	72
BIBLIOGRAFÍA.....	74
ANEXOS.....	76
Anexo A1: Tareas del Sprint 1.....	76
Anexo A2: Tareas del Sprint 2.....	76
Anexo A3: Tareas del Sprint 3.....	77
Anexo A4: Tareas del Sprint 4.....	77
Anexo A5: Tareas del Sprint 5.....	78
Anexo A6: Tareas del Sprint 6.....	78

ÍNDICE DE TABLAS

Tabla 1: Velocidades para los canales de bajada de 6MHz.	21
Tabla 2: Pila del Producto	44
Tabla 3: Listado de los Sprint	49
Tabla 4: Estimación del Sprint 1	50
Tabla 5: Estimación del Sprint 2	52
Tabla 6: Estimación del Sprint 3	54
Tabla 7: Estimación del Sprint 4	56
Tabla 8: Estimación del Sprint 5	58
Tabla 9: Estimación del Sprint 6	60

ÍNDICE DE FIGURAS

Figura 1: Arquitectura de una Red HFC.	14
Figura 2: Arquitectura de una red HFC con DOCSIS.	18
Figura 3: Componentes básicos de una Red DOCSIS.	22
Figura 4 Relación Gestor-Agente.....	25
Figura 5: Diagrama de Casos de Usos.....	63
Figura 6: Modelo Lógico de Datos.	64
Figura 7: Comparación del tiempo antes y después del sistema.....	67
Figura 8: Comparación del tiempo antes y después del sistema.....	68
Figura 9: Estadísticas de Muestras Relacionadas.....	68
Figura 10: Prueba de Muestras Relacionadas.....	69

INTRODUCCIÓN

La necesidad para que el público vea un cuadro de televisión sin interferencia, dio lugar a la creación de la industria cablegráfica de la televisión, en los Estados Unidos a finales de la década de 1940, cuando la forma más práctica para brindar transmisiones de televisión regulares fue el uso del espectro electromagnético por aire. Prescindir de sus problemas y limitaciones se hacía a veces muy difícil, pues las señales podían ser severamente debilitadas o bloqueadas por obstáculos naturales o artificiales. Surge así el concepto de *Community Antenna (CA)* o antena comunitaria, cuya idea básica consistía en utilizar un único punto de recepción y una red de conductores para distribuir las señales, y que se desarrolló paralelamente en varias ciudades pero se acreditó a Ed Parsons de Astoria, Oregón, Estados Unidos, como el primero en ponerlo en práctica en 1948.[1]

Como resultado de un proceso de unificación de métodos y conceptos, y en un ambiente altamente competitivo entre proveedores de servicios de telecomunicaciones, surge DOCSIS, estándar no comercial que define los requisitos de la interfaz de comunicaciones y operaciones para el soporte de transmisión de datos sobre sistemas de televisión por cable. Muchos operadores de estos servicios lo emplean para proporcionar acceso a Internet sobre una infraestructura HFC (red híbrida de fibra óptica y coaxial) existente.

La última versión de DOCSIS, DOCSIS 3.0, que data del 2006, aparece implementada en la gran mayoría del equipamiento actual para estos servicios, permitiendo la puesta en marcha de redes más complejas, tanto en tamaño como en funcionamiento, pudiendo alcanzar tasas de transmisión de más de 100 Mbps para un solo usuario.

En Cayo Santa María, provincia de Villa Clara, la empresa Telecable Internacional tiene en operación una Red DOCSIS pero para aprovechar al máximo sus beneficios se hace necesario un sistema informático que viabilice las labores de Administración del equipamiento y de la operación de la red, conocido en este campo como Sistema de Gestión de Red. Los altos precios

que impone el mercado a productos de este tipo debido a su complejidad, la escasez de quienes desarrollan este tipo de software específicamente, más la imposibilidad política de acceder a algunos de estos mercados, hace que sea muy difícil y costoso para el país adquirir una aplicación de este tipo para una red como la desplegada en dicho lugar en cuanto a extensión y cantidad de abonados.

Teniendo en cuenta todo lo anterior se identifica como **problema a resolver**: ¿Cómo implementar un Sistema de Gestión Básica para la Red DOCSIS de Telecable Internacional Cayo Santa María, capaz de administrar los dispositivos de la red para su correcto funcionamiento?

Se define como **idea a defender** el desarrollo de un Sistema Informático que sea capaz de gestionar y mejorar el funcionamiento de los dispositivos que se encargan del funcionamiento de la red, que proporcionará facilidad en la configuración y el monitoreo constante y en tiempo real de la red.

Se identifica como **objeto de estudio** la gestión de los procesos que llevan consigo la administración de los dispositivos de este tipo de red.

De este modo se deriva como **campo de acción** la gestión de los procesos que lleva consigo el funcionamiento del CMTS.

En la solución al problema que se ha planteado queda definido como **objetivo general** elaborar un Sistema de Gestión Básica para la Red DOCSIS de Telecable Internacional Cayo Santa María, que responda a las necesidades del usuario.

Del cual se desprenden los siguientes **objetivos específicos**:

- Investigar el estándar DOCSIS 3.0 y la descripción de sus principales características.
- Analizar el protocolo SNMP de administración de redes y sus implementaciones por los diferentes lenguajes de programación.
- Implementar el Sistema de Gestión Básica.
- Probar los resultados en un entorno real.

- Validar el sistema.

Para cumplir estos objetivos se realizaron las siguientes **tareas**:

- Recopilación de la información necesaria para el análisis de los temas relacionados con el estudio de las redes DOCSIS 3.0
- Conformación de una interfaz usuario-sistema.
- Evaluación de la efectividad de la propuesta en el entorno de red.
- Entrevistas a los usuarios del sistema para confirmar si el sistema cumple con los requisitos establecidos.

Su utilidad está dada por su **aporte práctico**, este conduce al desarrollo de un sistema informático para facilitar la gestión del CMTS y la red DOCSIS en general, elevando la organización y la rapidez con que fallas o dificultades pueden ser atendidas en la cabecera de Telecable Internacional Cayo Santa María, Villa Clara.

Estructura del documento de tesis: La tesis se estructura en 3 capítulos, conclusiones, recomendaciones, referencias bibliográficas, bibliografía y anexos.

Capítulo 1. Fundamentación Teórica.

En el capítulo se abordan aspectos relacionados con la fundamentación teórica del tema. Se describen los sistemas existentes vinculados al campo de acción y se hace un análisis crítico de los mismos. Se tratan también las tendencias, tecnologías, metodologías y lenguajes utilizados.

Capítulo 2. Planificación y Descripción de la Solución Propuesta.

En el capítulo se aborda acerca de la metodología Scrum y las características propias de dicha metodología.

Capítulo 3. Construcción y Validación de la Solución Propuesta.

En el capítulo se plantean y detallan una serie de diagramas que ayudan y guían en la implementación del modelo de sistema, así como el estudio de factibilidad y la validación de la propuesta.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

1.1 Introducción.

En este capítulo se abordan aspectos relacionados con la fundamentación teórica del tema, se describen los principales conceptos asociados al estándar Docsis. Se describen los sistemas existentes vinculados al campo de acción y se hace un análisis crítico de los mismos. Además se fundamentan las tendencias, tecnologías y metodologías, así como el sistema gestor de Bases de Datos y las herramientas utilizadas.

1.2 Redes Híbridas de Fibra óptica y Cable Coaxial (HFC).

A finales de los años 80 empezaron a funcionar las redes de televisión por cable con la arquitectura denominada HFC, como mejora de las redes que sólo funcionaban con cable coaxial (antiguas Redes de Televisión por Cable "CATV"). Con esta nueva arquitectura se buscaba resolver los problemas de gestión y mantenimiento que generaban las redes CATV. Esta tiene dos niveles jerárquicos, el principal formado por un tendido de fibra que distribuye la señal desde el centro emisor (cabecera) hasta cada zona del área que atiende, donde existe un nodo que se encarga de convertir la señal óptica en eléctrica para su distribución final en cable coaxial a los abonados, abarcando de 500 a 2000 abonados típicamente. Debido a la atenuación de las señales a través del cable coaxial, se hace necesario colocar amplificadores en el trayecto para mantener el nivel de señal. Estas nuevas redes, debido a la menor distancia a cubrir con cable coaxial, el número de amplificadores máximo en cascada disminuye y se establece alrededor de 5 (en muchas redes no hay más de 2 ó 3), obteniéndose una mejora en la calidad de la señal y sencillez de mantenimiento.

Un elemento importante en la implantación de redes HFC es la posibilidad de enviar una señal analógica en fibra sin necesidad de convertirla en una señal digital. Otro gran paso para estas redes fue la facilidad de la utilización de la red para el tráfico upstream (señales que pueden viajar en sentido contrario),

permitiendo las labores de monitoreo y servicios tales como el de internet y Voz sobre Protocolo de Internet (VoIP). Según esquemas adoptados, la señal de televisión utiliza el rango de frecuencias altas entre 85-500 MHz para el sentido descendente de las señales analógicas de TV, 500-750/860 MHz para los servicios de televisión digital, internet y VoIP, aunque esta planificación de frecuencias no es obligatoria ni necesaria. Para el sentido ascendente (upstream) se utilizan frecuencias entre 5 y 65 MHz, principalmente para el retorno de los datos provenientes de los clientes.[2]

De esta forma se logra emplear la red para los nuevos servicios sin prescindir de los servicios tradicionales o previos, consolidado una red de servicios integrados.

1.2.1 Arquitectura de una red HFC.

La figura1 muestra un ejemplo de arquitectura HFC. Se tienen un anillo de fibra óptica que se encarga de distribuir la señal desde el centro emisor ubicado en la cabecera regional a una serie de concentradores (cabeceras locales). Cada uno de estos concentradores a su vez reenvían la señal también por fibra óptica a una serie de convertidores opto-eléctricos y desde éstos mediante cable coaxial a los abonados.[2]

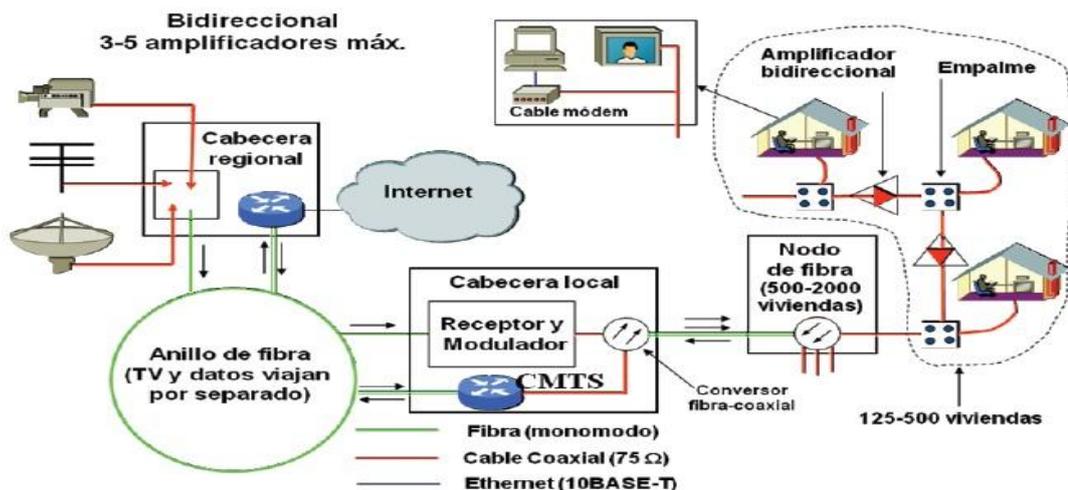


Figura 1: Arquitectura de una Red HFC.

Donde cada componente de la arquitectura se explican a continuación:

- **Cabecera de red (cabecera regional):** en ésta se recogen, comprimen, multiplexan las distintas señales ofrecidas por el operador de cable. Una cabecera de red puede atender millones de hogares.
- **Nodos de Red (cabeceras locales):** la cabecera de red alimenta a varios nodos de red enlazados a ella, típicamente y por razones de seguridad, vía anillos síncronos auto restaurables. Los nodos de red incluyen por lo general amplificadores y divisores ópticos. Un nodo de red puede atender hasta unos 80.000 hogares.
- **Nodos de fibra (nodos locales):** son los elementos intermedios de red, que amplifican y distribuyen la señal. Uno de estos nodos puede atender hasta 40.000 hogares.
- **Terminal de red óptica (TRO), conversor fibra-coaxial:** realiza la conversión electro-óptica, procesa la señal ascendente para su transmisión al nodo local.

Los usuarios se enlazan al terminal de red óptica a través de una red de tipo coaxial con topología árbol-rama. Esta red incluye un cable coaxial con múltiples ramificaciones, cada una de las cuales brinda servicios a través de nuevas ramificaciones. Un terminal de red óptica puede atender a unos 500 abonados, aunque normalmente se establece un número inferior, con fin de facilitar la implementación del canal ascendente.

1.2.2 Transmisión de Datos en Redes de CATV/HFC.

Emplear el cable coaxial como medio de transmisión, para aprovechar la red desplegada e implementar una red de datos, requiere separar en frecuencia las diferentes señales que lo utilizarán. Así, se separa una banda para la comunicación de los datos en sentido contrario, o como se le conoce, upstream o canal de subida, de la banda descendente o downstream, donde se ubican los canales convencionales de televisión, y los de datos en sentido descendente. Las señales de subida se transmiten en frecuencias por debajo de los 65 MHz, para no solaparse con los otros servicios. Esta banda es especialmente sensible al ruido, y se necesitan eficientes equipos para lograr la comunicación.[3]

Los nuevos amplificadores, que tienen que amplificar en los dos sentidos, separan las señales de bajada y subida en frecuencia, la amplifican, la vuelven a componer y la entregan, y se conocen como amplificadores bidireccionales.

Durante los años noventa, la industria del cable desarrolló un gran número de esquemas y normas para soportar la transmisión de datos en redes CATV/HFC, por lo que varios estándares emergieron en calidad de competidores: [4]

- **IEEE 802.14:** En 1994 este grupo de trabajo desarrolló una capa de MAC (Control de Acceso al Medio) para soportar ATM IP (*Modo de Transferencia Asíncronico*) sobre redes HFC. El canal de subida era TDMA (*Multiplexación por División de Tiempo*) con un tamaño de ranura de 8 bytes.
- **Multimedia Cable Network System's (MCNS) DOCSIS:** En respuesta a la competencia de DSL (*Línea de Suscripción Digita*), varios Operadores de Servicios Múltiples (MSO) crearon el MCNS (*Sistema de Red Multimedia de Cable*) para definir un sistema estándar para proveer servicios y datos sobre infraestructuras CATV. En 1997 lanzaron la versión 1.0 de DOCSIS. El canal de subida utilizaba TDMA con un tamaño de ranura configurable, y fue rápidamente aprobado por la industria.
- **DAVIC/DVB:** La organización europea DAVIC (*Consejo de Audio y Video Digital*) produjo el DAVIC 1.2 y el muy similar DVB-RCC (*Difusión de Video Digital con Canal de Retorno para Cable*), ambos estándares para cable módems, que definieron las PHY (Capas Físicas) y MAC para comunicaciones bidireccionales sobre redes CATV/HFC. Posteriormente la industria europea se movió hacia EuroDOCSIS, lógicamente por motivos económicos.

De todos estos triunfó DOCSIS, que fue patrocinado por los grandes fabricantes de equipos de comunicaciones como Cisco y Broadcom, y otros poderosos como Intel y Motorola, y que paulatinamente se consolidaría como el estándar para la transmisión de datos IP sobre infraestructuras HFC.

1.3 Especificación DOCSIS.

DOCSIS® es una marca registrada de CableLabs® (*Cable Television Laboratories, Inc.*), consorcio de investigación y desarrollo sin ánimos de lucro integrado por *Arris, BigBand, Broadcom, Cisco Systems, Harmonic, Intel, Motorola* y *Texas Instruments*; y marcó la entrada de los MSOs en el mercado de la *broadband*¹. [5]

Es además un estándar no comercial que dicta los requerimientos de una interfaz de comunicaciones para soportar el tráfico de datos IP en sistemas de cable, permitiendo transferencias de datos de alta velocidad en redes HFC, y con el objetivo adicional de crear interoperabilidad entre equipos desarrollados por diferentes fabricantes. [5]

Se describe en un grupo de especificaciones y reportes técnicos que exponen cómo deben operar los dispositivos y sus interfaces en capa Física (PHY), Control de Acceso al Medio (MAC) e Internet (IP). La versión europea de DOCSIS se llama EuroDOCSIS, y la principal diferencia es que en Europa los canales de cable tienen un ancho de banda de 8 MHz, mientras que en Norte América es de 6 MHz.

La primera versión de la especificación (DOCSIS 1.0) vio la luz en marzo de 1997, y tuvo una gran revisión (DOCSIS 1.1) en abril de 1999, que fundamentalmente adicionó algún soporte de QoS (*Calidad de Servicio*). La versión 2.0 (DOCSIS 2.0) se lanzó en diciembre de 2001, con una mejora en las velocidades de subida. Posteriormente en agosto de 2006, aparece DOCSIS 3.0, para proveer mayores velocidades tanto en subida como en bajada, así como dar soporte a IP versión 6. [6]

En cuanto a la capa que presta los datos, sobre la plataforma bidireccional, DOCSIS añade dos equipos cruciales para este fin:

- **CMTS:** Sistema de Terminación de Cable Modem, es el dispositivo que se encarga de enviar los datos en sentido descendente modulados por el canal de televisión elegido al efecto y también reciben de los cable módems

¹ Servicios de Banda Ancha

de los usuarios los datos que éstos envían a través del canal ascendente asignado. El CMTS se ubica generalmente en el centro emisor o cabecera de la red, desde allí se conecta al resto de la red (red de transporte) y a Internet por alguna tecnología WAN.

- **Cable Modem:** Dispositivo dispuesto en el extremo del usuario, se encarga de sintonizar el canal de televisión elegido para los datos y extraer los que le corresponden. Los datos que serán enviados a otra estación, ya sea en la misma red (cabecera local) o diferente, serán transmitidos por el canal ascendente. Las formas para conectar el cable modem al computador del usuario es por medio de la interfaz Ethernet de 10/100 Mbps que es una interfaz de alta velocidad y bajo costo; también se puede conectar el cable módem desde un puerto USB.

De esta manera, la arquitectura de una red DOCSIS se representaría como se muestra en la figura 2. El CMTS básicamente conecta Internet y la red interna del proveedor (*Back Office Network*) con la red HFC, reenviando paquetes entre esos dos grandes dominios y entre los canales de subida y bajada dentro la propia red de cable.[7]

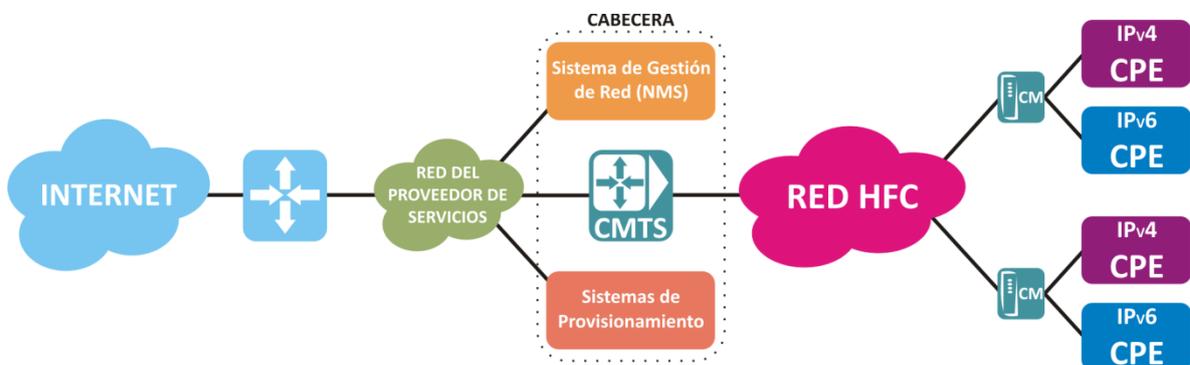


Figura 2: Arquitectura de una red HFC con DOCSIS.

Tanto el canal ascendente como el descendente son canales compartidos, pero en redes HFC sólo son compartidos por los usuarios de una misma zona, no entre zonas diferentes. Esto permite que en función de la densidad de usuarios del servicio de datos de cada zona, el operador puede adoptar estrategias como las siguientes:[3]

- En zonas con densidad baja se pueden agrupar varias en un mismo canal compartido, con lo que a efectos de datos se comporta todo como una misma zona.
- Al tener una densidad alta se pueden asignar varios canales ascendentes y/o descendentes a una misma zona, dando como resultado la división de ésta en dos partes. En el caso de los canales descendentes también se puede jugar con el ancho de banda.

El CMTS es el dispositivo más complejo de la red, ya que manipula toda la información. El CMTS realiza funciones de:

- Enrutamiento
- MODEM
- Mantenimiento y operación de la red

Las operaciones de mantenimiento se podrían descomponer a su vez, para una mejor comprensión en:

- Funciones de Monitoreo: se encargan del chequeo de los parámetros de operación incluyendo la detección y atención a fallas.
- Funciones de Configuración: que se encarga de la manipulación de los parámetros que determinan el funcionamiento de los dispositivos de la red.

La forma en que se realizan tanto las funciones de monitoreo como las de configuración, son las que caracterizan al proceso de Administración de Red, conociéndose a la plataforma o aplicación cuya función tributa directamente a este proceso como Sistema de Gestión de Red.

El Sistema de Gestión de la Red se compone de programas de aplicación que intervienen en la administración y monitoreo de los recursos, siendo los principales:

- Gestor SNMP: permite al operador configurar y monitorear el equipamiento, típicamente los CMs y el CMTS.

- Servidor Syslog: recolecta mensajes acerca de la operación de los dispositivos.

Pero para dar soporte a la red DOCSIS, además del Sistema de Gestión de Red, debe emplearse según especificaciones, otro sistema fundamental: el Sistema de Aprovisionamiento (*Provisioning System*), que no es objeto de esta investigación pero que cabe mencionarse, y se conforma de un conjunto de aplicaciones que intervienen en la puesta en operación de los CMs. Este grupo de aplicaciones incluye principalmente:

- Servidores DHCP: proveen información de configuración inicial como la dirección IP de los CMs y otras direcciones.
- Servidor de Archivos de Configuración (TFTP): permite que los CMs descarguen sus archivos de configuración cuando arrancan.
- Servidor de Protocolo de Tiempo RFC868: proporciona la referencia de tiempo para la red.

1.3.1 Características y Fundamentos general.

DOCSIS funciona principalmente entre las capas 1 y 2 del modelo OSI de la ISO, conocidas también como Capa Física (PHY) y Capa de Control de Acceso al Medio (MAC). En términos simples, las redes DOCSIS son redes IP sobre HFC, donde los paquetes IP son codificados y transportados como señales de televisión digital que coexisten con otros servicios también transportados por la red.

El espectro total es dividido en una parte para el *upstream* desde 5 MHz hasta 65 MHz (en algunos sistemas hasta 85 MHz, conocida como banda extendida de *upstream*), y la parte para el *downstream* a partir de los 108 MHz hasta la capacidad de la red. Así DOCSIS especifica un camino asimétrico de datos con flujos de bajada y subida en dos frecuencias separadas. Las portadoras de bajada y subida proveen dos canales compartidos para todos los CMs.

Downstream: Desde el CMTS hacia los CMs

La banda de *downstream* se comparte con los demás servicios que brinda la red. El ancho de banda para cada canal descendente puede ser de 6 MHz ó de 8 MHz para cumplir con las normas de los canales de difusión de televisión en Norteamérica y Europa. Los formatos de modulación que pueden ser empleados son: 64-QAM y 256-QAM, y en la tabla 1 se muestran las diferentes tasas que pueden alcanzarse. [7]

Tipo de Modulación	Tasa de Modulación	Bits por símbolo	Velocidad de Transferencia
64-QAM	5.056941 Msym/s	6	30.34 Mbps
256-QAM	5.360537 Msym/s	8	42.88 Mbps

Tabla 1: Velocidades para los canales de bajada de 6MHz.

Los paquetes enviados por el canal de bajada se separan en tramas MPEG de 188 bytes, con 4 bytes de encabezado y 184 bytes de carga útil. Aun cuando a todos los CMs llegan todas las tramas, normalmente se configuran para recibir solamente las que están direccionadas a su dirección MAC o a la dirección de difusión.

En el sentido descendente, DOCSIS emplea un sistema *primero-que-llega primero-que-se-atende* para el reenvío de paquetes, así los paquetes que arriban de Internet se reenvían inmediatamente como paquetes DOCSIS y se modulan en un canal *downstream* de RF para la transmisión.

Upstream: Desde los CMs hacia el CMTS

Las características de esta banda (ruido y distorsión) hacen que se necesiten mecanismos robustos, como modulaciones eficientes, para la transmisión de datos. A diferencia de la banda de *downstream*, en la que sólo hay un transmisor, en esta banda hay muchos transmisores y un solo receptor, por lo que la utilización del medio de transmisión deber ser cuidadosamente organizada y repartida entre todos los CMs. Por todo esto se conforman los CMTS con más puertos de subida que de bajada, permitiendo segmentar más la red en el sentido ascendente y lograr mejores niveles de señal.

En la banda de *upstream*, DOCSIS emplea dos esquemas básicos de acceso al medio: modo FDMA/TDMA, descrito como TDMA, y modo FDMA/TDMA/S-CDMA descrito como modo S-CDMA, permitiendo seis tasas de modulación y múltiples formatos para ésta. Con FDMA se divide la banda en múltiples canales de frecuencias. TDMA responde a la naturaleza de ráfaga de las transmisiones de subida y permite compartir un canal de RF de subida entre varios cable módems a través de la asignación dinámica de ranuras de tiempo. S-CDMA da la posibilidad de que varios CMs transmitan simultáneamente en el mismo canal de RF y sobre la misma ranura de tiempo TDMA, separados por diferentes códigos ortogonales.[7]

Los tipos de modulación establecidos para los moduladores/demoduladores de *upstream* son: QPSK, 8-QAM, 16-QAM, 32-QAM, 64-QAM y 128-QAM. El CMTS reserva el ancho de banda de *upstream* basado en peticiones de los CMs y las políticas de calidad de servicio. El canal de subida se divide en ranuras de tiempo multiplexadas, llamadas “mini-slots”, de una duración de 6.25 µs, y la oportunidad para transmitir en ellas es administrada por el CMTS, que las identifica mediante referencias de tiempo y que son utilizadas por los CMs según mecanismos de contención. La influencia de la distancia en el retardo implica una compensación de tiempo (*offset*) en las referencias de los CMs para que identifiquen con suficiente precisión la posición de los mini-slots.[7]

1.3.2 Operación Básica del Protocolo.

La configuración elemental de una red DOCSIS se muestra en la figura 3.

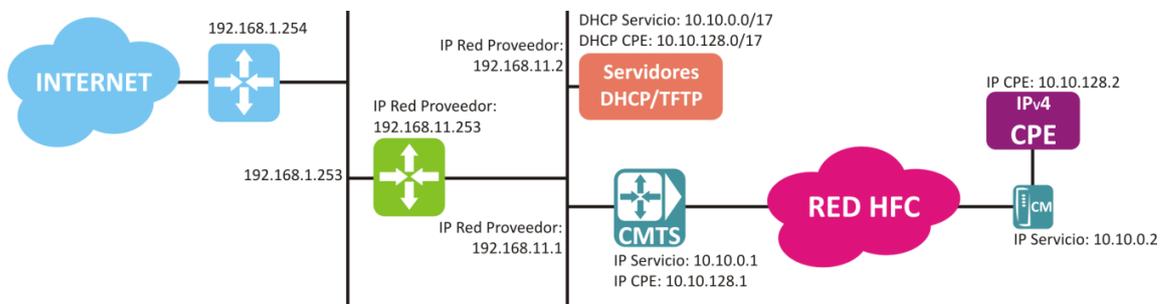


Figura 3: Componentes básicos de una Red DOCSIS.

Básicamente, para que funcione la red DOCSIS se debe tener en funcionamiento el CMTS, un servidor DHCP y uno TFTP, para que los CMs puedan registrarse. Normalmente se emplea un DHCP de dos niveles o rangos: uno para la operación de los cable módems a nivel de servicio, y otro para los equipos de premisas del cliente (CPE).[8]

Cuando un cable módem se conecta a la red coaxial, y se enciende, comienza a rastrear un canal descendente buscando en toda la banda una transmisión 64-QAM ó 256-QAM, que contenga información válida acerca de un canal de subida, para poder establecer una comunicación bidireccional. Si efectivamente lo encuentra y lo retiene, comienza lo que se conoce como *Ranging*. El *ranging* es un proceso que consta de dos etapas, la primera es el *Ranging* Inicial, en la que se configura el menor nivel de potencia que garantiza la comunicación. Después pasa a un nuevo estado conocido como *Ranging* de Mantenimiento (*Station Maintenance Ranging*), en el que el cable módem recibirá instrucciones desde el CMTS para hacer ajustes en cuanto a frecuencias de transmisión, amplitud, *timing offset* y pre-equalización. El *Ranging* de Mantenimiento de Estaciones ocurrirá al menos una vez cada 30 segundos para cada cable módem, realizando ajustes de forma continua y para que el CMTS conozca que los módems se encuentran en línea.[8]

De esta manera, el CMTS periódicamente envía tramas de gestión y otra clase de mensajes denominados mensajes MAP, para que los CMs puedan identificar futuras ranuras TDMA para el *upstream*. Inmediatamente al contar con ancho de banda para transmitir, el CM obtiene su IP y otras direcciones importantes mediante DHCP y descarga el archivo de configuración del servidor TFTP, con los parámetros que necesita para el acceso a la red, velocidades, QoS y otras configuraciones. Si las pruebas de validez son satisfactorias, se concluye el proceso con el Registro del CM en la red y se le asigna un Identificador de Servicio (SID). Una vez registrado el CM, se permite a los suscriptores transmitir tráfico IP.[8]

1.4 Gestión de redes DOCSIS.

DOCSIS, al operar en la capa IP, define además un estándar IP para la gestión de sus dispositivos, conocido como SNMP y ampliamente difundido como herramienta de administración de redes, y que es empleado por el Gestor SNMP dentro del Sistema de Gestión de Red para el intercambio de información de gestión con el equipamiento de la red HFC.

1.4.1 Protocolo Simple de Administración de Red (SNMP).

SNMP, o Protocolo Simple de Gestión de Redes, define un protocolo sencillo que permite obtener y modificar valores de variables específicas de sistemas, definidas en su arquitectura como objetos.[9]

SNMP emplea un modelo *gestor/agente*, integrado por un Gestor (NMS), un Sistema Gestionado, una Base de Datos de Información de Gestión y un Protocolo de Red.[10]

SNMP es en este caso el protocolo de red, y se utiliza en dos servicios fundamentales:[10]

Monitoreo: Encuesta las redes para valorar su funcionamiento, uso general y capacidad, determinación de problemas y análisis gráficos y estadísticos. Adicionalmente, incorpora componentes para la notificación de alertas a los administradores.

Configuración: Permite cambiar parámetros dinámicamente dentro del funcionamiento de la red, responder de manera rápida ante problemas y realizar pruebas en tiempo real de cómo un cambio afecta la red.

Estas rutinas son muy importantes en el mantenimiento de las redes. En el caso de las redes HFC, se dedicarían, por ejemplo, al chequeo del estado de los elementos de la red y a la configuración de la modulación y frecuencia de los canales de transmisión o recepción.

El Gestor es básicamente un software, que generalmente incluye una interfaz gráfica de usuario, que corre uno o más procesos que se comunican con los agentes en la red, y que posee toda la inteligencia de gestión para simplificar el

impacto de los agentes en los dispositivos gestionados. Los procesos de gestión recolectan, analizan, procesan y modifican los valores de los objetos en los agentes.

Un agente, por otro lado, se implementa en el dispositivo o sistema gestionado, y mantiene una base de datos local de los objetos que describen su estado, historia, e inciden en su funcionamiento. Operan enviando notificaciones sobre eventos y respondiendo peticiones de los gestores. Su interacción con el gestor se muestra gráficamente en la figura 4, que en una red DOCSIS se hablaría del software de gestión como el NMS y el CMTS, por ejemplo, como el agente.

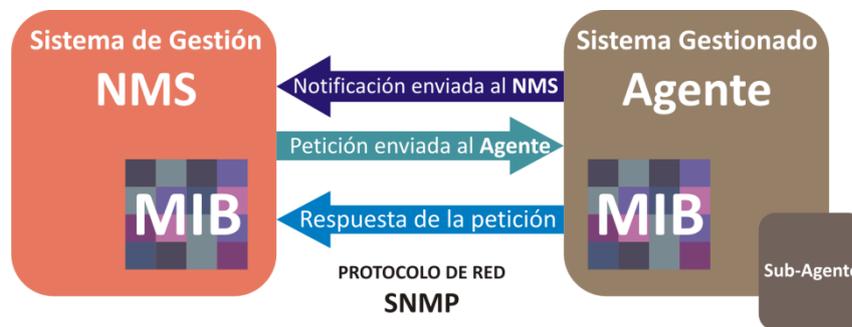


Figura 4 Relación Gestor-Agente

La Base de Información de Gestión (MIB) define un conjunto de objetos que pueden ser monitoreados y administrados con SNMP. Se asocia tanto con el sistema que gestiona como con el gestionado, para que ambos conozcan los objetos a los que se tiene acceso. Una MIB tiene una estructura de árbol que comienza en una raíz y con las ramas que organizan los objetos o variables en categorías lógicas, representando los objetos finales como las hojas de las ramas. Una MIB define un nombre textual para los objetos y explica su función y estructura.[10]

Son precisamente los valores de estos objetos, y su modificación, los que rigen el funcionamiento de la red, y por tanto operar con ellos es la tarea de un Sistema de Gestión de Red.

El empleo de este protocolo en lenguajes de programación se hace generalmente a través de APIs que implementan sus mecanismos, y que permiten la comunicación con los agentes.

1.5 Análisis de los Software Existentes.

En lo concerniente a la gestión de equipamiento DOCSIS, específicamente para Sistemas de Terminación de Cable Módems, aparecen software propietario de algunos de los fabricantes y sólo producen el software.

Entre los principales software´s comerciales que incluyen la gestión DOCSIS se pueden encontrar:

- Cablesight
- Cisco Network Registrar (CNR)
- Intraway Broadband Service Delivery Platform
- Sigma Systems Service Management Platform
- Weird Solutions Broadband Provisioner
- Kathrein CMS

Todos se caracterizan por sus altos precios de adquisición y poca información divulgada, y constituyen muchas veces el renglón fundamental de utilidades de estas compañías, por encima de la fabricación del equipamiento.

La compañía de Telecable Internacional en Cayo Santa María, Villa Clara, actualmente cuenta con un software llamado CMTS-Manager que cuenta con pocas funcionalidades. El mismo fue creado con el lenguaje Delphi y debido a las exigencias que hoy la empresa tiene, muchas no pueden ser resueltas por el mismo. Tampoco brinda documentación, por lo que es muy complejo para un programador poder modificarlo e incrementarle algunas funciones de nuevo interés para la compañía. Además este sistema se apoya en el gestor de base de datos Microsoft Access, el mismo no es muy confiable desde el punto de vista de seguridad ya que no cuenta con restricciones en el instante de manejar las tablas, y esto es un problema, ya que en las mismas se almacenan datos que son de gran valor para la compañía Telecable Internacional.

Debido a esta situación antes planteada es que se propone la realización de un sistema que además de incluir las funcionalidades del anterior, que son escasas, pueda informatizar los demás procesos que son llevados a cabo dentro de la compañía de Telecable Internacional para poder gestionar el CMTS. Una de las grandes ventajas con las que contará el software es que al ser creado con el lenguaje java, será un sistema multiplataforma en cuanto a sistema operativo. El sistema quedará montado sobre el gestor de base de datos de PostgreSQL, haciendo esto más confiable la seguridad y la integridad de los datos.

1.6 Flujo actual de los procesos y análisis de la ejecución.

El proceso de gestión del CMTS comienza con el monitoreo de la red, esto se realiza supervisando el funcionamiento de los dispositivos que conforman la red, luego se toman las decisiones pertinentes para darle solución a los distintos problemas que se pudieran presentar.

A partir del monitoreo es que comienza el proceso de configuración de la red apoyándose en el CMTS que es quien permite modificar los parámetros de funcionamiento de las interfaces, las notificaciones de error, las alarmas, y los cable modem. Actualmente el software no es capaz de realizar estos procesos y se hacen de forma manual mediante la consola, haciendo muy lento el proceso de monitoreo y configuración en la compañía Telecable Internacional.

1.7 Metodología de Desarrollo de Software y Lenguaje de Modelado.

Las Metodologías de desarrollo de software en ingeniería de software son un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información.[11]

También tienen como objetivo presentar un conjunto de técnicas tradicionales y modernas de modelado de sistemas, que permitan desarrollar software de calidad, incluyendo heurísticas de construcción y criterios de comparación de modelos de sistemas.[11]

Metodologías Tradicionales

Las metodologías tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto software. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. Sin embargo las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos óptimos cuando se trabaja en un entorno donde los requisitos no pueden predecirse o bien pueden variar.[11]

Entre las metodologías tradicionales o pesadas se puede citar:

- RUP (Rational Unified Procces)
- MSF (Microsoft Solution Framework)
- Win-Win Spiral Model
- Iconix

Metodologías Ágiles

Los procesos ágiles son una buena elección cuando se trabaja con requerimientos desconocidos o variables. Si no existen requisitos invariables, se carecería de un diseño estable y de seguir un proceso totalmente planificado, que no vaya a variar ni en tiempo, ni en dinero. En estas situaciones, sería más eficiente la utilización de un proceso adaptativa, ya que facilita la generación rápida de prototipos y de versiones previas a la entrega final, lo cual agrada al cliente.[11]

Las metodologías ágiles proporcionan una serie de pautas y principios junto a técnicas establecidas que puede que no curen todos los males, pero harán entrega del proyecto menos complicado y más satisfactorio tanto para los clientes como para los equipos de entrega.[11]

Entre las metodologías ágiles más destacadas hasta el momento se pueden nombrar:

- XP (Extreme Programming)
- Scrum
- Crystal Clear
- DSDM (Dynamic Systems Development Method)
- FDD (Feature Driven Development)

Luego del análisis de las diferentes metodologías se elige Scrum como metodología de desarrollo por las características siguientes:

- Es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Los roles principales en Scrum son el Scrum Master, que mantiene los procesos y trabaja de forma similar al director de proyecto, el ProductOwner, que representa a los stakeholders (clientes externos o internos), y el equipo que incluye a los desarrolladores.[12]
- Durante cada sprint, un período entre 15 y 30 días (la magnitud es definida por el equipo), el equipo crea un incremento de software potencialmente entregable (utilizable). El conjunto de características que forma parte de cada sprint viene de la pila del producto, que es un conjunto de requisitos de alto nivel priorizados que definen el trabajo a realizar. Los elementos de la pila del producto que forman parte del sprint se determinan durante la reunión de Planificación del Sprint. Durante esta reunión, el dueño del producto identifica los elementos de la pila del producto que quiere ver completados y los hace del conocimiento del equipo. Entonces, el equipo determina la cantidad de ese trabajo que puede comprometerse a completar durante el siguiente sprint. Durante el sprint, nadie puede cambiar la pila del sprint, lo que significa que los requisitos están congelados durante el sprint.
- Scrum permite la creación de equipos auto organizados impulsando la localización de todos los miembros del equipo, y la comunicación verbal entre todos los miembros y disciplinas involucradas en el proyecto.[12]
- Un principio clave de Scrum es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan (a menudo llamado requirement schurn), y que los desafíos

impredecibles no pueden ser fácilmente enfrentados de una forma predictiva y planificada. Por lo tanto, Scrum adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido, y centrándose en maximizar la capacidad del equipo de entregar rápidamente y responder a requisitos emergentes.[12]

- Existen varias implementaciones de sistemas para gestionar el proceso de Scrum, que van desde notas amarillas "post-it" y pizarras hasta paquetes de software. Una de las mayores ventajas de Scrum es que es muy fácil de aprender, y requiere muy poco esfuerzo para comenzarse a utilizar.[12]

UML

El Lenguaje de Modelado Unificado (UML – Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un producto de software que responde a un enfoque orientado a objetos. Este lenguaje fue creado por un grupo de estudiosos de la Ingeniería de Software formado por: Ivar Jacobson, Grady Booch y James Rumbaugh en el año 1995. Desde entonces, se ha convertido en el estándar internacional para definir organizar y visualizar los elementos que configuran la arquitectura de una aplicación orientada a objetos.

UML no es un lenguaje de programación sino un lenguaje de propósito general para el modelado orientado a objetos y también puede considerarse como un lenguaje de modelado visual que permite una abstracción del sistema y sus componentes.[13]

Entre sus objetivos fundamentales se encuentran:[13]

- Ser tan simple como sea posible, pero manteniendo la capacidad de modelar toda la gama de sistemas que se necesita construir.
- Necesita ser lo suficientemente expresivo para manejar todos los conceptos que se originan en un sistema moderno, tales como la concurrencia y distribución, así como también los mecanismos de la ingeniería de software, como son el encapsulamiento y los componentes.

- Debe ser un lenguaje universal, como cualquier lenguaje de propósito general.
- Imponer un estándar mundial.

1.8 Uso de Lenguaje de Programación y Sistema Gestor de Base de Datos

Para la confección del sistema se utilizó el lenguaje de programación Java y el gestor de bases de datos PostgreSQL.

1.8.1 Java

La empresa Sun Microsystems decidió introducirse en el mercado de la electrónica de consumo para desarrollar programas para pequeños dispositivos electrónicos (tostadoras, microondas, TV interactiva, etc.), para ello “Sun” creó una filial denominada FirstPerson Inc. James Gosling de “FirstPersonInc”, a partir del C++ crea un nuevo lenguaje de programación (1991), que llamó Oak para solucionar el gran problema de “programación” en la electrónica de consumo:[14]

- En la electrónica de consumo, los chips electrónicos correspondientes cambian muy rápidamente: una pequeña diferencia en el precio de un chip, por ejemplo.
- Al utilizar el lenguaje “C++”, si cambiamos el chip, es necesario rehacer todos los programas, para adaptarlos al nuevo dispositivo electrónico.
- Un programa escrito en “OAK” no necesita rehacerse para compilarse de nuevo, al cambiar el chip.

En definitiva, si programamos en “Oak” no es necesario cambiar el programa, si varía el chip donde está implementado. En 1995 tras un cambio de nombre y mejoras en su diseño se presentó el “Java” en sociedad.

Características

Entre sus características se puede mencionar:[15]

- El “Java” es un lenguaje de programación completo orientado a objetos.

- El “Java” se diseñó a partir del “C++” con el propósito de:
 - Ser reducido
 - Sencillo
 - Rápido
 - Eficiente
 - Transportable (en diversas plataformas y sistemas operativos)
- Con “Java” se puede crear dos tipos de programas:
 - Aplicaciones Completas
 - Applets

¿Qué lo hace distinto de los demás lenguajes?

Una de las características más importantes es que los programas “ejecutables”, creados por el compilador de Java, son independientes de la arquitectura. Se ejecutan indistintamente en una gran variedad de equipos con diferentes microprocesadores y sistemas operativos.[15]

- De momento, es público. Puede conseguirse un JDK (Java Developer's Kit) o Kit de desarrollo de aplicaciones Java gratis. No se sabe si en un futuro seguirá siéndolo.
- Permite escribir *Applets* (pequeños programas que se insertan en una página HTML) y se ejecutan en el ordenador local.
- Se pueden escribir aplicaciones para intrarredes, aplicaciones cliente/servidor, aplicaciones distribuidas en redes locales y en Internet.
- Es fácil de aprender y está bien estructurado.
- Las aplicaciones son fiables. Puede controlarse su seguridad frente al acceso a recursos del sistema y es capaz de gestionar permisos y criptografía. También, según “Sun”, la seguridad frente a virus a través de redes locales e Internet está garantizada. Aunque al igual que ha ocurrido con otras tecnologías y aplicaciones, se han descubierto, y posteriormente subsanado, “agujeros” en la seguridad de Java.

1.8.2 PostgreSQL.

El sistema gestor de Bases de datos relacionales Orientadas a Objetos conocido como PostgreSQL está derivado del paquete Postgres escrito en Berkeley. Con cerca de una década de desarrollo PostgreSQL es el gestor de bases de datos de código abierto más avanzado hoy en día, ofreciendo control de concurrencia multi-versión, soportando casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, y tipos y funciones definidas por el usuario), contando también con un amplio conjunto de enlaces con lenguajes de programación (incluyendo C, C++, Java, Perl, tcl y Python).[16]

¿Qué es Postgres?

Los sistemas de mantenimiento de Bases de datos relacionales tradicionales (DBMS) soportan un modelo de datos que consisten en una colección de relaciones con nombre, que contienen atributos de un tipo específico. En los sistemas comerciales actuales, los tipos posibles incluyen, numéricos de punto flotante, enteros, cadenas de caracteres, cantidades monetarias y fechas. Está generalmente reconocido que éste modelo será inadecuado para las aplicaciones futuras de procesado de datos. El modelo relacional sustituyó modelos previos en parte por su “simplicidad espartana”. Sin embargo, como se ha mencionado, esta simplicidad también hace muy difícil la implementación de ciertas aplicaciones. Postgres ofrece una potencia adicional sustancial al incorporar los siguientes cuatro conceptos adicionales básicos en una vía en la que los usuarios pueden extender fácilmente el sistema:[16]

- Clases
- Herencia
- Tipos
- Funciones

Otras características aportan potencia y flexibilidad adicional:

- Restricciones (Constraints)
- Disparadores (triggers)
- Reglas (rules)
- Integridad transaccional

Estas características colocan a Postgres en la categoría de las Bases de Datos identificadas como *objeto-relacionales*. Nótese que estas son diferentes de las referidas como *orientas a objetos*, que en general no son bien aprovechables para soportar lenguajes de Bases de Datos relacionales tradicionales. Postgres tiene algunas características que son propias del mundo de las bases de datos orientadas a objetos. De hecho, algunas Bases de Datos comerciales han incorporado recientemente características en la que Postgres fue pionera.[16]

1.8.2.1 SQL

Como en el caso de los más modernos lenguajes relacionales, SQL está basado en el cálculo relacional de tuplas. Como resultado, toda consulta formulada utilizando el cálculo relacional de tuplas (o su equivalente, el álgebra relacional) se puede formular también utilizando SQL. Sin embargo, hay capacidades que van más allá del cálculo o del álgebra relacional. Aquí se tiene una lista de algunas características proporcionadas por SQL que no forman parte del álgebra relacional y del cálculo relacional:[17]

- Comandos para inserción, borrado o modificación de datos.
- Capacidades aritméticas: En SQL es posible incluir operaciones aritméticas así como comparaciones, por ejemplo $A < B + 3$. Nótese que no + ni otros operadores aritméticos aparecían en el álgebra relacional ni en el cálculo relacional.
- Asignación y comandos de impresión: es posible imprimir una relación construida por una consulta y asignar una relación calculada a un nombre de relación.
- Funciones agregadas: Operaciones tales como *promedio (average)*, *suma (sum)*, *máximo (max)*, etc. Se pueden aplicar a las columnas de una relación para obtener una cantidad única.

1.9 Herramientas utilizadas.

NetBeans

Es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.[18]

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos.[18]

La Plataforma NetBeans es una base modular y extensible usada como una estructura de integración para crear aplicaciones de escritorio grandes. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones.[18]

La plataforma ofrece servicios comunes a las aplicaciones de escritorio, permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación. Entre las características de la plataforma están:[18]

- Administración de las interfaces de usuario (ej. menús y barras de herramientas)
- Administración de las configuraciones del usuario
- Administración del almacenamiento (guardando y cargando cualquier tipo de dato)
- Administración de ventanas
- Framework basado en asistentes (diálogos paso a paso)

Visual Paradigm

Visual Paradigm es una herramienta CASE: Ingeniería de Software Asistida por Computación. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación.[20]

Esta herramienta permite aumentar la calidad del software, a través de la mejora de la productividad en el desarrollo y mantenimiento del software. Aumenta el conocimiento informático de una empresa ayudando así a la búsqueda de soluciones para los requisitos. También permite la reutilización del software, portabilidad y estandarización de la documentación, además del uso de las distintas metodologías propias de la Ingeniería del Software.[20]

ERStudio

ER/Studio es una herramienta que modela los datos, se usa para el diseño y la construcción lógica y física de base de datos. Su ambiente es de gran alcance, de varios niveles del diseño. ER/Studio fue diseñada para hacer más fácil de entender el estado actual de los datos de la empresa, ayuda a organizaciones para tomar decisiones en cómo resolver embotellamientos de los datos, elimina redundancia y alcanza en última instancia usos de más alta calidad que entreguen datos más eficientes y exactos a la empresa. Las capacidades de diseño que contiene, ayudan a crear un diseño lógico que puede transformarse en cualquier número de diseños físicos. Como resultado, se puede mantener un diseño lógico normalizado mientras se des-normalizan los diseños físicos para su desempeño. ER/Studio mantiene ligaduras entre todos los niveles de su diseño por lo tanto puede mezclar cambios en cualquier dirección entre ellos. ER/Studio revisa la normalización y la compilación con la sintaxis de la plataforma de la base de datos.[21]

1.10 Conclusiones del Capítulo.

En el presente capítulo, han sido aclarados los conceptos asociados al dominio del problema, realizándose además una investigación acerca de soluciones

dadas hasta el momento como alternativas de informatización a los procesos de la gestión de redes DOCSIS.

Para ello se utilizó la metodología de desarrollo de software Scrum, se siguió el paradigma de la programación orientada a objetos y la arquitectura cliente-servidor, apoyándose en el lenguaje de modelado UML para la realización del diseño. Para la implementación se hizo uso de las herramientas NetBeans como IDE de programación y Visual Paradigm para la creación de los modelos. Como gestor de Bases de Datos se utilizó PostgreSQL y como lenguaje de programación Java.

CAPÍTULO 2: PLANIFICACIÓN Y DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.

2.1 Introducción.

En este capítulo tomando como metodología Scrum se definen la pila del producto donde van a estar incluidos todos los requisitos funcionales del sistema, la pila de los Sprint y la planeación de cada uno, apoyado en las técnicas de estimación de un sprint. También se definen las tareas para cada Sprint y los requisitos no funcionales del sistema.

2.2 Pila del Producto.

Son los requisitos del sistema. Se parte de la visión del resultado que se desea obtener; y evoluciona durante el desarrollo. Es el inventario de características que desea el propietario. El responsable de la pila del producto es una única persona y se le denomina: propietario del producto. Además la pila del producto describe cómo va a quedar esa historia y como podrías probarla.[22]

Pila del Producto:

ID	Nombre	Importancia	Estimación Inicial	Como Probarlo	Observaciones
1	Autenticar Usuario	30	7	Entrar usuario y contraseña, en caso que sean correctos el sistema mostrará la ventana principal. En caso que sea incorrecto el sistema mostrará un	Se requiere un algoritmo de encriptación para que las contraseñas sean guardadas de forma

				mensaje de alerta.	segura.
2	Gestionar Usuario (Insertar, Modificar y Eliminar el usuarios)	40	5	Entrar, abrir ventana de gestionar usuario, escoger la opción, Insertar, Modificar o Eliminar usuario. En el caso de insertar, se llenan los campos y el sistema mostrará un mensaje diciendo que fue insertado correctamente. En el caso de modificar se modifican los campos escogidos y en el caso de eliminar se escoge el usuario que se eliminará, luego verificar en la lista de usuarios.	Se necesitan realizar consultas a la base de datos.
3	Cambiar Contraseña	30	3	Entrar, abrir ventana de cambiar contraseña, se introduce el usuario, la contraseña anterior y la nueva dos veces, en caso que los datos introducidos sean correctos se mostrará un mensaje satisfactorio, en caso contrario se mostrará un mensaje de alerta	Se necesitan hacer consultas a la base de datos y utilizar un algoritmo de encriptación para guardar con seguridad la contraseña del usuario en la base de datos.

				diciendo que algo está erróneo en los datos introducidos.	
4	Modificar parámetros de funcionamiento del canal Downstream.	8	30	Entrar, abrir ventana de Modificar parámetros de funcionamiento del canal Downstream y modificar los parámetros escogidos, el sistema mostrará un mensaje de parámetros modificados.	Se necesita realizar una petición al CMTS, este mostrará los parámetros a modificar del canal.
5	Modificar parámetros de funcionamiento del canal Upstream.	7	30	Entrar, abrir ventana de Modificar parámetros de funcionamiento del canal Upstream y modificar los parámetros escogidos, el sistema mostrará un mensaje de parámetros modificados.	Se necesita realizar una petición al CMTS, este mostrará los parámetros a modificar del canal.
6	Establecer parámetros de funcionamiento del CMTS.	5	30	Entrar, abrir ventana de Establecer parámetros de funcionamiento del CMTS, cuando son introducidos los valores el sistema mostrará un mensaje de parámetros establecidos correctamente.	Se establecen los parámetros de, intervalo de inserción, intervalo de mensajes UCD e intervalo de sincronización.
7	Monitoreo de las interfaces	5	30	Entrar, abrir ventana de Monitoreo de las	Ver que estén habilitadas o

	del CMTS.			interfaces del CMTS y comprobar que estén en correcto funcionamiento.	deshabilitadas.
8	Gestionar las interfaces del CMTS. (Habilitar/ Deshabilitar)	3	40	Entrar, abrir ventana de gestionar las interfaces del CMTS, habilitar o deshabilitar la interfaz, ir a ventana de monitoreo y ver si se habilitó o deshabilitó según el caso.	Realizar petición al CMTS que me muestre la lista de interfaces.
9	Mostrar rendimiento del CMTS.	2	30	Entrar, abrir ventana de rendimiento del CMTS, el sistema debe mostrar el uso del CPU, memoria y temperatura del CMTS.	Preferiblemente mostrarlo en gráficas.
10	Obtener la información disponible de todos los CM conectados al CMTS.	5	20	Entrar, abrir ventana de Información de los CM conectados al CMTS, ver la información de todos los CM conectados.	Mostrar los campos de los CM y escoger cuales se quiere mostrar.
11	Mostrar información del estado de la red.	5	30	Entrar, abrir ventana sobre el estado de la red, seleccionar guardar estado de la red y el sistema debe emitir un reporte con todos los parámetros de los	Tomar los datos de los módems conectados en ese instante.

				módems en ese instante en la red.	
12	Monitorear la utilización de los recursos de comunicaciones de los canales de Downstream.	5	30	Entrar, abrir ventana de utilización de recursos de los canales de Downstream, el sistema debe guardar cada un tiempo 'x' la utilización de los recursos de comunicación de los canales de Downstream.	Hay que hacer una salva cada un tiempo "x" automáticamente de la utilización de los recursos.
13	Monitorear la utilización de los recursos de comunicaciones de los canales de Upstream.	5	20	Entrar, abrir ventana de utilización de recursos de los canales de Upstream, el sistema guarda cada un tiempo 'x' la utilización de los recursos. Ir a Visualizar Historial de recursos de comunicación de los canales de upstream y comprobar que se ha ido guardando la información.	Hay que hacer una salva cada un tiempo "x" automáticamente de la utilización de los recursos.
14	Visualizar Historial de recursos de comunicación de los canales de	5	30	Entrar, abrir ventana de Visualizar historial de recursos de comunicación de los canales de Upstream, ver que se ha ido guardando la	Agregarle el campo fecha en la base de datos. Hacerle un filtro para verlo por días, semanas o mes.

	upstream.			información del uso de los recursos.	
15	Monitorear los indicadores de ruido en los canales de Upstream.	5	40	Entrar, abrir ventana de monitoreo de los indicadores de ruido, se debe ver mediante una gráfica los indicadores de ruido en los canales de Upstream.	Mostrarlo en una gráfica.
16	Confeccionar reportes temporales de los indicadores de ruido en los canales Upstream.	5	30	Entrar, abrir ventana de monitorear los indicadores de ruido de los canales de Upstream, pulse el botón de crear reporte de ruidos el sistema se encargue de crear un reporte de los indicadores de ruido.	El reporte debe dar la opción de imprimirlo.
17	Almacenar notificaciones emitidas por el CMTS.	5	20	Entrar, abrir ventana de notificaciones del CMTS, y comprobar que se está realizando una salva de las notificaciones emitidas por el CMTS.	Guardar las traps emitidas por el CMTS
18	Detectar condiciones de error.	5	30	Entrar, abrir ventana configurar condiciones de error e introducir las condiciones permitidas para ciertos parámetros de error, al cumplirse	Configurar un conjunto de condiciones, y mostrar los mensajes de alerta en tiempo

				alguna de las condiciones el sistema mostrará un mensaje de alerta.	real.
--	--	--	--	---	-------

Tabla 2: Pila del Producto

2.3 Requerimientos no funcionales del sistema.

Los requerimientos no funcionales del sistema propuesto son los siguientes:

Requerimiento de Usabilidad:

Los factores principales que deben considerarse al hablar de usabilidad son la facilidad de comprensión, la capacidad de uso y la satisfacción con la que las personas son capaces de hacer sus tareas gracias al uso del producto con el que están trabajando, factores que fueron creados pensados por el usuario.

Facilidad de comprensión:

- Las funcionalidades del sistema deben ser de un fácil acceso.
- El programa debe permitir que el usuario corrija la respuesta antes de que ésta sea aceptada por el programa.
- El programa debe aceptar como correctas, respuestas parciales cuando es adecuado.
- Permitir al usuario pedir ayuda en cualquier momento.
- Permitir al usuario usar un menú para seleccionar partes del programa.
- El tamaño y color de la letra debe permitir leer con facilidad.
- Los despliegues de datos deben ser flexibles (por ejemplo, reducidos a escala, colapsando la ventana, moviendo la ventana de lugar para su adecuada lectura).
- La cantidad de colores en la pantalla debe ser adecuada para el tipo de información que contiene.
- Se deben dar a los colores las connotaciones estándares (ejemplo: rojo para parar o peligro).

- Los gráficos deben estar apropiadamente posicionados en la pantalla.
- Los gráficos y efectos visuales ayudan a entender los contenidos.

Capacidad de uso:

- La ayuda para el uso del software se deben hallar rápido y fácilmente.
- El texto de la documentación (impresa o en línea) es claro y legible.
- Deben existir instrucciones para las fallas más comunes.

Interfaz gráfica:

- Debe haber variedad de pantallas.
- El diseño de la interfaz debe evitar la pérdida de tiempo.
- Las opciones se deben localizar rápidamente y ser consistente en la ubicación de las funciones e íconos en la pantalla.

Operabilidad:

- La secuencia de los elementos del menú deben ser lógica.
- El tiempo de carga en la computadora (el tiempo de inicio para la puesta en práctica) debe ser lo suficientemente breve.
- Se debe salir del software de una manera fácil (no hace falta recurrir al uso de muchas pantallas para acceder a la salida del software y la salida del programa debe ser fácil de identificar y realizar).

Requerimiento de Fiabilidad:

- Es la capacidad del producto de software para mantener un nivel especificado de rendimiento cuando es utilizado bajo condiciones específicas.

Recuperación:

- El software debe recuperarse fácilmente después de una caída o falla (puede volverse a abrir el programa sin ningún inconveniente después de una falla).
- Se debe permitir a los usuarios trabajar con el producto de software el tiempo necesario.
- La velocidad de re inicialización debe ser rápida.

Tolerancia a fallas:

- Tener en cuenta ¿Qué tan frecuente ocurre una falla, aún si el usuario lo opera incorrectamente?
- El software debe presentar una explicación y opciones para la solución, cuando ocurre la falla.
- Las opciones de solución de fallas deben resolver el problema.
- La falla en la que no ofrece explicaciones, se resuelve reiniciando la computadora (la falla bloquea la computadora y se arregla al apagar y prender el equipo otra vez).

Requerimiento de Confiabilidad:

- Se define la confiabilidad como la capacidad del producto de software para mantener un nivel de ejecución especificado cuando se usa bajo las condiciones especificadas.
- La confiabilidad tiene incluidos varios atributos entre los que se encuentran: disponibilidad, fiabilidad, seguridad, confidencialidad, integridad y mantenibilidad.
- Todos en su conjunto posibilitan la operación de un software libre de fallos en un entorno determinado.

Requerimientos de Soporte.

- Los servicios de instalación y mantenimiento del sistema deberán realizarse por personal calificado, teniendo en cuenta las configuraciones necesarias para su correcto funcionamiento.
- Las pruebas del sistema se realizarán en el local de cabecera de TELECABLE donde prestan sus servicios los técnicos encargados de la distribución de los paquetes de televisión hacia los hoteles del cayo. Dichas pruebas permitirán evaluar en la práctica la funcionalidad y las ventajas de este nuevo producto.
- El sistema debe propiciar su mejoramiento y la anexión de otras opciones que se le incorporen en un futuro.

Requerimientos Políticos – Culturales.

- El desarrollo del sistema debe estar en correspondencia con la cultura organizacional de la empresa.

Requerimientos Legales.

- La herramienta propuesta responde a los intereses de la Compañía Telecable Internacional perteneciente a la Corporación Cimex.SA.

Requerimientos de software.

- La aplicación debe poderse ejecutar en entornos de Windows y Linux (Multiplataforma).
- Se necesita instalar una máquina virtual de java para poder correr la aplicación, debido a que el lenguaje utilizado fue java.
- La pc del cliente debe estar conectada a la red de datos de la empresa y tener instalado el gestor de base de datos PostgreSQL.

Requerimiento de Hardware.

- Se requiere de una máquina que funcione como servidor de aplicaciones y de base de datos (Pentium DualCore 2.0 GHz ,1 Gb RAM y 40 Gb de disco duro).

Requerimientos de Seguridad.

- La aplicación implementará varios niveles de acceso, por medio de usuarios que desempeñan diferentes roles.
- Garantizará la seguridad en la transmisión de datos utilizando algoritmos de encriptación.

2.4 Planeación de los Sprint.

La planificación de un Sprint es una reunión crítica, probablemente la más importante de Scrum. Una planificación de un Sprint mal ejecutada puede arruinar por completo todo el Sprint. [22]

El propósito de la planificación de los Sprint es proporcionar al equipo suficiente información como para que puedan trabajar en paz y sin interrupciones durante

unas pocas semanas, y para ofrecer al dueño del producto suficiente confianza como para permitirse. [22]

Una planificación de un Sprint está dividida en varias partes:

- Una meta de Sprint.
- Una fecha concreta para la Demo del Sprint.
- Una Pila de Sprint (lista de historias incluidas en el Sprint).
- Historias incluidas en el Sprint.
- Cómo probar cada historia del Sprint.
- Una lista de miembros (y su nivel de dedicación, si no es del 100%).
- Un lugar y momento definidos para el Scrum Diario.
- Historias divididas en tareas.

2.4.1 Listado de los Sprint.

Número de Sprint	Duración (Días)	Participantes	Factor de dedicación
Sprint 1	15	Miguel Angel Cortés Yaima Toledo Alejandro León	0,70
Sprint 2	15	Miguel Angel Cortés Yaima Toledo Alejandro León	0,68
Sprint 3	15	Miguel Angel Cortés Yaima Toledo Alejandro León	0,65
Sprint 4	15	Miguel Angel Cortés Yaima Toledo Alejandro León	0,68
Sprint 5	15	Miguel Angel	0,68

		Cortés Yaima Toledo Alejandro León	
Sprint 6	15	Miguel Angel Cortés Yaima Toledo Alejandro León	0,714

Tabla 3: Listado de los Sprint

2.4.2- Técnica de Estimación de un Sprint.

Existen dos técnicas para la estimación de la velocidad con que se va a trabajar en el proyecto:

1) Ojo de buen cubero.

No requiere de ninguna fórmula, se basa en la apreciación del equipo. El ojo de buen cubero funciona bastante bien para equipos pequeños y sprint cortos.[22]

2) Cálculo de velocidad basado en días-hombre disponibles y factor de dedicación.

La velocidad estimada es una medida de “cantidad de trabajo realizado”, donde cada elemento se evalúa en función de su estimación inicial.[22]

(Velocidad Real)

- $$\text{FACTOR D DEDICACIÓN} = \frac{\text{(Velocidad Real)}}{\text{(Días Hombres-Disponibles)}}$$

- $$\text{VELOCIDAD ESTIMADA} = \text{(Días Hombres-Disponibles)} \times \text{(Factor de dedicación)}$$

La velocidad real es la suma de las estimaciones iniciales que se completaron en el último Sprint.[22]

2.4.3 Descripción de los Sprint.

Sprint 1:

Metas:

Las metas de este Sprint son:

- Permitir a cada Usuario interactuar con el sistema de acuerdo al rol y a los privilegios que tenga asignado.
- Insertar, Modificar y Eliminar Usuarios.
- Cambiar Contraseña de un Usuario.

Pila del Sprint:

- Autenticar Usuario.
- Gestionar Usuario.
- Cambiar Contraseña.

Estimación de Historias del Sprint 1.

Cálculo de la velocidad estimada para el Sprint 1 utilizando la técnica de cálculo de velocidad basado en días-hombres disponibles y factor de dedicación.

Trabajadores	Días-Hombres (disponibles)	Factor de Dedicación
Miguel Angel Cortés	10	0.70
Yaima Toledo	5	0.70
Alejandro León	7	0.70

Tabla 4: Estimación del Sprint 1

Velocidad Estimada = $22 * 0.70$

Velocidad Estimada = 15 (puntos de historia)

Historias incluidas en el Sprint:

1. Autenticar Usuario: 7 puntos de historia inicialmente utilizando la técnica de ojo de buen cubero.
2. Gestionar Usuario: 5 puntos de historia inicialmente utilizando la técnica de ojo de buen cubero.

3. Cambiar Contraseña: 3 puntos de historia inicialmente utilizando la técnica de ojo de buen cubero.

Como probar cada historia del Sprint.

1. Autenticar Usuario:

Introducir el usuario y la contraseña, en caso que ambos sean correctos el sistema mostrará la ventana principal, pero solo con los permisos de este usuario, en caso que se hayan introducido erróneamente los datos, el sistema mostrará un mensaje de error.

2. Gestionar Usuario:

Entrar, abrir ventana de gestionar usuario, escoger la opción, Insertar, Modificar o Eliminar Usuario. En el caso de insertar, se llenan los campos y el sistema mostrará un mensaje diciendo que fue insertado correctamente. En el caso de modificar se modifican los campos escogidos y en el caso de eliminar se escoge el usuario que se eliminará, luego verificar en la lista de usuarios.

3. Cambiar Contraseña:

Entrar, abrir ventana de cambiar contraseña, se introduce el usuario, la contraseña anterior y la nueva dos veces, en caso que los datos introducidos sean correctos se mostrará un mensaje satisfactorio, en caso contrario se mostrará un mensaje de alerta diciendo que algo está mal en los datos introducidos.

Lista de Miembros:

Miguel Angel Cortés: 60% de trabajo en el Sprint.

Yaima Toledo: 10% de trabajo en el Sprint.

Alejandro León: 30% de trabajo en el Sprint.

Lugar y momento definido para el Scrum diario:

Departamento de Informática: 10:00 am

Historias divididas en tareas:

Las tareas del Sprint 1 se pueden ver en el Anexo A1.

Sprint 2:

Metas:

Las metas de este Sprint son:

- Cambiar los parámetros de funcionamiento del canal Downstream.
- Cambiar los parámetros de funcionamiento del canal Upstream.

Pila del Sprint:

- Modificar los parámetros de funcionamiento del canal Downstream.
- Modificar los parámetros de funcionamiento del canal Upstream.

Estimación de Historias del Sprint 2.

Cálculo de la velocidad estimada para el Sprint 2 utilizando la técnica de cálculo de velocidad basado en días-hombres disponibles y factor de dedicación.

Factor de dedicación: $15/22=0,68$

Trabajadores	Días-Hombres (disponibles)	Factor de Dedicación
Miguel Angel Cortés	15	0,68
Yaima Toledo	2	0.68
Alejandro León	6	0,68

Tabla 5: Estimación del Sprint 2

Velocidad Estimada = $23 * 0.68$

Velocidad Estimada = 15 (puntos de historia)

Historias incluidas en el Sprint:

1. Modificar parámetros de funcionamiento del canal Downstream: 8 puntos de historia inicialmente utilizando la técnica de ojo de buen cubero.

2. Modificar parámetros de funcionamiento del canal Upstream: 7 puntos de historia inicialmente utilizando la técnica de ojo de buen cubero.

Como probar cada historia del Sprint.

1. Modificar parámetros de funcionamiento del canal Downstream.

Entrar, abrir ventana de Modificar parámetros de funcionamiento del canal Downstream y modificar los parámetros escogidos, el sistema mostrará un mensaje de parámetros modificados.

2. Modificar parámetros de funcionamiento del canal Upstream.

Entrar, abrir ventana de Modificar parámetros de funcionamiento del canal Upstream y modificar los parámetros escogidos, el sistema mostrará un mensaje de parámetros modificados.

Lista de Miembros:

Miguel Angel Cortés: 60% de trabajo en el Sprint.

Yaima Toledo: 10% de trabajo en el Sprint.

Alejandro León: 30% de trabajo en el Sprint.

Lugar y momento definido para el Scrum diario:

Departamento de Informática: 10:00 am

Historias divididas en tareas:

Las tareas del Sprint 2 se pueden ver en el Anexo A2.

Sprint 3:

Metas:

Las metas de este Sprint son:

- Establecer los parámetros de funcionamiento del CMTS como intervalo de inserción, intervalo de mensajes UCD y ancho del canal.
- Monitorear las interfaces del CMTS.

- Habilitar o Deshabilitar las interfaces del CMTS.
- Visualizar parámetros de rendimiento del CMTS como: uso de CPU, memoria y temperatura.

Pila del Sprint:

- Establecer parámetros de funcionamiento del CMTS.
- Monitoreo de las interfaces del CMTS.
- Gestionar las interfaces del CMTS.
- Mostrar rendimiento del CMTS.

Estimación de Historias del Sprint 3.

Cálculo de la velocidad estimada para el Sprint 3 utilizando la técnica de cálculo de velocidad basado en días-hombres disponibles y factor de dedicación.

Factor de dedicación: $15/23=0,65$

Trabajadores	Días-Hombres (disponibles)	Factor de Dedicación
Miguel Angel Cortés	14	0,65
Yaima Toledo	2	0.65
Alejandro León	6	0,65

Tabla 6: Estimación del Sprint 3

Velocidad Estimada = $22 * 0.68$

Velocidad Estimada = 15 (puntos de historia)

Historias incluidas en el Sprint:

1. Modificar parámetros de funcionamiento del canal Downstream.
2. Modificar parámetros de funcionamiento del canal Upstream.

Como probar cada historia del Sprint.

1. Modificar parámetros de funcionamiento del canal Downstream:

Entrar, abrir ventana de Modificar parámetros de funcionamiento del canal Downstream y modificar los parámetros escogidos, el sistema mostrará un mensaje de parámetros modificados.

2. Modificar parámetros de funcionamiento del canal Upstream:

Entrar, abrir ventana de Modificar parámetros de funcionamiento del canal Upstream y modificar los parámetros escogidos, el sistema mostrará un mensaje de parámetros modificados.

Lista de Miembros:

Miguel Angel Cortés: 60% de trabajo en el Sprint.

Yaima Toledo: 10% de trabajo en el Sprint.

Alejandro León: 30% de trabajo en el Sprint.

Lugar y momento definido para el Scrum diario:

Departamento de Informática: 10:00 am

Historias divididas en tareas:

Las tareas del Sprint 3 se pueden ver en el Anexo A3.

Sprint 4:

Metas:

Las metas de este Sprint son:

- Visualizar los datos de todos los CM conectados.
- Visualizar el comportamiento de la red.
- Visualizar el uso de los recursos de comunicaciones del canal Downstream.

Pila del Sprint:

- Obtener la información disponible de todos los CM conectados al CMTS.
- Mostrar información del estado de la red.
- Monitorear la utilización de los recursos de comunicación de los canales de Downstream.

Estimación de Historias del Sprint 4.

Cálculo de la velocidad estimada para el Sprint 4 utilizando la técnica de cálculo de velocidad basado en días-hombres disponibles y factor de dedicación.

Factor de dedicación: $15/22=0,68$

Trabajadores	Días-Hombres (disponibles)	Factor de Dedicación
Miguel Angel Cortés	14	0,68
Yaima Toledo	2	0.68
Alejandro León	6	0,68

Tabla 7: Estimación del Sprint 4

Velocidad Estimada = $22 * 0.68$

Velocidad Estimada = 15 (puntos de historia)

Historias incluidas en el Sprint:

1. Obtener la información disponible de todos los CM conectados al CMTS: 5 puntos de historia inicialmente utilizando la técnica de ojo de buen cubero.
2. Mostrar información del estado de la red: 5 puntos de historia inicialmente utilizando la técnica de ojo de buen cubero.
3. Monitorear la utilización de los recursos de comunicación de los canales de Downstream: 5 puntos de historia inicialmente utilizando la técnica de ojo de buen cubero.

Como probar cada historia del Sprint.

1. Obtener la información disponible de todos los CM conectados al CMTS:

Entrar, abrir ventana de Información de los CM conectados al CMTS, ver la información de todos los CM conectados.

2. Mostrar información del estado de la red:

Entrar, abrir ventana sobre el estado de la red, seleccionar guardar estado de la red y el sistema debe emitir un reporte con todos los parámetros de los módems en ese instante en la red.

3. Monitorear la utilización de los recursos de comunicaciones de los canales de Downstream:

Entrar, abrir ventana de utilización de recursos de los canales de Downstream, el sistema debe guardar cada un tiempo 'x' la utilización de los recursos comunicación de los canales de Downstream.

Lista de Miembros:

Miguel Angel Cortés: 60% de trabajo en el Sprint.

Yaima Toledo: 10% de trabajo en el Sprint.

Alejandro León: 30% de trabajo en el Sprint.

Lugar y momento definido para el Scrum diario:

Departamento de Informática: 10:00 am

Historias divididas en tareas:

Las tareas del Sprint 4 se pueden ver en el Anexo A4.

Sprint 5:

Metas:

Las metas de este Sprint son:

- Visualizar el uso de los recursos de comunicaciones del Upstream.
- Visualizar historial de los recursos de los canales de Upstream.

- Visualizar los indicadores de ruido en los canales de Upstream.

Pila del Sprint:

- Monitorear la utilización de los recursos de comunicaciones de los canales de Upstream.
- Visualizar Historial de recursos de comunicación de los canales de Upstream.
- Monitorear los indicadores de ruido en los canales de Upstream.

Estimación de Historias del Sprint 5.

Cálculo de la velocidad estimada para el Sprint 5 utilizando la técnica de cálculo de velocidad basado en días-hombres disponibles y factor de dedicación.

Factor de dedicación: $15/22=0,68$

Trabajadores	Días-Hombres (disponibles)	Factor de Dedicación
Miguel Angel Cortés	14	0,68
Yaima Toledo	2	0.68
Alejandro León	5	0,68

Tabla 8: Estimación del Sprint 5

Velocidad Estimada = $21 * 0.68$

Velocidad Estimada = 15 (puntos de historia)

Historias incluidas en el Sprint:

1. Monitorear la utilización de los recursos de comunicación de los canales de Upstream: 5 puntos de historia inicialmente utilizando la técnica de ojo de buen cubero.

2. Visualizar Historial de recursos de comunicación de los canales de Upstream: 5 puntos de historia inicialmente utilizando la técnica de ojo de buen cubero.
3. Monitorear los indicadores de ruido en los canales de Upstream: 5 puntos de historia inicialmente utilizando la técnica de ojo de buen cubero.

Como probar cada historia del Sprint.

1. Monitorear la utilización de los recursos de comunicación de los canales de Upstream:

Entrar, abrir ventana de utilización de recursos de los canales de Upstream, el sistema guarda cada un tiempo 'x' la utilización de los recursos. Ir a Visualizar Historial de recursos de comunicación de los canales de upstream y comprobar que se ha ido guardando la información.

2. Visualizar Historial de recursos de comunicación de los canales de upstream:

Entrar, abrir ventana de Visualizar historial de recursos de comunicación de los canales de Upstream, ver que se ha ido guardando la información del uso de los recursos.

3. Monitorear los indicadores de ruido en los canales de Upstream:

Entrar, abrir ventana de monitoreo de los indicadores de ruido, se debe ver mediante una gráfica los indicadores de ruido en los canales de Upstream.

Lista de Miembros:

Miguel Angel Cortés: 60% de trabajo en el Sprint.

Yaima Toledo: 10% de trabajo en el Sprint.

Alejandro León: 30% de trabajo en el Sprint.

Lugar y momento definido para el Scrum diario:

Departamento de Informática: 10:00 am

Historias divididas en tareas:

Las tareas del Sprint 5 se pueden ver en el Anexo A5.

Sprint 6:

Metas:

Las metas de este Sprint son:

- Crear reportes sobre los indicadores de ruido de los canales Upstream.
- Almacenar notificaciones del CMTS.
- Alertar sobre condiciones de error.

Pila del Sprint:

- Confeccionar reportes temporales de los indicadores de ruido en los canales Upstream.
- Almacenar notificaciones emitidas por el CMTS.
- Detectar condiciones de error.

Estimación de Historias del Sprint 6.

Cálculo de la velocidad estimada para el Sprint 6 utilizando la técnica de cálculo de velocidad basado en días-hombres disponibles y factor de dedicación.

Factor de dedicación: $15/21=0,714$

Trabajadores	Días-Hombres (disponibles)	Factor de Dedicación
Miguel Angel Cortés	14	0,714
Yaima Toledo	2	0.714
Alejandro León	6	0,714

Tabla 9: Estimación del Sprint 6

Velocidad Estimada = $21 * 0.714$

Velocidad Estimada = 15 (puntos de historia)

Historias incluidas en el Sprint:

1. Confeccionar reportes temporales de los indicadores de ruido en los canales Upstream: 5 puntos de historia inicialmente utilizando la técnica de ojo de buen cubero.
2. Almacenar notificaciones emitidas por el CMTS: 5 puntos de historia inicialmente utilizando la técnica de ojo de buen cubero.
3. Detectar condiciones de error: 5 puntos de historia inicialmente utilizando la técnica de ojo de buen cubero.

Como probar cada historia del Sprint.

- 1- Confeccionar reportes temporales de los indicadores de ruido en los canales Upstream:

Entrar, abrir ventana de monitorear los indicadores de ruido de los canales de Upstream, pulse el botón de crear reporte de ruidos el sistema se encargue de crear un reporte de los indicadores de ruido.

- 2- Almacenar notificaciones emitidas por el CMTS:

Entrar, abrir ventana de notificaciones del CMTS, y comprobar que se está realizando una salva de las notificaciones emitidas por el CMTS.

- 3- Detectar condiciones de error:

Entrar, abrir ventana configurar condiciones de error e introducir las condiciones permitidas para ciertos parámetros de error, al cumplirse alguna de las condiciones el sistema mostrará un mensaje de alerta.

Lista de Miembros:

Miguel Angel Cortés: 60% de trabajo en el Sprint.

Yaima Toledo: 10% de trabajo en el Sprint.

Alejandro León: 30% de trabajo en el Sprint.

Lugar y momento definido para el Scrum diario:

Departamento de Informática: 10:00 am

Historias divididas en tareas:

Las tareas del Sprint 6 se pueden ver en el Anexo A6.

2.5 Conclusiones del Capítulo.

En este capítulo, tomando como guía la Metodología Scrum se realizó la pila del producto donde se definieron los requerimientos del sistema, los requisitos no funcionales del sistema, la pila de los Sprint y la descripción de cada sprint apoyado de las técnicas de estimación de Sprint. También se definieron las tareas para cada sprint que serán explicadas en los anexos.

CAPÍTULO 3: CONSTRUCCIÓN Y VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.

3.1 Introducción.

En el presente capítulo se plantean y detallan una serie de diagramas que ayudan y guían en la implementación del modelo de sistema, como son: el diagrama de casos reales de uso, modelo físico, además se define la concepción general de la ayuda, así como el estudio de factibilidad y la validación de la propuesta.

3.2 Historias Técnicas.

Diagrama de Casos Reales de Uso.

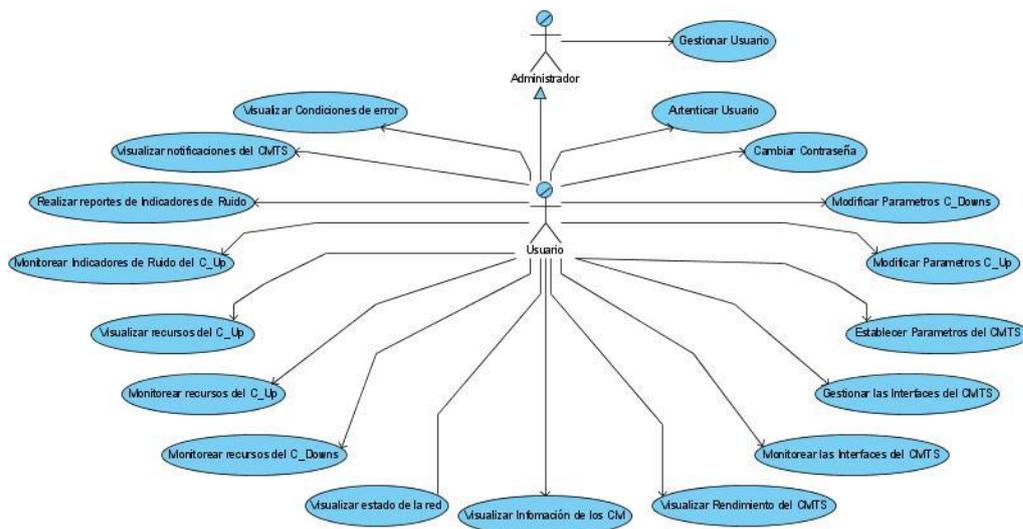


Figura 5: Diagrama de Casos de Usos.

Modelo Lógico de los Datos.

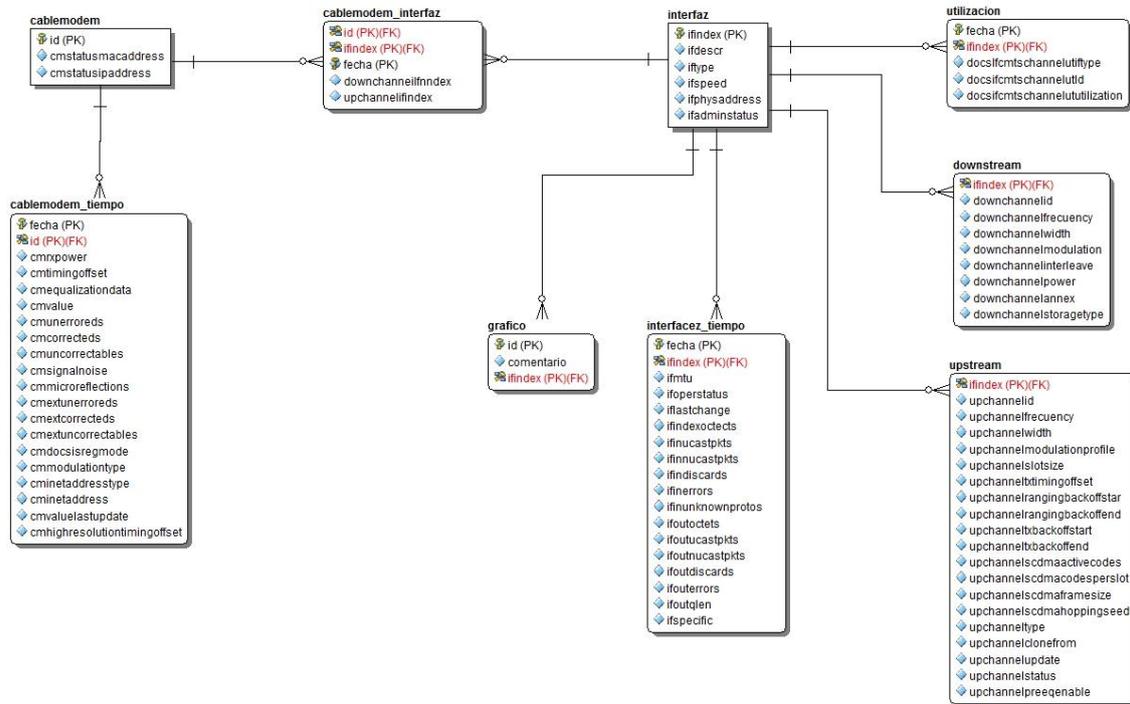


Figura 6: Modelo Lógico de Datos.

3.3 Principios de Diseño.

El diseño de sistemas se define como el proceso de aplicar ciertas técnicas y principios con el propósito de definir un dispositivo, un proceso o un sistema, con suficientes detalles como para permitir su interpretación y realización física.

3.3.1 Estándares en la Interfaz de Aplicación.

La primera impresión del usuario cuando interactúa con el sistema es el diseño de la interfaz gráfica. Es por ello que, para lograr una apariencia adecuada y que el usuario se sienta cómodo, se tienen en cuenta varios aspectos, sobre todo relacionados con tipografía, colores, gráficos, navegación, composición del sistema, etc., que a continuación se detallan. En el sistema el diseño de la interfaz está basado en ventanas, se muestran diferentes colores para resaltar algunos datos en particular, predominando el gris oscuro por la seriedad que trasmite. El vocabulario es lo menos técnico posible, acercándose al utilizado

por los usuarios. Se dispone de imágenes identificativas en algunas ventanas para ayudar al usuario en el momento de interactuar con el sistema. La letra utilizada en todo el sistema es Consolas (12) lográndose un diseño estándar en todo el sistema. Los mensajes de error son pequeños, claros y en español. Las interacciones con el sistema se basan en selecciones de tipo menú y en acciones físicas sobre botones, identificados con un nombre y con un ícono para una mayor comprensión de la acción que realiza. La entrada de información por parte de los usuarios se realiza a través de los componentes visuales utilizados en las ventanas. Las operaciones que se ejecutan al acceder a la información almacenada en la base de datos son rápidas. Todo esto se ha hecho con el objetivo de que el uso del sistema brinde comodidad al usuario.

3.3.2 Tratamiento de Errores.

En el sistema propuesto se deben evitar, minimizar y tratar los posibles errores, con el fin de garantizar la integridad y confiabilidad de los datos que se registran y muestran. Las posibilidades de introducir información errónea por parte del usuario deben ser mínimas, manteniendo un nivel de validación de la información y en caso de errores comunicar los mismos a través de mensajes y cuadros de alerta. Los mensajes de errores que emita el sistema tendrán un lenguaje de fácil comprensión para los usuarios.

3.3.3 Concepción General de la Ayuda.

La ayuda constituye una parte imprescindible en todo sistema. En el menú principal aparece una opción "Ayuda", que explicará de forma detallada cómo funciona el sistema, tratando de aclarar los puntos que podría causar duda al usuario. Cada una de las opciones del sistema, así como las consideraciones que se asumen en la ejecución de ellas está propiamente documentada para evitar cualquier tipo de confusión por parte del usuario. Cada aspecto de la ayuda ha sido diseñado con el objetivo de expresar explícitamente cómo y en qué orden debe operar el usuario.

3.4 Beneficios Tangibles e Intangibles.

Los beneficios obtenidos a partir del desarrollo del software se resumen en la eficiencia que se puede lograr en el proceso de chequeo y configuración de la red DOCSIS. Se reduce el tiempo de configuración de los parámetros de funcionamiento del CMTS. La información se encuentra segura y no está a expensas de errores en el instante del manejo de los datos.

Los beneficios obtenidos con el desarrollo del software permiten agilizar los procesos de Gestión del CMTS en la cabecera de Telecable Internacional en Cayo Santa María, Villa Clara.

3.4.1 Análisis de los Costos y Beneficios.

Para analizar si es factible o no elaborar el sistema es necesario analizar el costo, contra los beneficios de realizar el mismo.

Como resultado del presente trabajo de diploma, el costo del sistema para la empresa Telecable es insignificante si se compara con los software's que brinda el mercado internacional, siendo de gran aporte para el capital monetario de la misma. La utilización de este sistema facilitará la gestión del CMTS y ayudará a los supervisores a la toma de decisiones en tiempo real.

3.5 Validación de la Solución Propuesta.

Prueba T para muestras pareadas.

Una prueba T de Student, o Test-T es cualquier prueba en la que el estadístico utilizado tiene una distribución T de Student si la hipótesis nula es cierta. Se aplica cuando la población estudiada sigue una distribución normal pero el tamaño de la muestra es demasiado pequeño como para que el estadístico en el que está basada la inferencia esté normalmente distribuido, utilizándose una estimación de la desviación típica en lugar del valor real.[24]

Para realizar la validación del sistema informático propuesto se tomaron 15 observaciones del tiempo en minutos en que demoraba el proceso de gestión

dígase monitoreo y configuración del CMTS antes y después del sistema informático. Como se observó en la figura 7 que el procedimiento antes del software demoraba como promedio 8 minutos y después solo 4 minutos como promedio, por lo que a simple vista se aprecia la existencia de diferencias significativas entre ellos. Para comprobarlo estadísticamente se realizó la Prueba T para comparar las medidas antes y después del sistema.

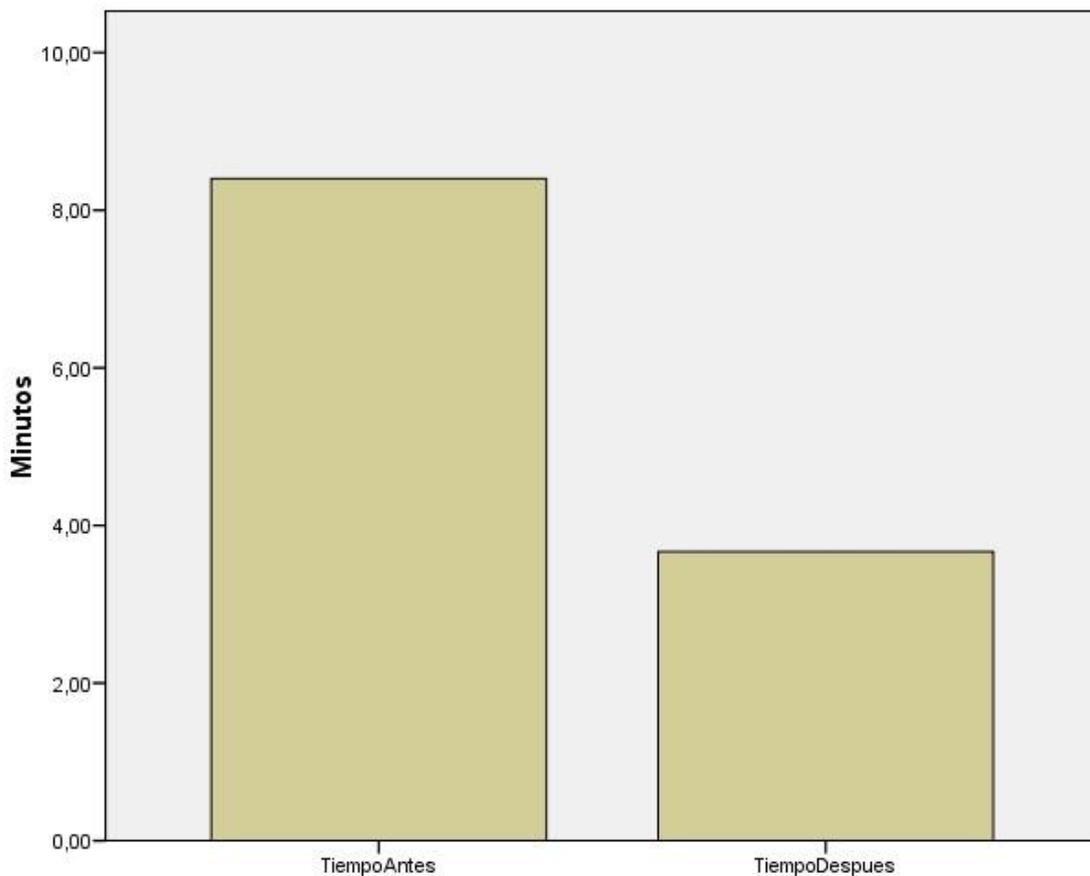


Figura 7: Comparación del tiempo antes y después del sistema.

Primeramente se comprobó que ambas variables (X-Tiempo antes del sistema y Y-Tiempo después del sistema.) seguían una distribución normal mediante la Prueba Kolmogorov-Smirnov.

Dicha prueba contrasta la hipótesis nula que plantea que la variable no sigue una distribución normal contra la hipótesis alternativa en que se considera que la variable sigue una distribución normal. Tomando como referencia un nivel de significación del 5 %, si este es menor que la significación asintótica, entonces

se rechaza H_0 , de lo contrario se acepta. Utilizando un nivel de significación de 0,05 al comparar con la significación asintótica de los estadísticos calculados (0,865 y 0,489) puede concluirse que se rechaza la hipótesis nula, y se acepta la hipótesis alternativa demostrando que ambas variables siguen una distribución normal, por tanto al cumplirse este supuesto puede realizarse la Prueba T.

Prueba de Kolmogorov-Smirnov para una muestra

		TiempoAntes	Tiempo Despues
N		15	15
Parámetros normales ^{a,b}	Media	8,4000	3,6667
	Desviación típica	3,48056	1,98806
Diferencias más extremas	Absoluta	,155	,215
	Positiva	,155	,199
	Negativa	-,116	-,215
Z de Kolmogorov-Smirnov		,599	,834
Sig. asintót. (bilateral)		,865	,489

a. La distribución de contraste es la Normal.
b. Se han calculado a partir de los datos.

Figura 8: Comparación del tiempo antes y después del sistema.

La Prueba T para muestras relacionadas plantea como hipótesis nula que la media de X es igual que la media de Y, considerando que no hay diferencias significativas entre ellas y la hipótesis alternativa plantea que la media de X es diferente a la media Y, es decir, que existen diferencias significativas entre ambas variables. Utilizando un nivel de significación de 0,05 al comparar con la significación del estadístico calculado (0,00) puede concluirse que se rechaza la hipótesis nula, y se acepta la alternativa, demostrando que hay diferencias significativas entre ambas medias.

Antes estas ventajas se puede plantear que el sistema informático es rápido, confiable y maneja de forma segura toda la información.

Estadísticos de muestras relacionadas

		Media	N	Desviación tip.	Error tip. de la media
Par 1	TiempoAntes	8,4000	15	3,48056	,89868
	TiempoDespues	3,6667	15	1,98806	,51331

Figura 9: Estadísticas de Muestras Relacionadas.

Prueba de muestras relacionadas

	Diferencias relacionadas					t	gl	Sig. (bilateral)
	Media	Desviación típ.	Error típ. de la media	95% Intervalo de confianza para la diferencia				
				Inferior	Superior			
Par 1 TiempoAntes - TiempoDespues	4,73333	1,90738	,49248	3,67708	5,78960	9,611	14	,000

Figura 10: Prueba de Muestras Relacionadas.

3.6 Conclusiones del Capítulo.

En el presente capítulo se plantearon una serie de diagramas que ayudaron en la implementación del modelo del sistema, como son: el diagrama de casos reales de uso, modelo lógico. Además se definió el tratamiento de errores, el estudio de factibilidad, se pudo ver los beneficios tangibles e intangibles. Así como se desarrolló el análisis de costos y beneficios y la validación de la propuesta.

CONCLUSIONES

Después de haber cumplidos los objetivos planteados, se plantean las siguientes conclusiones:

- Se realizó un estudio de los principales conceptos asociados al dominio del problema, se seleccionó como metodología para el desarrollo del software Scrum; java como lenguaje de programación; NetBeans v.7.1, ERStudio y Visual Paradigm como herramientas utilizadas y PostgreSQL como gestor de Base de Datos.
- Se elaboró un sistema informático que responde a las necesidades de la compañía de Telecom Internacional que radica en Cayo Santa María, Villa Clara. Se utilizó como metodología de desarrollo del software Scrum, se modelaron diferentes artefactos que describen las características del sistema utilizando UML.
- Se implementó el sistema diseñado para la gestión básica para la red DOCSIS de Telecom Internacional que radica en el Cayo Santa María, Villa Clara. El sistema se utilizó durante un período de prueba, que permitió realizar algunos ajustes.
- Se realizó la factibilidad del sistema obteniendo resultados satisfactorios y la validación del sistema utilizando la prueba T, demostrando estadísticamente la diferencia en los procesos referentes a la gestión del CMTS antes y después de creado el sistema.

RECOMENDACIONES

A pesar de que los objetivos trazados para la realización del trabajo de diploma fueron cumplidos, se recomienda:

1. Probar al máximo las funcionalidades que brinda el sistema durante un período amplio de tiempo para comprobar de forma práctica todas sus funcionalidades y obtener los datos necesarios para su mejora.
2. Continuar el estudio del estándar DOCSIS con el objetivo de incrementar las funcionalidades del sistema.

REFERENCIAS BILIOGRÁFICAS.

- [1] Pici, «Introduction to Cable Television», 2007.
- [2] «HFC Resumen».
- [3] Ramos, Rodriguez, y Santa, «HFC DOCSIS», 2011.
- [4] N. Shah y D. Kouvatsos, «A Tutorial on DOCSIS: Protocol and Performance Models».
- [5] B. Volpe, «Docsis Tutorial Series». 29-jun-2009.
- [6] Tooley y Bowman, «An overview of the DOCSIS (Cable Internet) Platform. Sandvine® Intelligent Broadband Networks», 2010, dic. 2010.
- [7] «Physical Layer Specification. Data Over Cable Service Interface Specifications DOCSIS 3.0 ed. USA.» CABLE TELEVISION LABORATORIES, 2010.
- [8] Carcamo, «Instalación CMTS», 2009. [Online]. Available: www.tripleplay.mx/support/instalacion-cmts.pdf.
- [9] Mauro y SCHMIDT, «Essential SNMP», 2da ed. USA: O'Reilly., 2005.
- [10] M. A. Miller, «Managing Internetworks with SNMP», 2da ed. USA: IDG Books Worldwide, Inc., 2004.
- [11] «Metodologías Tradicionales». [Online]. Available: <http://www.eumed.net/Metodologiastradicionales.htm>.
- [12] «Scrum y XP desde las trincheras».
- [13] «Que es UML». .
- [14] «Java 2 Manual FV».
- [15] Sun Microsystems, Inc, «El lenguaje de Programación Java™».
- [16] «PostgreSQL», 19-oct-2011. [Online]. Available: <http://grupove.org.ve/postgresql>.
- [17] «Lenguaje SQL», 02-jul-2009. [Online]. Available: <http://www.arsys.es/soporte/programacion>.
- [18] «NetBeans - EcuRed», 07-abr-2013. [Online]. Available: zotero://attachment/86/. [Accessed: 07-abr-2013].
- [19] «NetBeans - EcuRed», 07-ene-2013. [Online]. Available: <http://www.ecured.cu/index.php/NetBeans>. [Accessed: 07-ene-2013].
- [20] «Visual Paradigm - EcuRed», 07-abr-2013. [Online]. Available: zotero://attachment/83/. [Accessed: 07-abr-2013].

- [21] «Embarcadero Er/ Studio - Ensayos de Colegas - Gualber123». [Online]. Available: <http://www.buenastareas.com/ensayos/Embarcadero-Er-Studio/1251734.html>. [Accessed: 10-abr-2013].
- [22] H. Kniberg, «Scrum y XP desde las trincheras», *Version Gratuita ONLINE*.
- [23] «Sistema de Promoción y Gestión Comercial para la Oficina de Transferencia Tecnológica de la Universidad de Cienfuegos.», *CUJAE (UH)*.
- [24] «Prueba T de Student»». [Online]. Available: www.es.wikipedia.org/wiki/Prueba_t_de_Student.

BIBLIOGRAFÍA.

- [1] N. Shah y D. Kouvatsos, «A Tutorial on DOCSIS: Protocol and Performance Models».
- [2] J. H. Green, «Access Technologies: DSL and Cable. Executive Briefings in Key Technologies.», *USA: McGraw-Hill Companies, Inc.*
- [3] Tooley y Bowman, «An overview of the DOCSIS (Cable Internet) Platform. Sandvine® Intelligent Broadband Networks», 2010, dic. 2010.
- [4] «Arquitectura de Software». [Online]. Available: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/hernandez_j_pg/capitulo2.pdf
- [5] J. Peña, «Artículo para la revista Linux Actual número 17: “Gestión SNMP con Linux”», 12-feb-2001. [Online]. Available: <file:///I:/Descargas/snmp.html>. [Accessed: 07-mar-2013].
- [6] Downey, «DOCSIS 3.0 Overview», *Cisco Systems, Inc.*
- [7] B. Volpe, «Docsis Tutorial Series». 29-jun-2009.
- [8] Sun Microsystems, Inc, «El lenguaje de Programación Java™».
- [9] «Embarcadero Er/ Studio - Ensayos de Colegas - Gualber123». [Online]. Available: <http://www.buenastareas.com/ensayos/Embarcadero-Er-Studio/1251734.html>. [Accessed: 10-abr-2013].
- [10] Mauro y SCHMIDT, «Essential SNMP», 2da ed. *USA: O'Reilly.*, 2005.
- [11] Ramos, Rodriguez, y Santa, «HFC DOCSIS», 2011.
- [12] «HFC Resumen».
- [13] Pete Deemer, «Información Básica de Scrum». 2009.
- [14] Carcamo, «Instalación CMTS», 2009. [Online]. Available: www.tripleplay.mx/support/instalacion-cmts.pdf.
- [15] Pici, «Introduction to Cable Television», 2007.
- [16] «Java 2 Manual FV».
- [17] A. Z. Gorostiza, «La Disciplina de Arquitectura». [Online]. Available: <http://profesores.fi-b.unam.mx/adanzg/SwEng/Presentaciones/DisciplinaArquitectura.pdf>.
- [18] «Lenguaje SQL», 02-jul-2009. [Online]. Available: <http://www.arsys.es/soporte/programacion>.
- [19] M. A. Miller, «Managing Internetworks with SNMP», 2da ed. *USA: IDG Books Worldwide, Inc.*, 2004.
- [20] Amaro Calderón, Sarah Dámaris Valverde Rebaza. Jorge Carlos, «Metodologías Ágiles». 2007.
- [21] «Metodologías Tradicionales». [Online]. Available: <http://www.eumed.net/Metodologíastradicionales.htm>.
- [22] «NetBeans - EcuRed», 07-abr-2013. [Online]. Available: zotero://attachment/86/. [Accessed: 07-abr-2013].
- [23] «Physical Layer Specification. Data Over Cable Service Interface Specifications DOCSIS 3.0 ed. USA.» *CABLE TELEVISION LABORATORIES*, 2010.
- [24] «PostgreSQL», 19-oct-2011. [Online]. Available: <http://grupove.org.ve/postgresql>.
- [25] N. S. Kouvatsos, «Protocol and Performance Models», *University of Bradford, UK*.
- [26] «Prueba T de Student»». [Online]. Available: www.es.wikipedia.org/wiki/Prueba_t_de_Student.
- [27] «Que es UML». .

- [28] C. Ruata y J. Palacio, «Scrum Manager Proyectos». 2009.
- [29] «Scrum y XP desde las trincheras».
- [30] «Sistema de Promoción y Gestión Comercial para la Oficina de Transferencia Tecnológica de la Universidad de Cienfuegos.», *CUJAE (UH)*.
- [31] Parziale, Britt, Davis, y Forrester, «TCP/IP Tutorial and Technical Overview.», *USA: International Bussines Machines Corporation*.
- [32] «Televisión por cable - EcuRed», 07-abr-2013. [Online]. Available: <zotero://attachment/80/>. [Accessed: 07-abr-2013].
- [33] M. Domínguez-Dorado, *Todo Programación*. Madrid: Iberprensa, 2005.
- [34] T. Lockhart, «Tutorial de PostgreSQL».
- [35] «Visual Paradigm - EcuRed», 07-abr-2013. [Online]. Available: <zotero://attachment/83/>. [Accessed: 07-abr-2013].

ANEXOS

Anexo A: Tareas por Sprint.

Anexo A1: Tareas del Sprint 1.

Autenticar Usuario	Crear interfaz gráfica
	Validar mensajes de salidas
	Validar y encriptar contraseña
	Activar menú de navegación
Gestionar Usuario	Crear interfaz gráfica
	Insertar, eliminar y modificar un Usuario.
	Validar mensajes de Salidas
	Encriptar contraseña
Cambiar Contraseña	Crear interfaz gráfica
	Validar mensajes de salidas
	Encriptar contraseña

Anexo A2: Tareas del Sprint 2.

Modificar los parámetros de funcionamiento del canal Downstream.	Crear interfaz gráfica
	Modificar los parámetros del canal
	Validar mensajes de salidas
Modificar los parámetros de funcionamiento del canal Upstream.	Crear interfaz gráfica
	Modificar los parámetros del canal
	Validar mensajes de Salidas

Anexo A3: Tareas del Sprint 3.

Establecer parámetros de funcionamiento del CMTS.	Crear interfaz gráfica
	Modificar parámetros de funcionamientos del CMTS
	Validar mensajes de salidas
Monitoreo de las interfaces del CMTS	Crear interfaz gráfica
	Monitorear el funcionamiento de las interfaces.
Gestionar las interfaces del CMTS	Crear interfaz gráfica
	Habilitar o deshabilitar las interfaces del CMTS
	Validar mensajes de salidas
Mostrar rendimiento del CMTS.	Crear interfaz gráfica
	Monitorear el funcionamiento del CMTS.

Anexo A4: Tareas del Sprint 4.

Obtener la información disponible de todos los CM conectados al CMTS.	Crear interfaz gráfica
	Visualizar datos de los CM
	Validar mensajes de salidas
Mostrar información del estado de la red.	Crear interfaz gráfica
	Monitorear el funcionamiento de la red.
	Validar mensajes de salidas
Monitorear la utilización de los recursos de comunicaciones de los canales de Downstream.	Crear interfaz gráfica
	Visualizar los recursos de los canales Downstream
	Validar mensajes de salidas

Anexo A5: Tareas del Sprint 5.

Visualizar el uso de los recursos de comunicaciones del Upstream.	Crear interfaz gráfica
	Visualizar los recursos de los canales Downstream
Visualizar historial de los recursos de los canales de Upstream.	Crear interfaz gráfica
	Visualizar historial de los recursos
Visualizar los indicadores de ruido en los canales de Upstream.	Crear interfaz gráfica
	Visualizar los indicadores de ruido del canal
	Validar mensajes de salidas

Anexo A6: Tareas del Sprint 6.

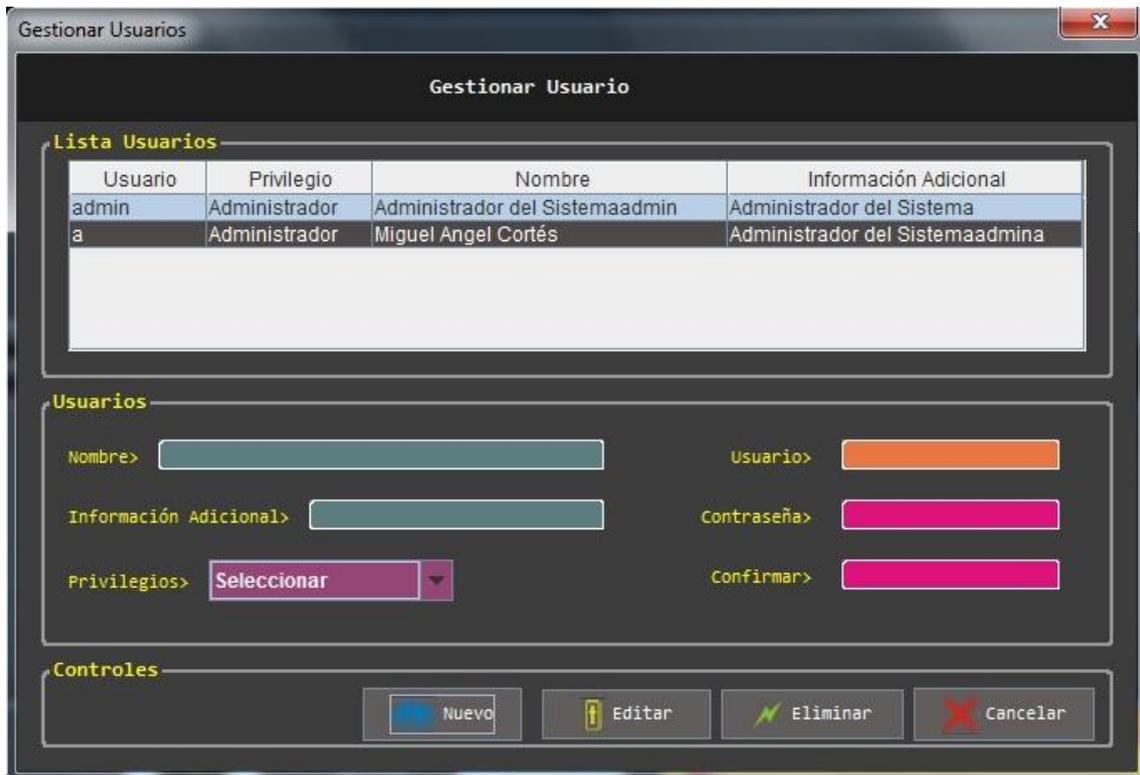
Confeccionar reportes temporales de los indicadores de ruido en los canales Upstream.	Crear interfaz gráfica
	Mostrar reportes de los indicadores de ruido del canal
	Validar mensajes de salidas
Almacenar notificaciones emitidas por el CMTS.	Crear interfaz gráfica
	Almacenar notificaciones emitidas por el CMTS
	Validar mensajes de salidas
Detectar condiciones de error.	Crear interfaz gráfica
	Mostrar condiciones de error

Anexo B: Prototipos de Interfaz de los casos de uso.

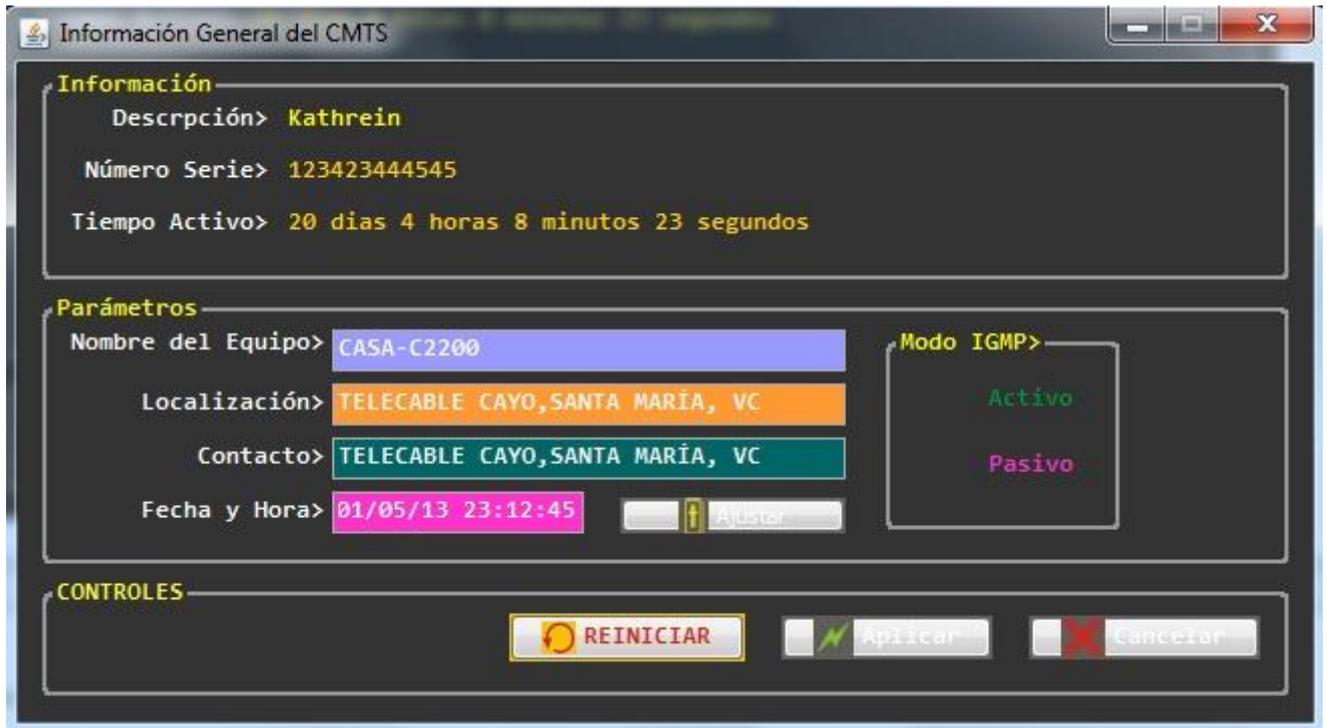
Anexo B1: Autenticar Usuario.



Anexo B2: Gestionar Usuario.



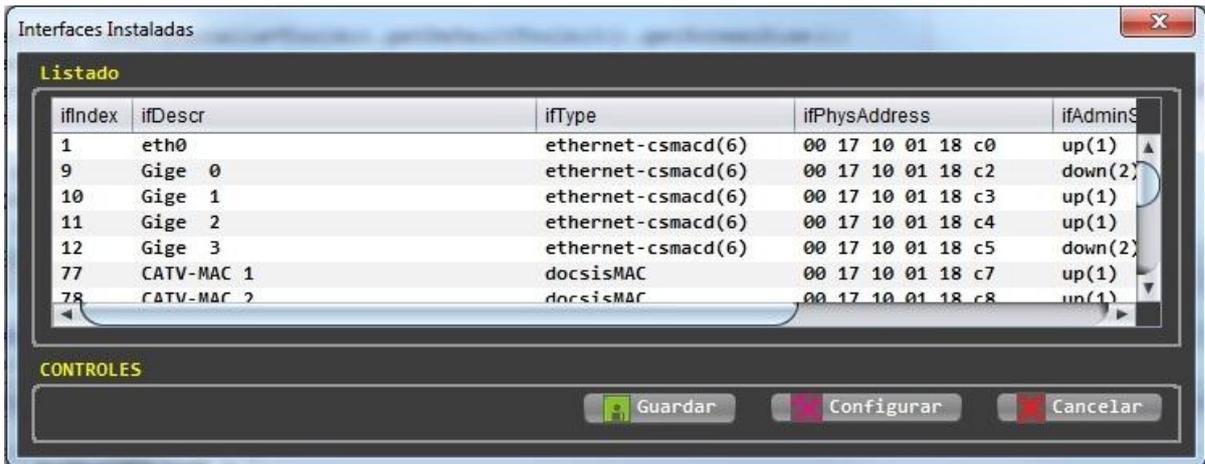
Anexo B3: Gestionar CMTS.



Anexo B4: Configurar Servidor.



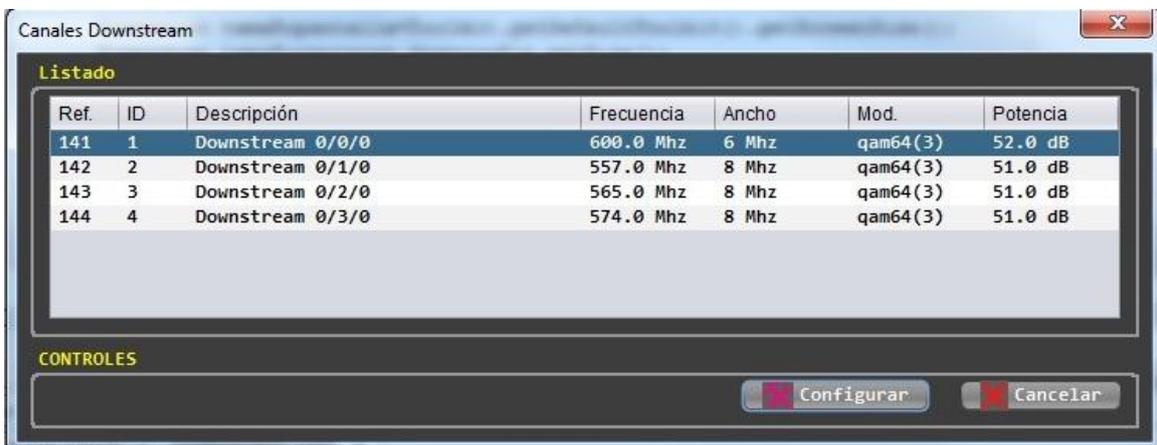
Anexo B5: Visualizar Interfaces.



Anexo B6: Visualizar Interfaces Mac.



Anexo B7: Visualizar Canales Downstream.



Anexo B8: Visualizar Canales Upstream.

Canales Upstream

Listado

Ref.	ID	Descripción	Frecuencia	Ancho	Mod.	Potencia Esperada
397	1	Logical Upstream Channel 1/0/0	26.00 Mhz	6400 Mhz	3	200
398	2	Logical Upstream Channel 1/1/0	15.00 Mhz	6400 Mhz	3	0
399	3	Logical Upstream Channel 1/2/0	23.00 Mhz	6400 Mhz	3	150
400	4	Logical Upstream Channel 1/3/0	37.00 Mhz	6400 Mhz	3	300

CONTROLES

Configurar Cancelar

Anexo B9: Visualizar Cable Modem.

Estado de los Cable Modems

Cable Modems Conectados

Información General

MAC> 77
 Estado> 77
 Up Time> cubitas.cfg
 Offset> 77
 No. Serie> 77
 Fecha> 07/05/2013 22:45

Señal

RxPwr> 77
 RxSNR> 77
 MoD> 77
 µRef> 77
 TxPwr> 77
 TxSNR> 77
 Pre EQ

Servidores

DHCP> 999.999.999.999
 ToD> 999.999.999.999
 TFTP> 999.999.999.999
 A. Conf cubitas.cfg

Servicio

CMS-MD> 77
 MD-CM-SG-ID> 77
 RCP-ID> 77
 RCC-ID> 77
 RCS> 77
 TCS> 77

Estadísticas

T1> 77 UCD inválido> 77 Palabras Código>
 T2> 77 MAP inválido> 77 Sin Error> 77
 T3> 77 Ex> 77 Pérdida SYNC> 77 Corregidas> 77
 T4> 77 Incorregidas> 77
 Tasa Error> 77

CONTROLES

Actualizar
 Cada 30 s
 Filtrar
 Restablecer
 Salvar Estado
 RESET CM
 RESET TODOS
 Reportes
 Cancelar

Anexo B10: Eventos Docsis.

Configuración de Eventos de DOCSIS

Control

IP Servidor SYSLOG> - - - -
 Estado Control Mensajes> Sin Restricciones
 Umbral Control Mensajes> 3
 Intervalo Control Mensajes> 3254345453 ???... (1..1234567890)

Parámetros

	LOCAL (No volátil)	TRAPS	SYSLOG	LOCAL (volátil)
Emergencia>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Alerta>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Critico>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Error	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Advertencia>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Noticia>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Información>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Depuración>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

CONTROLES

Limpiar Log Aplicar Cancelar

Anexo B11: Monitoreo Principal.

The screenshot shows a configuration window titled "Monitoreo Principal" with a close button (X) in the top right corner. The window is divided into several sections:

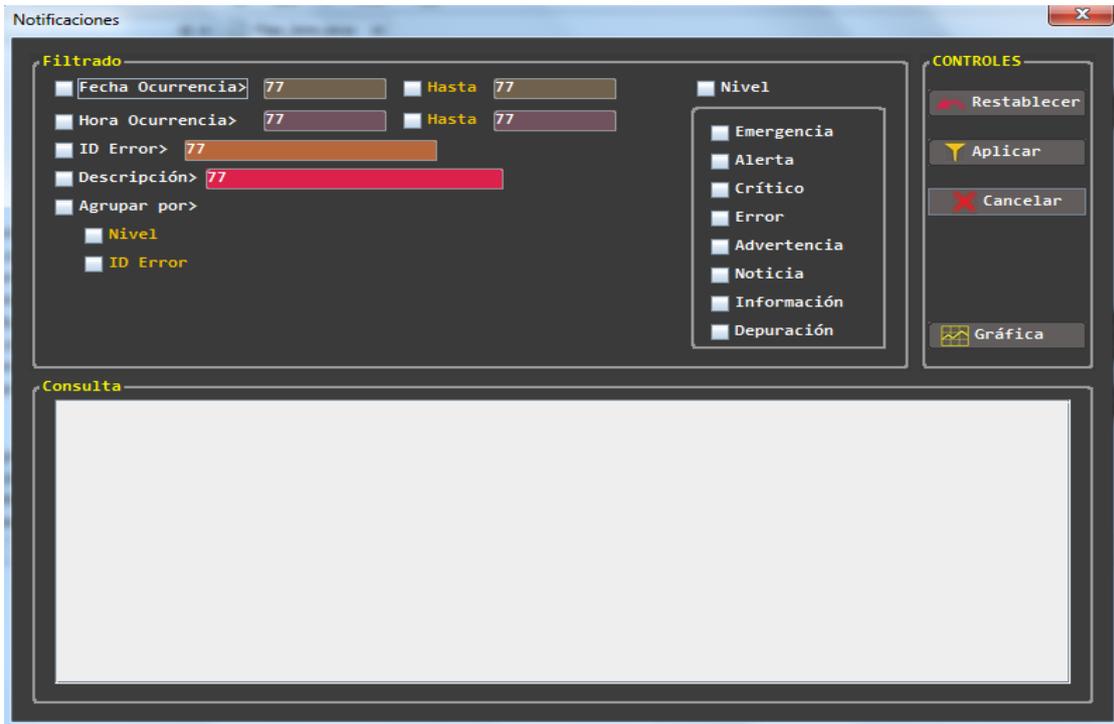
- CPU:** Contains a text field for "OID Gráfica" with the value ".1.3.1.4.1.20858.10.13.1.1.1.4.2" and a "Temporizador" (timer) field with a green slider and "s" unit, with "s min 5s" below it.
- Memoria:** Contains a text field for "OID Gráfica" with the value ".1.3.1.4.1.20858.10.13.1.1.1.4.2", a "Temporizador" field with a green slider and "s" unit, with "s min 5s" below it, and a "Tamaño Total" field with a purple slider.
- Gráficas de Tráfico y Estado:** Contains three "Intervalo" (interval) fields with green sliders and "s" units: "Intervalo Utilización" (min 10s), "Intervalo Abonados" (min 20s), and "Intervalo SNR" (min 30s).
- Notificaciones:** Contains two checkboxes and text fields: "Maxima Cantidad de SYSLOG" (checked, value 20000, range min 100 - max 1000) and "Máxima Cantidad de Traps" (checked, value 2000, range min 100 - max 1000).
- CONTROLES:** Contains two buttons: "Aplicar" (Apply) with a green lightning bolt icon and "Cancelar" (Cancel) with a red X icon.

Anexo B12: Flap List.

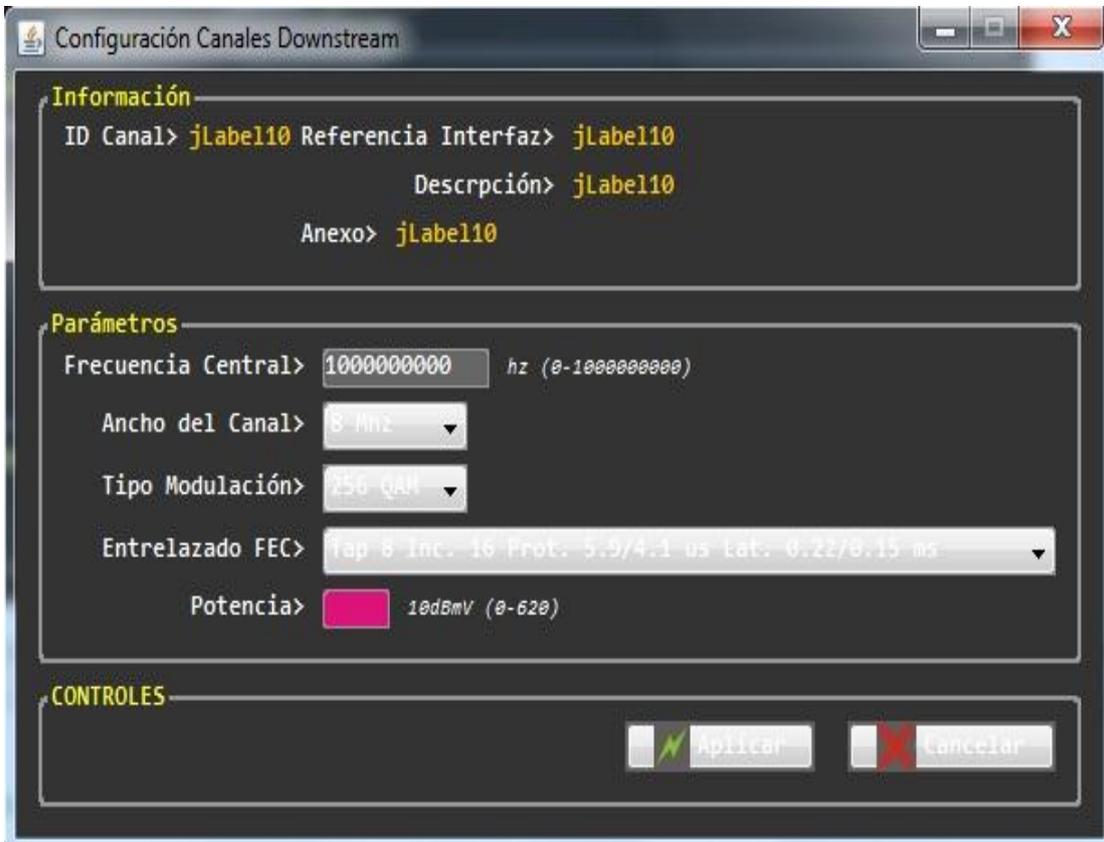
The screenshot shows a window titled "Flap List" with standard window controls (minimize, maximize, close) in the top right corner. The window contains:

- A section titled "Listado de Cable Módems que Fallan" (List of Cable Modems that Fail) with a large empty rectangular area below it.
- A "CONTROLES" (Controls) section at the bottom containing three buttons: "Reportes" (Reports) with a green icon, "Configurar" (Configure) with a purple icon, and "Cancelar" (Cancel) with a red X icon.

Anexo B13: Notificaciones.



Anexo B14: Canales Down Configuración.



Anexo B15: Canales Up Configuración.

Configuración Canales Upstream

Información

ID Canal> 1 Referencia Interfaz> 397

Descripción> Logical Upstream Channel 1/0/0

Tipo Canal> atdma(2)

Parámetros

Frecuencia Central> 26000000 hz (0-100000000)

Ancho del Canal> 6400 kHz

Perfil Modulación> 3

Habilitar Pre-Ecuación

ATDMA

Slot Size> 1 ticks de 6.25µs por mini-slot

Mecanismos de Vuelta Atrás

Alineación

Inicio Backoff> 0 2* (0-15) 16-propietario

Final Backoff> 4 2* (0-15) 16-propietario

Transmisión

Inicio Backoff> 1 2* (0-15) 16-propietario

Final Backoff> 15 2* (0-15) 16-propietario

SCDMA

Códigos Activos> 0 64-128
Excepto: 67, 71, 73, 79, 83, 89...

Códigos x MSlot> 0 2-32

Tamaño Trama> 0 0-32 (intervalos de extensión)

Semilla Salto Código> 0 15 bits

Recepción de Señal

Potencia Esperada> 300 10dBm (-160..160)

CONTROLES

Aplicar Cancelar